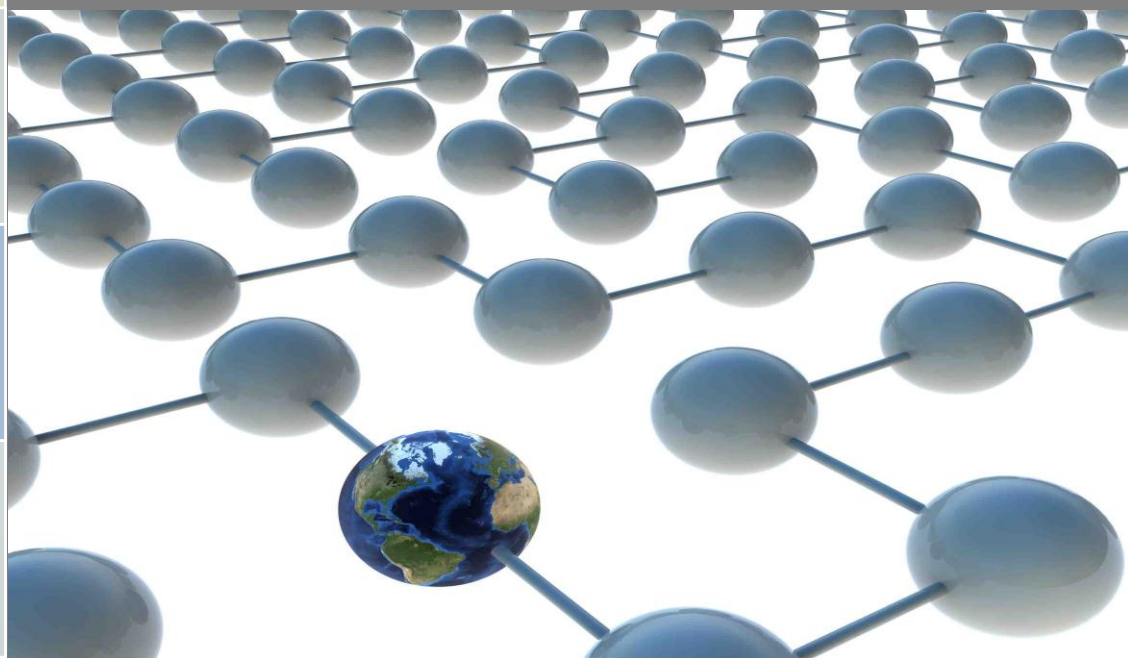


М.АРИПОВ, С.ДОТТОЕВ, М.ФАЙЗИЕВА

WEB ТЕХНОЛОГИЯЛАРИ

ўқув қўлланма



Тошкент 2013



Бугунги кунда таълим тизимида ахборот коммуникацион технологияларидан, жумладан Web технологиялар ва маълумотлар базасини бошқариш тизимларидан фойдаланиш, ҳамда маълумотларни бошқаришни автоматлаштириш муҳим вазифалардан ҳисобланмоқда. Ушбу ўқув қўлланмада шу масалалар хусусида, мавжуд муаммолар, уларни бартараф этиш усуллари ҳақида фикр юритилган.

Ўқув қўлланмада биргаликда ишлатилувчи занжир Apache+PHP+MySQL тизими ҳақида ёритилган бўлиб, ушбу учлик ёрдамида Webга мўлжалланган маълумотлар базасини яратиш ва улардан фойдаланиш бўйича маълумотлар берилган. Шунингдек, қўлланмада терминологик масалаларга ҳам эътибор берилган ва ёритилган.

Ўқув қўлланма бакалавр ва магистратура мутахассислиги талабалари учун мўлжалланган бўлиб, Web технологиялари ва Web дастурлаш фанларининг асосий мазмунини очиб беришга қаратилган. Қўлланмада талабалар мустақил ўзлаштириши учун топшириқлар ва уларни ечимлари, дастурий кодлар берилган.

Тақризчилар: А.Холжигитов, физика-математика фанлари доктори, профессор, ТАТУ илмий ишлар бўйича проректори.

Б. Абдурахимов, физика-математика фанлари доктори, профессор, ЎзМУ “Информатика ва тадбиқий дастурлаш” кафедраси мудири.

Муаллифлар: А.Арипов - физика-математика фанлари доктори, ЎзМУ “Информатика ва тадбиқий дастурлаш” кафедраси профессори.

С.Доттоев - ТДПУ, Ахборот технологиялари маркази дастурчиси

М.Файзиева - ТДПУ, Ахборот технологиялари маркази дастурчиси

КИРИШ

Интернетнинг пайдо бўлиши ўз навбатида мулоқотни амалга ошириш учун янги коммуникацион технологиялар, ҳамда маълумотлар базаси билан ишловчи технологияларни ишлаб чиқишни тақозо этди. Шу кунгача интернетда саҳифалар яратувчи ва улар билан ишловчи оддий технологиялар ишлаб чиқилган эди. РНР – МББТ(маълумотлар базасини бошқариш тизими) билан амалда осон ва тез боғлана олиш имкониятини берувчи Web-дастурлашнинг янги технологияларидан бири ҳисобланади.

РНР Webда ишловчилар учун html-саҳифаларни тез ва осон, динамик ҳолда ишлаши ва ўзгартириш учун имкониятлар яратиб берувчи технологиялардандир. РНР технологиясининг ажойиб хусусиятларидан бири - очик дастурий кодга эгаллиги ва бепул тарқатилишидир. РНР платформага боғлиқ эмаслиги, турли хил платформаларда ишлай олиш имконияти билан ажралиб туради. Шунинг учун дунёда кўплаб Web сайт яратувчилар ушбу технологияни танлашади. Сабаби, у бир кунда миллиондан ортиқ мурожаатларга хизмат қила олади, ҳамда унда скриптларни яратиш ва созлаш бошқа дастурлаш тилларига нисбатан осонроқдир.

Ушбу қўлланма уч қисмдан иборат: биринчи қисмда РНР технологияси ва унинг имкониятлари ҳақида сўз боради. РНР технологиясининг асосий конструкциялари, ички функциялари, объектга йўналтирилган дастурлаш борасидаги имкониятлари ҳақида маълумотлар берилган. РНР технологияси ёрдамида яратилган дастурларда HTML нинг баъзи элементларини ишлатиш ҳам батафсил ёритилган.

Иккинчи қисм маълумотлар базасини бошқариш тизимларининг (МББТ) имкониятлари ва кутубхоналари, РНР технологиясининг MySQL МББТ билан интеграциясига бағишланган. Ушбу бўлимда РНР технологияси ва MySQL МББТ ёрдамида динамик WEB сайтлар яратиш лойиҳалари ва уларнинг дастур кодлари мисоллар ёрдамида тушунтириб ўтилган.

Учинчи қисм РНР технологиясининг Web-тизимлардан фойдаланишда авторизацияни ташкил қилишга бағишланган.

1 ҚИСМ. PHP ТЕХНОЛОГИЯСИ

I БОБ. PHP ВА MySQL

Бу бобда PHP ни яратилиш тарихи (PHP/FI дан PHP5 гача); унинг имкониятлари (самарадорлиги, маълумотлар базаси билан интеграцияси, бириктирилган кутубхоналари, ўрганишда соддалиги, дастур коди); ишлатилиш усуллари; MySQL маълумотлар базаси ва унинг яратилиш тарихи, имкониятлари(самарадорлиги, соддалиги, хавфсизлиги, техник имкониятлари ва ҳ); Apache+PHP дастурий таъминотини ўрнатиш ва созлаш кабилар ўрганилади.

1.1. PHP тарихи ва унинг имкониятлари

PHP технологияси *Расмус Ледорф(Rasmus Lerdorf)* томонидан 1994 йилнинг куз ойларида яратилган. PHP технологиясининг биринчи версияси Perl да ёзилган бўлиб, CGI оболочка сифатида бўлган. PHP технологияси яратилганига унча кўп вақт бўлмаганлиги, яъни нисбатан янгилиги сабабли бу технология ҳақида маълумот анча кам. 1995 йилнинг бошларида **PHP** нинг бир мунча кенгайтирилган ва функционал кенгроқ имкониятли сотув версияси ишлаб чиқилди. Ушбу пакетнинг номи **Personal Home Page Tools** бўлиб, ўша вақтда PHP етарлича имкониятларга эга бўлган. 1995 йилнинг ўрталарига келиб эса форма ёрдамида маълумотлар билан ишловчи Form Interpreter (FI) қўшилди. Бу версия маълумотлар базаси билан ишлай олди, яъни mSQL ни қўллаш олди. 1998 йили шу технологиянинг аввалгиларига қараганда мукамалроқ ва қайтадан ёзилган PHP3 версияси яратилди. Шундан сўнг PHP дунё бўйича янада оммалашиб кетди.

NetCraft нинг ҳисоб китобларига кўра 1999 йилга келиб PHP дунё бўйича 1 миллиондан ортиқ сайтда ишлатилган. 2001 йилнинг январидан эса бу сон 5 миллионга етган. Бугунги кунга келиб, дастурчиларнинг 100 мингдан ортиғи PHP дан фойдаланишмоқда. PHP эса Apache серверида ишловчи 14,5 млн та сайтнинг тахминан 52% ида ишлатилмоқда.

Php нинг PHP3 версияси HTML саҳифада сценарий сифатида ёзилади. Унинг кўпгина синтаксислари C, C++, Java ва Perl (булар ҳам дастурлаш тиллари) лардан олинган ва PHP нинг янги имкониятлари қўшилган. Шуни таъкидлаш керакки, PHP бу дастурлаш тиллари орасида энг яхшисига айланди ва турли дастурлаш тилларидаги етишмовчиликларни максимал даражада бартараф қила олди. Унда хотирани белгилаш, ўзгарувчилар турларини кузатиш(оддий шартларда), ўзгарувчиларни аниқлаш зарурати туғилмайди. PHP технологияси барча асосий маълумотлар базасига осонгина боғлана олади ва бир қанча амалларни бажара олади.

PHP технологиясининг имкониятларига батафсил тўхталамиз.

Самарадорлиги. PHP самарадорлиги юқори бўлган WEB саҳифалар яратишда юқори натижа берувчи ягона технологиядир. У бир кунда миллиондан ортиқ мурожаатларга хизмат қила олади. PHP да скриптларни яратиш ва созлаш бошқа дастурлаш тилларига қараганда осонроқдир. PHP

кодлари html хужжатида матн сифатида ёзилади, турли IDE (Ишлаб чиқишнинг интеграллашган муҳити) заруратини истисно этади.

Маълумотлар базаси билан интеграцияси. PHP бир қанча маълумотлар базасини бошқариш тизимлари билан алоқа ўрната олади. У MySQL га қўшимча равишда PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, Interbase ва Sybasелар билан ҳам боғлана олади. Open Database Connectivity Standard (Маълумотлар омбори билан Очиқ Алоқа Интерфейси Стандарти) ни қўллаб, ODBC драйвери мавжуд бўлган ихтиёрий маълумотлар базасига уланиши мумкин. Бу Microsoft ва бошқа компаниялар маҳсулотлари учун тарқатилади.

Бириктирилган кутубхоналарининг мавжудлиги. Турли хил масалаларни ечишда кўпгина кичкина CGI – сценарийларини ёзишга ҳожат йўқ, чунки бу саҳифани юкланишини секинлаштиради ва сайт саҳифаларини яратишда халақит беради. PHP технологияси WEB билан боғланиш ва бир қанча умумий масалаларни ҳал қилиш имкониятини яратувчи, қўшимча ташқи кутубхоналар ёрдамида кенгайтирилган турли фойдали функциялар тўпламига эгадир. Унинг ёрдамида «тезгина» GIF тасвирлар яратиш, бошқа тармоқ хизматларига уланиш, электрон почта хабарларини жўнатиш, cookie-тўплamlари билан ишлаш, саҳифага киришга рухсат бериш, турли мураккабликдаги маълумотлар базаларини яратиш ва улар устида иш олиб бориш ҳамда расм ёки PDF хужжатларини яратиш мумкин ва буларнинг барчаси бир неча қатор код ёзиш билангина ҳал қилинади.

Нархи. PHP пакети мутлақо бепулдир. Унинг охириги версияларини <http://www.php.net> сайтидан бепул ёзиб олиш мумкин.

PHP ни ўрганиш. PHP синтаксиси бошқа дастурлаш тилларига асосланган, биринчи навбатда C ва Perl га. Агар фойдаланувчи C ёки Perl билан ёки C++ ёхуд Java билан таниш бўлса, PHPда дастур ёзишда тезда юқори самарадорликка эришиши мумкин.

Кўчувчанлиги. PHP пакетини турли операцион тизимлар бошқарувида ишлатиш мумкин. PHP кодини Linux ва FreeBSD сингари бепул, Solaris ва IRIX сингари коммерцион UNIX операцион тизимлар муҳитида, Microsoft Windows нинг турли версияларида яратиш имконияти мавжуд. Одатда, дастурлар турли хил PHP ўрнатилган операцион тизимларда ҳеч қандай ўзгаришларсиз ишлайди.

Хажми. PHP технологиясида яратилган ҳар қандай WEB-сайт хотирадан унча катта ҳажмни эгалламайди ва у очиқ тизим сифатида ишловчи илғор технология.

Дастур коди. Фойдаланувчи PHP дастури коди билан ишлаш ҳуқукига эгадир. Коммерцион мақсадли ёпиқ дастурлардан фарқли равишда, ушбу дастурлаш тилига бирор қўшимчалар ёки ўзгартиришлар киритиш мумкин. Тайёрловчи-фирма янги таҳрирлар (патчлар) ни чиқаришини кутиб ўтиришга ҳожат йўқ. Тайёрловчи дастурлар бозорини тарқ этади ёки ушбу дастурни қўллашдан воз кечади деб ўйланиш ҳам керак эмас.

PHP технологияси Perl, Microsoft нинг Active Server Pages(ASP), Java Server Pages(JSP) ва Allaire Cold Fusion технологияларига рақиб бўла олади

ва MySQL МББТ билан биргаликда функционал, динамик WEB сайтларни ярата олади.

PHP нинг мукамаллигини ва афзаллигини текшириш учун HTML билан солиштириб кўрамиз.

HTML сайт яратиш учун яхши тил, лекин уни кўшимча функциялари йўқлиги сабабли доимо янгилаб туриш керак. Бу ҳолда сайтни яна қайтадан кўриб чиқишга тўғри келади. Бу муаммони PHP - web дастурлаш тили ва MySQL маълумотлар базасини бошқариш тизими ёрдамида ҳал қилиш жуда осон, чунки PHP ва MySQL мукамал динамик сайтларни тезгина ярата олади. **HTML** ва **HTML+PHP+MySQL** да яратилган сайтларнинг фарқларини кўриб чиқамиз.

Вазифалари	HTML	HTML+PHP+MySQL
1:Яратиш мумкин бўлган сайт типлари	✗ Фақатгина статик сайтларни яратиш мумкин. Маълумотларни янгилаш қўлда амалга оширилади. (0)	✚ Бошқариш панели ёрдамида динамик сайтларни яратиш мумкин. (+1)
2: Маълумотларни сақлаш усули	✗ Маълумотлар файлларда сақланади. Маълумотларни янгилаш, ўзгартириш ва ўчириш қийин. (0)	✚ Маълумотлар МБ да сақланади. Бу шуни кўрсатадики, қидириш осон, умид-ишонч юқори, сайтни келажаги бор. (+1)
3: Ўзгартириш тартиблари	✗ Сайт ҳолатини ўзгартириш қийин. Сайт саҳифаларидаги элементларни янгисига тезда алмаштиришни амалга ошириб бўлмайди. (0)	✚ Сайтнинг ҳохлаган элементини (дизайнига мослаб ёзувларини ҳам) блокчи тузилма ёрдамида бир неча дақиқада ўзгартириш мумкин. (+1)
4: Кўшимча скриптларни ишлатиш	✗ Скриптларни (сўров, форум, мехмон, чат ва бошқа) ишлатишни имконияти йўқ. (0)	✚ Сайтнинг керакли блокчига скрипт ташкил қилиш мумкин. Бу имконияти ҳисобига, PHP сайт чегараланмаган ҳолда кенгая олади (+1)
5: Сайтни хостингга жойлаштириш	✚ Сайтни бепул хостингга жойлаштириш мумкин. Бу айниқса кичикроқ бюджетли фойдаланувчиларга қўл келади. (+1)	✖ Пуллик хостинглар бепуларига қараганда ишончлироқ. PHP сайтини жойлаштириш учун минимум 20\$ тўлашингиз зарур бўлади. (-1)
6: Ҳисоблагич	✗ Бирор бир файл неча марта юклаб олингани, сайт саҳифаларига неча марта мурожаат бўлгани, ҳозирда сайтда неча фойдаланувчи ишлаётгани ва бошқаларни ҳисоблаш имконияти мавжуд эмас. (0)	✚ MySQL да эса бундай масалалар осон ва тез ечилади. Ихтиёрий элементнинг, базадаги мавжуд маълумотлар (мақола, файл, изоҳ ва бошқалар) нинг сонини ҳисоблаш ва натижани сайтнинг керакли қисмига чоп этиш мумкин. (+1)
7: Сайт ҳажми	✗ Сайт кенгайтирила бошлаганда яратилаётган файллар сонини ошириш сайт ҳажминини катталаштиради, у ҳолда сайтни бошқариш ва сайт	✚ Барча маълумотлар МБ сида сақланади, сайт ҳажми амалда ўзгармайди. (+1)

	захира нусхасини олиш ҳам кийинлашади. (0)	
8: Қўшимча функциялар	✗ Мақолалар, дарслар, материаллар ва бошқаларни рейтингини яратиш, изоҳ, махфий бўлимлар ва бошқалардан фойдаланиш мумкин эмас. (0)	✚ Сайтни такомиллаштириш, янги саҳифалар яратиш, тайёр натижаларни ва бошқаларни ишлатиш учун ихтиёрий қўшимча модул ёзишингиз мумкин. (+1)
9: Ўрганишнинг кийинлиги	✚ Дунё бўйича жуда содда ва тез ўргана олиш мумкин бўлган, хаттоки кодлари хатолар билан ёзилганда ҳам ишлай оладиган тил. (+1)	✖ HTML га қараганда мураккаброк ва биров кийинрок, лекин сайт яратадиган бошқа ихтиёрий тилларга нисбатан осон. (-1)
10: Ташрифчилар учун қулайлиги	✗ Сайтдаги бир категорияга тегишли бўлган кўпгина материалларни автоматик тарзда бўлиб саҳифада кетма кет кўрилишини таъминлаш мумкин эмас. Сайтда оддийгина саҳифадан саҳифага ўтишни ҳам қўлда ёзиш керак. (0)	✚ Барча саҳифадан саҳифага ўтишлар учун унча катта бўлмаган битта код бўлагини ёзиш кифоя, ундан ташқари катта саҳифаларни бўлакларга бўлиш ҳам мумкин. (+1)
11: Блокли тузилмани жойлаштириш (конструктор сифатида)	✗ Ҳажми катта, бошқариш мураккаб бўлган сайтларни мантиқий қисмларга (бўлакларга) ажратиш мушкулроқ. (0)	✚ Сайтни ишлаб чиқишда блокли моделдан фойдаланилса, у ҳолда сайт «эгиловчан» бўлади ва уни бошқариш мумкин, блокларни керакли йўналиш бўйича кўчириш ва ихтиёрий дизайнга жойлаштириш мумкин. (+1)
12: Ўқиш ва сайт яратиш учун қизиқиш	✗ Ҳамма нарсa бир хил ва зерикарли. Тил жуда содда, хатоларни амалда аниқлаш мумкин эмас, мантиқий амаллар қатнашиши кам. (0)	✚ PHP сайтини яратиш давомида мантиқий фикрлашга, баъзи масалаларни ҳал қилишда ва ечимини топишда кўпроқ изланишга ва хатоларни ўрганишга тўғри келади. Шунинг учун тилнинг турли имкониятларини ўрганиб, ҳар доим янгиликлардан хабардор бўлиб туриш лозим. (+1)
13: Захирага нусха олиш	✗ Захирага нусха олишда сайтнинг кўпгина файлларини ўзингизнинг компьютерингизда сақлаб қўйишингиз лозим. (0)	✚ Барча маълумотлар МБ сида сақланади, захирага нусха олиш бир қанча секундлар давомида бажарилади. Захира нусхаларини ўзингизни компьютерингизда ёки серверда сақлашингиз мумкин. (+1)

Жадвал натижаларидан сайтни оддий HTML да яратишдан кўра HTML+PHP+MySQL ёрдамида яратиш мукамал, ишончли, қулай ва осон эканлиги кўриниб турибди.

Хулоса ўрнида айтадиган бўлсак, PHP

- ✓ Юқори самарадорлиги;
 - ✓ Турли маълумотлар базаси тизимларининг интерфейслари мавжудлиги;
 - ✓ Web билан боғланганда кўпгина масалаларни ечиш учун кутубхоналарни яратиши;
 - ✓ Ўрганиш ва фойдаланишнинг қулай ва соддалиги;
 - ✓ Дастур кодларининг очиқлиги;
- каби имкониятларга эга бўлган мукамал технология ҳисобланади.

1.2. MySQL: тарих ва имкониятлар

SQL (Structured Query Language — Структуралашган сўровлар тили) — реляцион МБ билан ишлашда қўлланиладиган сўровлар тили.

MySQL(My Structured Query Language) – реляцион маълумотлар базасини бошқариш тизимидир. MySQL нинг тарихига назар солсак – бу тизим Михаэль Видениус томонидан яратилган ва дастурий таъминот бозорига 1996 йили чиқарилган бўлса-да, аслида бу тизимдан 1979 йилдан буён кенг фойдаланилмоқда.

MySQL ни фақат mSQL камчиликларига жавоб сифатида қараш нотўғридир. Унинг яратувчиси Михаэль Видениус (у Monty номи билан ҳам маълум) 1979 йилда швед компанияси хисоблаш ишларида фойдаланиш учун UNIREG номли маълумотлар базасини бошқариш воситасини яратди. 1979 йилдан сўнг UNIREG бир неча тилларда ёзилди ва катта маълумотлар базаларини қўллаш учун кенгайтирилди.

1994 йилда ТсХ WWW учун амалий дастурлар ярата бошлади ва бу лойихани қўллашда UNIREG дан фойдаланди. Бироқ, UNIREG катта харажатлар талаб қилгани учун, ундан Web-саҳифаларни динамик ишлаб чиқиш учун муваффақиятли фойдаланиб бўлмади. ТсХ шундан сўнг SQL ва mSQL га мурожаат қилди. Лекин ўша пайтда mSQL фақат 1.x релизлари шаклида мавжуд эди. mSQL 1.x версиялари ҳеч қандай индексларни қўлламас эди ва шунинг учун UNIREG дан унумдорлиги паст эди. Видениус mSQL автори Хюз билан боғланди ва mSQL ни UNIREG даги V+ ISAM қайта ишловчисига улаш фикри билан қизиқтирмоқчи бўлди. Лекин Хюз шу пайтга келиб, mSQL 2 ни яратиш йўлида анча иш қилган ва индекслар билан ишлаш воситаларини яратган эди.

Шундан сўнг ТсХ дастурчилари ўз талабларига кўпроқ мос келувчи маълумотлар базалари тизимини яратишга қарор қилдилар. Дастурчилар UNIREG ни асос қилиб олдилар ва сони ошиб бораётган ўзга дастурчилар утилиталаридан фойдаландилар. Ўз тизимлари учун бошида mSQL учун яратилган API(*application programming interfaces – иловаларни дастурлаш интерфейси*) билан деярли устма-уст тушувчи API яратдилар. 1995 йил май ойига келиб ТсХ компанияси ички талабларини қаноатлантирувчи маълумотлар базаси, - MySQL 1.0 га эга бўлдилар.

MySQL нинг имкониятлари билан танишиб чиқамиз.

Самарадорлиги. MySQL – фойдаланиш учун жуда тезкор, ишончли, осон ва оддий тизимдир. MySQL client/server технологияси асосида ишлайди. Ўзининг енгил қўлланилиши билан MySQL тизими Internet тармоғида жуда кенг тарқалган. Ҳозирги вақтда WEBга мўлжалланган маълумотлар базасини ва яхши Internet проектларини MySQL тизимисиз яратиб бўлмайди. У утилита кўринишда ёки код кўринишда ва шулар қатори бошқа дастурлар ичига илова қисмида жойлаштирилса ҳам бўладиган тизим. MySQL амалий, ҳамда барча Internet – проектлари учун идеал ҳолатда тўғри келади. Кўпинча провайдерлар ҳам хостинг серверга маълумотлар базасини қўшиб тавсия қилишади ва шулар қатори ҳамма дизайнерлар ҳам MySQL-ни танлашмоқда.

Арзон нархлилиги ва фойдаланишдаги соддалик

Дастур кодинининг очиқлиги. MySQL пакети очиқ кодли дастурий таъминот ҳисобланади, уни бундай номланишининг сабаби бу – ҳар бир фойдаланувчи дастурни ишлата олиши ва кодини ўқиб чиққан ҳолда ички кодини ўрганиши ва ўзига мослаштириб ўзгартириши мумкин. Худди PHP каби MySQL нинг дастурий кодини очиқ кўриш ва ўзгартириш умкин.

Арзон нархлилиги - MySQL каби очиқ кодли дастурларни Интернет тармоғидан кўчириб олиш ва ундан бепул фойдаланиш мумкин. MySQL пакетига талаб бўлган пайтларда очиқ кодли дастурий таъминот лицензиясига асосан бепул тарқатилади. Унча қиммат бўлмаган нархда унинг савдо лицензиясини ҳам олиш мумкин. MySQL регламент лицензияси GPL(General Public License) орқали берилган ва бу лицензияни қуйидаги Web-сайт орқали олиш мумкин: <http://www.gnu.org./license/>.

Созлаш ва ўрганишда соддалиги. Кўпгина замонавий МББТларида SQL қўлланилган. Агар фойдаланувчи бирорта МББТлари билан таниш бўлса, MySQL га ўтишда у ҳеч қандай қийинчиликка дуч келмайди. Худди кўпгина дастурлар сингари MySQL ни созлаш ва ўрнатиш ҳам содда.

Хавфсизлиги

Маълумотлар хавфсизлигини юқори даражада таъминлайди. Mysql тизими маълумотларга рухсат беришни бошқаради, маълумотлар билан ишлаш учун бир вақтнинг ўзида бир нечта фойдаланувчига тезкор рухсатни таъминлайди ва фойдаланувчиларга фақатгина рухсат берилган маълумотлар билан ишлашига кафолат беради. Шунинг учун, Mysql тизими кўп фойдаланувчили, кўп оқимли (потокли) тизимдир.

Пароллар ва ҳуқуқларнинг мукамал тизими. Тизим парол ва ваколатларга (привилигияларга) асосланган юқори даражали, мукамал ва хавфсиз бўлиб, хост воситалари орқали текширувни ташкил қилишга имкон яратади. Пароллар химояланган, боғланиш вақтида барча трафик пароллари шифрланади. Пароллар жадвалида 16 тагача калит мавжуд бўлиб, ҳар бир калит 15 тагача майдондан фойдалана олиши мумкин.

MySQL МББТининг техник имкониятлари

- Ядро потокларидан фойдаланишда тўлиқлигича кўп оқимлиликлка мос келади. Яъни, кўп процессорли машиналарда ҳам бемалол ишлай олади;
- Оқимлиликлка асосланган, хотирани жуда тезкор тақсимлаш тизими;
- MySQL кенг спектрли дастур интерфейси(API)ни қўллаб қувватлашни таъминлайдиган мижоз-сервер тизимидир. API-интерфейси ёрдамида C, C++, Eiffel, Java, Perl, PHP, Python, Ruby ва Tel тилларига боғлана олади;
- SQL функциялари юқори оптималлаштирилган класслар кутубхонасидан фойдаланган ҳолда амалга оширилади ва етарлича тез бажарилади. Одатда, сўров инициализация қилингандан сўнг ҳеч қандай хотира тақсимооти амалга оширилмайди;
- MySQL коди савдо ва шунингдек очиқ дастур коди сифатида хотира бузилишини текширувчи асбоблар ёрдамида синаб кўрилган;

- MySQL серверидан мижоз – сервер тармоқ муҳитида алоҳида дастур сифатида ҳам фойдаланиш мумкин. Бундан ташқари, у кутубхона (lib) кўринишида ҳам тақдим этилади ва бу кўринишда алоҳида мустақил дастурларга осонлик билан бириктириб олиниши мумкин. Бундай дастурлар алоҳидаланган ва тармоққа уланмаган муҳитларда ҳам қўлланилиши мумкин;
- Кўчувчанликни таъминлашда Automake, Autoconf ва Libtool – GNU инструменларидан фойдаланилади;
- Сақлашнинг транзакцион ва транзакционмас механизмларини таъминлаб бера олади;
- Ташқи сақлаш механизмларини қўшиш жудаям оддий. Бу нарса хусусий яратилган маълумотлар базасига SQL интерфейсини қўшиш керак бўлганда қўл келади;
- Иккилик иерархияга асосланиб индекслар сиқиладиган, юқори тезликли диск (MyISAM) жадвалларидан фойдаланади;
- MySQL ўзида кўпоқимли SQL серверни жамлаган, маълумотлар базасида турли ҳисоб-китобларни, ундан ташқари турли хил мижоз дастурларини ва кутубхонани бошқаради. Шунингдек, MySQL серверини кўп оқимли кутубхона сифатида ўрнатиб, бошқа дастурлар билан осон боғлаш мумкин ва ихчам, анчайин тез, бошқариш осон маҳсулотга эга бўлинади.

Маҳаллийлаштириш

- Барча маълумотлар ISO8859_1 форматида сақланади;
- Сатрли маълумотлар билан ишловчи барча амалларда қайта ишланаётган сатр белгиларининг регистрга аҳамият қаратилмайди. Масалан: терак ~ ТерАК. Регистрга сезгир бўлган мўтадил сатрли барча устунларни таққослаш ва MySQL тизимини ишга тушириш вақтини ўзгартириш мумкин;
- Тизим мижозларга хатолар ҳақида турли хил тилларда хабар бера олади. latin1 (ISO-8859-1), german, big5, ujis ва бошқалар ёрдамида кодли жадвалларни тўлиқ қўллаб қувватлайди. Масалан, жадвал номлари ва устун номларида ҳам 'a', 'ä' ва 'b' скандинавия белгиларини ишлатиш мумкин.
- MySQL тизими бир қанча турли хил кодировкаларни қўллаб қувватлайди, шунинг учун улар компиляция вақтида ҳам, бажарилиш вақтида ҳам бир хилда кўрсатилиши мумкин. Барча маълумотларни саралаш танланган кодировкага (танланмаган ҳолда шведча набор ишлатилади) асосланган ҳолда бўлади ва улар танланган кодировкада сақланади.

Турли операцион тизимларга мослиги

Бир қанча турли операцион тизимларда ва платформаларда ишлайди. MySQL дан UNIX, Microsoft Windows ва бошқа турли муҳитларда ҳам фойдаланиш мумкин.

Тармоқлараро боғланиш

ODBC драйвер (дастур коди билан биргаликда). Мижозлар MySQL серверига ихтиёрий платформада TCP/IP сокети ёрдамида боғланишлари мумкин. Windowsнинг NT (NT, 2000 или XP) тизимида мижозлар канал номлари орқали, UNIX тизимида эса мижозлар базага UNIX-домени сокети файли орқали боғланишлари мумкин. Connector/ODBC интерфейсининг ички

коди очик бўлиб, ODBC-боғланишини ишлатиш орқали MySQL ни мижоз дастурлари билан боғланишида ёрдам беради. Мижоз дастурий таъминоти Windows ёки UNIX бошқаруви ёрдамида бажарилади. Масалан, MySQL тизимига боғланиш учун MS Access ни ишлатиш мумкин. Худди шундай JDBC боғланишини ишлатиш ёрдамида Connector/JDBC интерфейси Java да MySQL ни мижоз дастурлари билан ўзаро келишувини таъминлайди. Мижоз дастурий таъминоти Windows ёки UNIX бошқаруви ёрдамида бажарилади. Connector/JDBC интерфейсининг ички коди очик.

Масштабланиш ва чегара.

Жадвалда 64 тагача индексни ишлатишга рухсат беради (MySQL 4.1.2 версияларида эса 32 тагача). Ҳар бир индекс ўзида 1 дан 16 тагача устун ёки устун қисмини жамлайди. Индекснинг максимал узунлиги 1000 байт (MySQL 4.1.2 версиясида 500 байт).

Жадвалнинг ҳажми

Жуда катта ҳажмдаги маълумотлар базасини қўллаб қувватлайди. Масалан, MySQL маълумотлар базаси ўзида 50 млн тагача ёзувларни жамлай олади. Бошланғич ҳолатга кўра жадвалнинг максимал ҳажми 4 Гбайт га тенг. MySQL нинг баъзи бир версияларида MyISAM сақлаш механизми ёрдамида жадвал ҳажмини 8 Гбайтгача катталаштириш мумкин (масалан MySQL 3.23 да). Жадвалнинг максимал ҳажмини кўриш учун SHOW TABLE STATUS ёки `myisamchk -dv жадвал_номи` буйруқларидан фойдаланилади. Агар катта жадваллар фақатгина ўқиш учун ишлатилаётган бўлса, у ҳолда уларни `myisampack` утилита ёрдамида сиқиш ва бир нечта жадвалларни битта жадвалга бирлаштириш мумкин. Қоидага кўра, бу утилита жадвални бошланғич ҳажмидан 50% гача сиқиши мумкин. Натижада етарлича катта жадваллар билан ишлаш имконияти туғилади.

MySQL MERGE кутубхонасига эга бўлиб, ушбу кутубхона идентификация тузилишли MYISAM жадваллари тўпламини худди бир бирлашган жадвал сингари бошқаришга имкон беради.

MySQL узлуксизлиги (Стабильность)

Асосий дастур коди ўтган асрнинг 80 йиллари бошларида ишлаб чиқилган. У сақлашнинг оригинал механизмидан фойдаланиб мустаҳкам базани таъминлайди ва ISAM жадвалининг форматини қўллаб қўллайди.

MySQL нинг айни вақтда жуда кўп муҳолифлари мавжуд бўлиб, улар қаторига mSQL, dbm, Hyperware, Informix, Interbase, Sybase, PostgreSQL, Microsoft SQL Server ва Oracle лар қиради.

MySQL нинг имкониятларини кўриб чиққан ҳолда муҳолифлари, яъни бошқа МББТлари билан солиштириб кўрамиз. MySQL тизими катта маълумотлар базаси билан ишлаш жараёнида мавжуд ўхшаш ҳолатларни таққослашда, юқори тезликни тўлиқ таъминлашни бошқариш учун ишлаб чиқарилган. MySQL шубҳасиз, жуда ҳам тез ишлайди. Унинг самарадорлиги нисбий тестлаш натижаларини <http://dev.mysql.com/tech-resources/crash-me.php> саҳифасида кўриш мумкин. Кўпгина бундай тестлар натижаси шуни кўрсатадики, MySQL ўзининг муҳолифларидан кўра анча тез ишлайди.

Қуйида MySQL муҳолифлари билан техник ишлашнинг солиштириш кўрсатилган. Бу эталон солиштириш – бир ёқлама шунинг учун амалларни бажариш вақти озгина ўзгариши мумкин.

(Битта машинада NT 4.0 тизимида олиб борилган):

Маълумотлар базасини бошқариш тизимлари	Индекс бўйича 2000000 сатрни ўқиш (секунд)	350768 сатр киритиш ёки қўшиш(секунд)
mysql	367 (249)	381 (206)
mysql_odbc	464	619
db2_odbc	1206	3460
informix_odbc	121126	2692
ms-sql_odbc	1634	4012
oracle_odbc	20800	11291
solid_odbc	877	1801
sybase_odbc	17614	4802

Жадвалдан шунини кўришимиз мумкинки, MySQL муҳолифлари ичида энг тезкор ишловчи МББТ экан.

Биз, маълумотлар базаси нима эканлигини, MySQL маълумотлар базасини бошқариш тизими имкониятлари, ишлаш принципи, сошлаш усуллари ва бошқа муҳолифлари билан фарқини кўриб чиқдик.

Хулоса қилиб шунини айтиш мумкинки MySQL бу:

- Ҳар томонлама. РМББТ бўлганлиги сабабли бир вақтнинг ўзида бир нечта жадвалларга мурожаат қилиш имконига эга ва бир нечта сўровларни қўллаш олади.
- Бир юришда боғланган бир қанча маълумотлар билан яхши алоқа ўрната олади.
- Турли хил компиляцияларни кенг соҳада назорат қилади.
- Кўп оқимли, алоқанинг оптималлик тарафи бир нечта ўтиш (проход) натижасида кўпгина маълумотлар қўшилиши, мосланувчан системанинг афзалликлари.
- Жадвалларга кўп марта мурожаат бўлганда, уларнинг дискдаги асли эмас, тезкор хотирадаги вақтинчалик ишлатиладиган хеш-жадваллари қўланилади.
- Оптималлашган бир томонлама мультибоғланишдан фойдаланиб, жуда тезкор боғланишга эга.

Маълумотлар базасини бошқариш тизимлари бу - ҳар бир шахс ёки бирон бир катта корхона маълумотларини бир жойга тўплаган ҳолда бошқаришни қулайлаштиради ва вақтни тежайди. Шунинг учун ҳам ҳозир ҳеч бир интернет провайдерлари маълумотлар базасисиз тармоқда иш олиб боролмайдилар.

1.3. XAMPP дастурий таъминотини ўрнатиш ва созлаш

Ушбу бўлимда энг яхши серверлардан бири билан танишиш мумкин. Бу — Apache серверидир. Apache –Internet да кенг тарқалган web-серверлардан биридир. Ҳозирги вақтда Apache дастурий таъминоти тахминан бутун жаҳон Web-узел(тугун)ларининг ярмида ўрнатилган. Шунингдек, баъзи иловалар ҳам тавсия этилади. Булар PHP ва машҳур MySQL МББТ интерпретаторидир. Китобни бу қисми билан танишгандан ва интернетдан дистрибутивларни олгандан (ҳаммаси бепул) кейин Web да профессионал даражада ишлаш учун зарур барча инструментларга эга бўлинади.

PHP дастурлаш тилини ўрганишни бошлаш учун Apache виртуал серверини барча зарур модулларини, ҳамда MySQL маълумотлар базасини ўрнатиш ва созлаш лозим.

Бунинг энг содда ва қулай усули - Apache, MySQL, PHP лар бирлаштирилган XAMPP пакетини ўрнатишдир.

XAMPP — ўзида Apache, MySQL, PHP интерпретатори, Perl дастурлаш тили ва бир қанча қўшимча кутубхоналарни жамлаган кроссплатформали йиғилган Web –сервер.

XAMPP —бу :

- **X** (ихтиёрий тўртта операцион тизим),
- **A**раche,
- **M**ySQL,
- **P**HP,
- **P**erl.

XAMPP барча 32-разрядли Microsoft операцион тизимларида(98/2000/XP/2003/Vista/7), ҳамда [Linux](#), [Mac OS X](#) ва [Solaris](#) тизимларида ишлайди. Дастур [GNU General Public License](#) лизензиясига асосан барчага очик ва бепул тарқатилади. Бу дастур web-серверлар билан ишлаш, динамик саҳифалар яратиш учун қулай.

XAMPP ни ўрнатиш учун [zip](#), [tar](#) ёки [exe](#), форматидаги файлни юклаб олиш зарур, дастур компонентлари созлашни талаб этмайди. Дастур доимий тарзда таркибига [Apache/MySQL/PHP](#) ва [Perl](#) ларнинг янги версияларини олиб янгиланиб туради. Дастурнинг фойдаланувчи интерфейси содда тузилган. XAMPP ни ўрнатиш унинг компонентларини ҳар бирини алоҳида ўрнатишга нисбатан кам вақт олади.

Linux учун XAMPP

Linux тизимида тарқалиши(SuSE, RedHat, Mandrake ва Debian да тестланган), таркиби: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, eAccelerator, SQLite ва IMAP C-Client лардан иборат.

Windows учун XAMPP

Windows 98, NT, 2000 ва XP лар учун жорий этилган. Бу версия таркибига қуйидагиларни олади: Apache, MySQL, PHP + PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail

Transport System for Win32 and NetWare Systems v3.32, JpGraph, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, ва WEBDAV + mod_auth_mysql.

Windows учун XAMPP Lite

Windows 98, NT, 2000 ва XP лар учун жорий этилган. Бу Windows учун XAMPP нинг кичрайтирилган версияси ва у таркибига қуйидагиларни олади: Apache, PHP (PEAR сиз), MySQL, phpMyAdmin, OpenSSL, SQLite.

Огоҳлантириш: lite версияси учун ҳеч қандай қўшимчалари мавжуд эмас. XAMPP Lite – "Юклаш-Ишга тушириш-Ўчириш - Унутиш" пакети.

Ўрнатиш ва ўчириш

• XAMPP ни ўрнатишнинг энг осон усули — ўрнатгичдан фойдаланиш (38Mb).

• Ўрнатгичсиз ҳам ўрнатиш мумкин.

• Ўчириш учун барча серверлар ва XAMPP хизматлари ёпилиши талаб этилади ва XAMPP папкаси ўчирилади (агар XAMPP ўрнатгич ёрдамида ўрнатилган бўлса, дастурни ва реестридаги ёзувларни ўчириш учун ўчириш дастурини ишга туширилади).

Дастурий таъминотнинг актуал версияларидан бирини таркиби билан танишиб чиқамиз:

• XAMPP 1.7.2

- Apache 2.2.12 (IPv6 enabled) + OpenSSL 0.9.8k
- MySQL 5.1.37 + PBXT engine
- PHP 5.3.0
- phpMyAdmin 3.2.0.1
- Webalizer 2.21-02 + GeoIP lite
- FileZilla FTP Server 0.9.32
- msmtmp 1.4.17

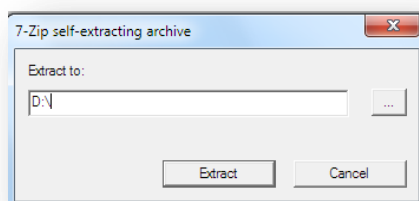
Дастурни ўрнатиш икки босқичда амалга оширилади: *биринчи босқич* - дастурни ўрнатиш, *иккинчи босқич* - дастурни тестлаш.

XAMPP дастурий таъминотини ўрнатиш

Биринчи босқич

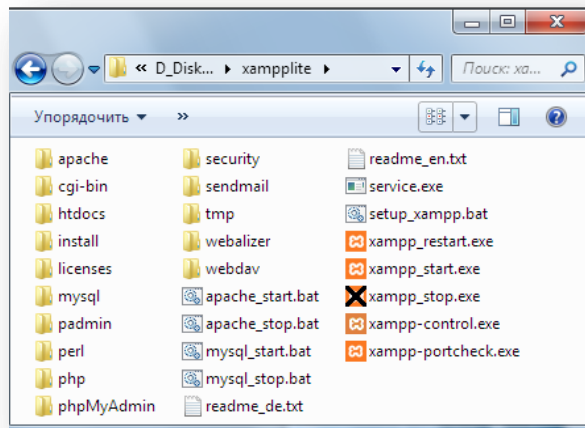
XAMPP ни ўрнатгичдан фойдаланиб ўрнатиш.

1. *Файллари ўзи ўрнатиловчи ҳажми кичикроқ 7-ZIP архив (Installer ва ZIP архивини ҳам юклаб олиш мумкин) дастурий таъминотини юклаб олинади. Дастурий таъминот юклаб бўлингандан сўнг, ехе файли ишга туширилади. Ҳосил бўлган ойнада ўрнатиш учун диск танланади « D: \»:*



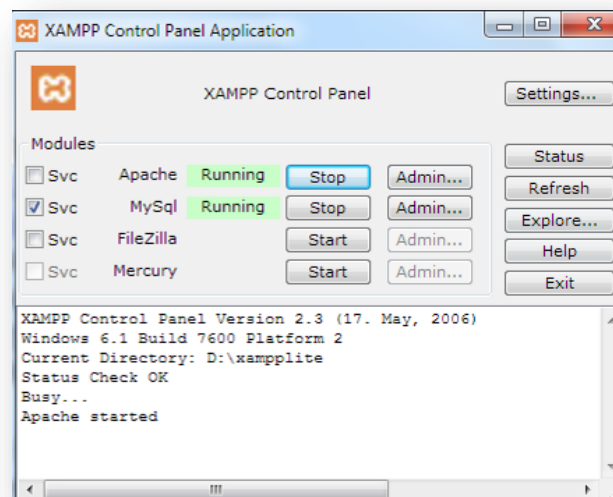
1.1-расм. XAMPP дастурий таъминотини ўрнатиш
Ўрнатгич барча файлларни ўрнатади.

2. Ўрнатгандан сўнг кўрсатилган манзил бўйича қидирилади(D:/xampplite).



1.2-расм. XAMPP дастурий таъминотини ўрнатилган папка

3. Серверни ишга тушириш ва ўчириш (отключить) учун xampp-control.exe ишга туширилади.



1.3-расм. XAMPP ДТ нинг бошқарув панели

Ўрнатгичсиз ўрнатиш.

Аввал ушбу (<http://www.apachefriends.org/en/xampp.html>) саҳифага кирилади ва юклаб олиш мумкин бўлган ушбу (<http://www.apachefriends.org/en/xampp-windows.html#641>) саҳифадан мос ўрнатгич юклаб олинади. Ҳажми тахминан 45 Мб.

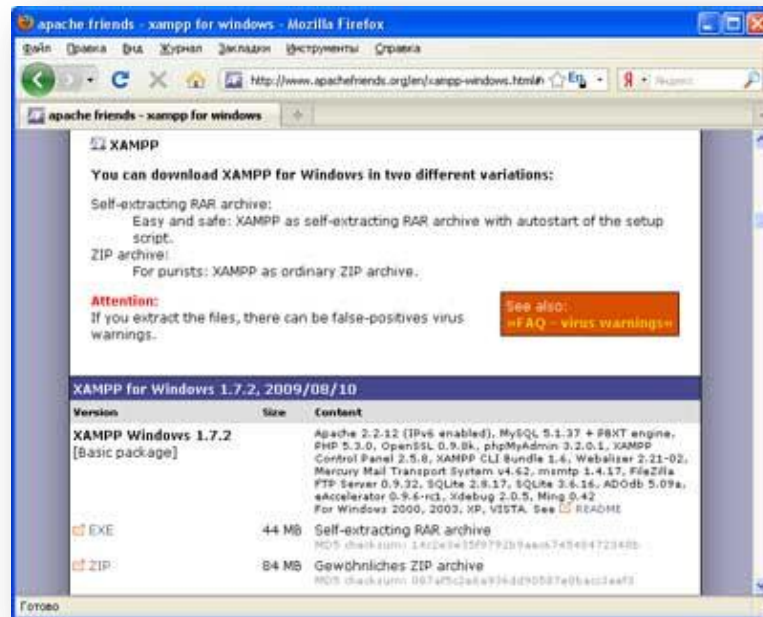
XAMPP Windows 1.7.2 [Basic package] ва XAMPP Lite 1.7.2.

Web -сервер версияси	XAMPP Windows 1.7.2 [Basic package]	XAMPP Lite 1.7.2
Кутубхоналар	Apache 2.2.12 (IPV6 enabled)	Apache 2.2.12 (IPV6 enabled)

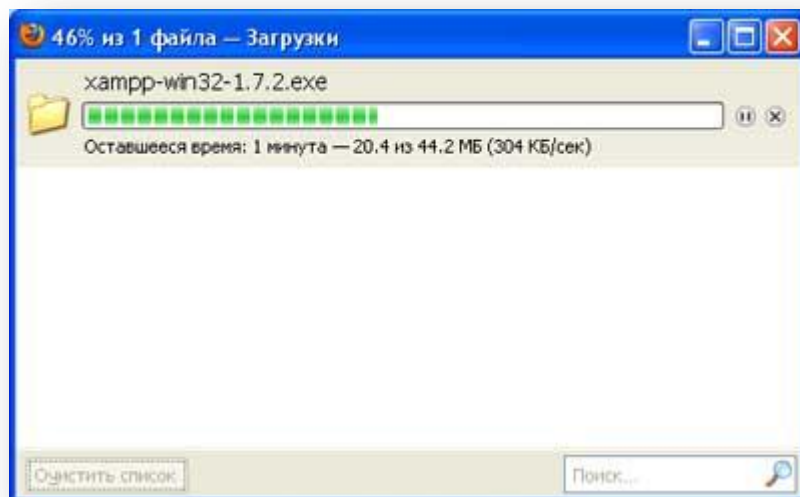
	MySQL 5.1.37 (Community Server) with PBXT engine 1.0.08-rc	MySQL 5.1.37 (Community Server) with PBXT engine 1.0.08-rc
	PHP 5.3.0 + PEAR (PEAR, Mail_Mime, MDB2, Zend)	PHP 5.3.0 + PEAR
	Perl 5.10.0 (Bundle::Apache2, Bundle::Apache::ASP, Bundle::Email, Bundle::DBD::mysql, DBD::SQLite)	Mini Perl 5.10.0
	XAMPP Control Version 2.5.8 (ApacheFriends Edition)	XAMPP Control Version 2.5.8 (ApacheFriends Edition)
	XAMPP CLI Bundle 1.6	XAMPP CLI Bundle 1.6
	XAMPP Port Check 1.4	XAMPP Port Check 1.4
	XAMPP Security 1.1	XAMPP Security 1.1
	SQLite 2.8.17	SQLite 2.8.17
	SQLite 3.6.16	SQLite 3.6.16
	OpenSSL 0.9.8k	OpenSSL 0.9.8k
	phpMyAdmin 3.2.0.1	phpMyAdmin 3.2.0.1
	ADODB v5.09a	—
	FPDF v1.6	—
	Zend Framework 1.9 Minimal Package (via PEAR)	—
	Mercury Mail Transport System v4.62	—
	msmtp 1.4.17 (a sendmail compatible SMTP client)	msmtp 1.4.17 (a sendmail compatible SMTP client)
	FileZilla FTP Server 0.9.32	—
	Webalizer 2.21-02 (with GeoIP lite)	Webalizer 2.21-02 (with GeoIP lite)
	Xdebug 2.0.5 for PHP	—
	eAccelerator 0.9.6-rc1 for PHP	—
	Ming 0.4.2 for PHP	Ming 0.4.2 for PHP
	PDF with pdflib lite v7.0.4p4 for PHP	PDF with pdflib lite v7.0.4p4 for PHP
Тизимга кўйилган талаблар	128 MB RAM	128 MB RAM
	256 MB free fixed disk	160 MB free fixed disk
	Windows 2000, XP (Server 2003), Vista (Server 2008)	Windows 2000, XP (Server 2003), Vista (Server 2008)
	all systems 32 bit (64 bit not tested)	all systems 32 bit (64 bit not tested)
Ҳажми	EXE 44MB ZIP 84MB	EXE 26MB ZIP 56MB
Юклаб олиш саҳифаси	http://www.apachefriends.org/en/xampp-windows.html#641	http://www.apachefriends.org/en/xampp-windows.html#646

Бу иккала пакетни ўрнатиш жараёни бир хил кечади. Ўзингизга мос пакетни танланг.

Дастурий таъминот юклаш жараёни:

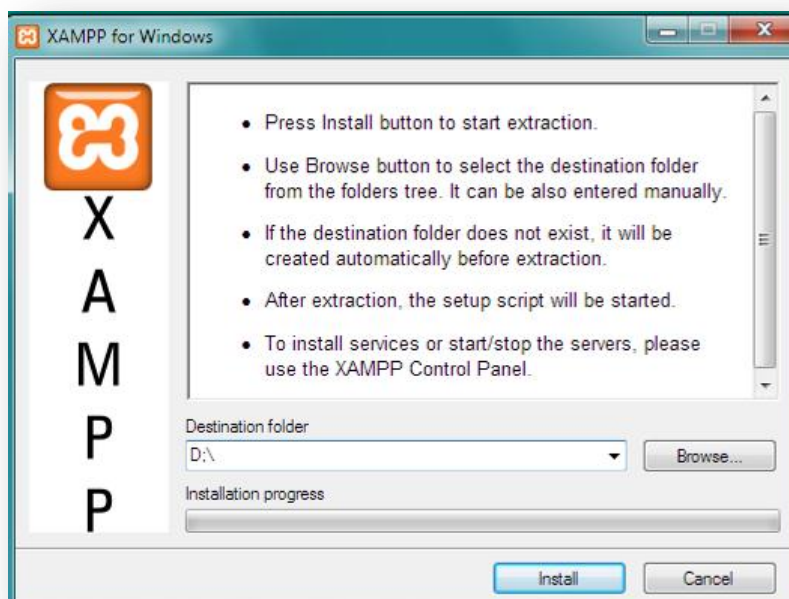


1.4-расм. XAMPP ни ушбу саҳифада юклаб олиш мумкин



1.5-расм. XAMPPни юклаб оламиз.

1. XAMPPни ўрнатишни бошлаймиз. Серверни жойлаштириш учун манзилни танлаш йўли кўрсатилган ойна очилади. Одатда дастур C:\ дискни кўрсатади. C:\ дискда одатда операцион тизим ўрнатилганлиги учун, биз D:\ дискни танлаймиз. Install тугмасини босамиз ва дистрибутивлар тўлиқ очилишини кутамиз.

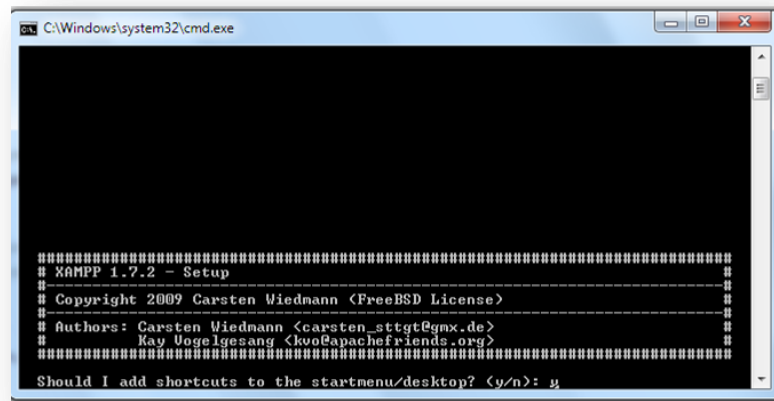


1.6-расм. XAMPP дастурий таъминотини ўрнатиш ойнаси



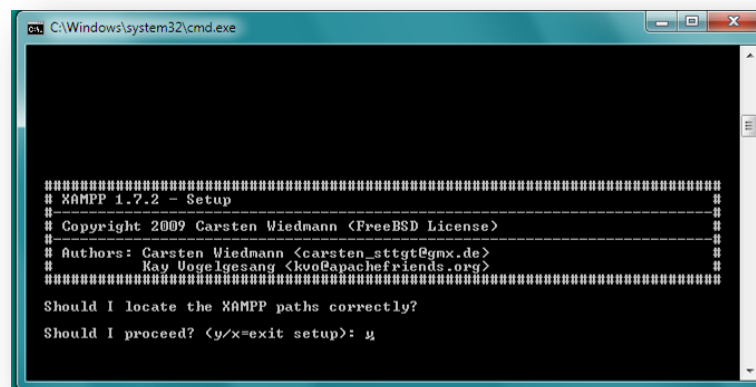
1.7-расм. XAMPP ДТнинг дистрибутивларининг очилиш жараёни

2. Дистрибутивлар очилгандан сўнг буйруқлар сатри ишга тушади, шундан сўнг ўрнатишнинг асосий қисми бажарилади. XAMPP иш столига ва пуск менюсига ёрлиқ яратишни таклиф қилади, *enter* ни босамиз.



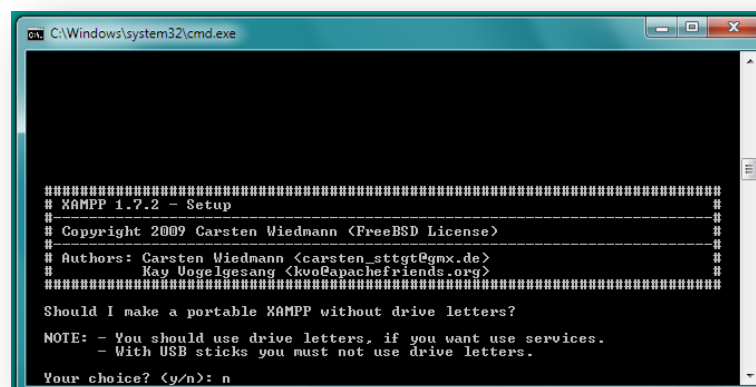
1.8-расм. XAMPP ДТнинг ёрлиқ яратилиш ойнаси

3. Бу ойнада XAMPPни ўрнатиш каталоги тўғри танланганлиги сўралади, одатга кўра сервер C:\xampp манзилига жойлаштирилади, enter ни босамиз.



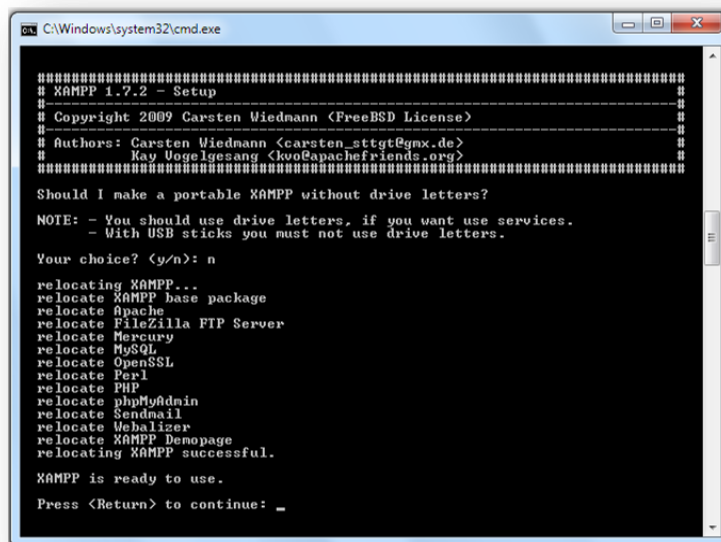
1.9-расм. XAMPP ДТнинг ўрнатиш каталогини тўғрилини текшириш ойнаси

4. Ихчам портатив версиясини ўрнатишни ҳохлайсизми? Enter ни босамиз,. XAMPP қаттиқ диска ўрнатилади.



1.10-расм. XAMPP ДТни ўрнатиш

5. XAMPPни ўрнатишни сўнгги босқичи. Давом эттириши учун enter ни босамиз.



```

C:\Windows\system32\cmd.exe

#####
# XAMPP 1.7.2 - Setup
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
# Authors: Carsten Wiedmann (carsten_sttgt@gmx.de)
# Kay Vogelgesang (kvo@apachefriends.org)
#####

Should I make a portable XAMPP without drive letters?
NOTE: - You should use drive letters, if you want use services.
      - With USB sticks you must not use drive letters.

Your choice? (y/n): n

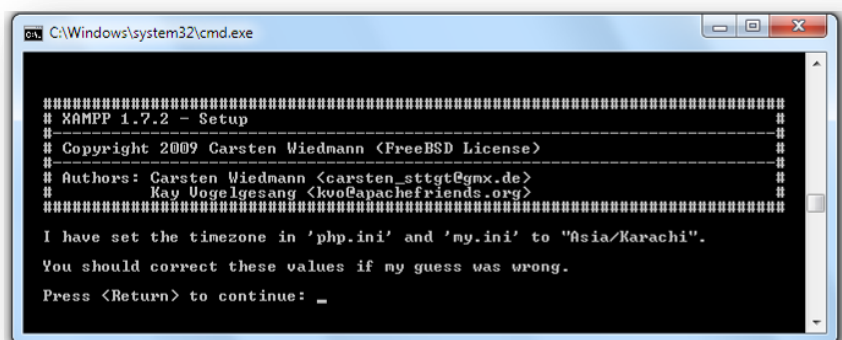
relocating XAMPP...
relocate XAMPP base package
relocate Apache
relocate FileZilla FTP Server
relocate Mercury
relocate MySQL
relocate OpenSSL
relocate Perl
relocate PHP
relocate phpMyAdmin
relocate Sendmail
relocate Webalizer
relocate XAMPP Demopage
relocating XAMPP successful.

XAMPP is ready to use.

Press <Return> to continue: _
  
```

1.11-расм. XAMPP ДТни ўрнатишнинг сўнгги босқичи

6. Соат вақтини автоматик созлаш. Enter ни босамиз.



```

C:\Windows\system32\cmd.exe

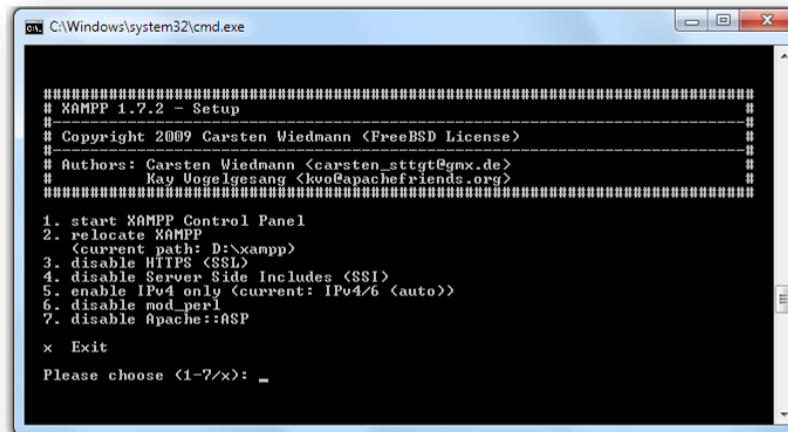
#####
# XAMPP 1.7.2 - Setup
# Copyright 2009 Carsten Wiedmann (FreeBSD License)
# Authors: Carsten Wiedmann (carsten_sttgt@gmx.de)
# Kay Vogelgesang (kvo@apachefriends.org)
#####

I have set the timezone in 'php.ini' and 'my.ini' to "Asia/Karachi".
You should correct these values if my guess was wrong.

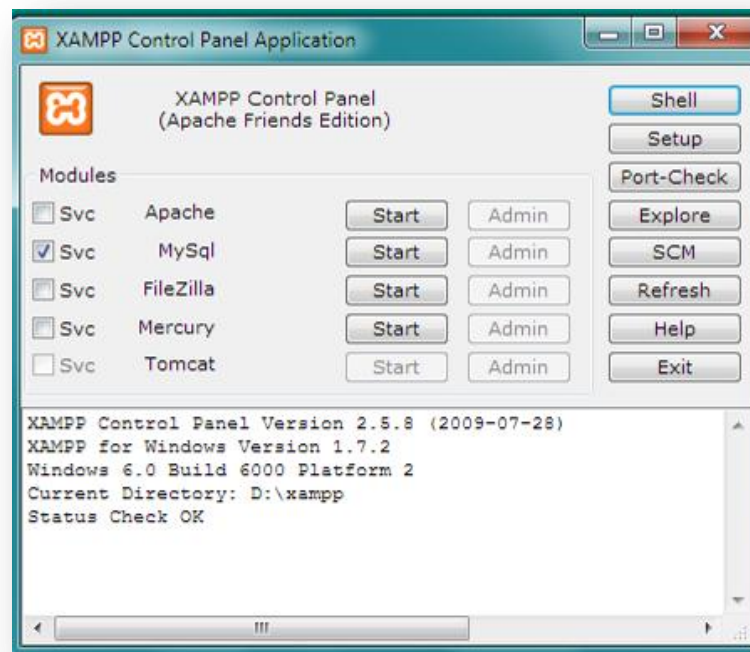
Press <Return> to continue: _
  
```

1.12-расм. XAMPPни соат вақтини созлаш ойнаси

7. XAMPP (XAMPP Control Panel)ни бошқарув панелини ишга туширамиз, 1ни танлаймиз ва enterни босамиз.



1.13-расм. XAMPPни бошқарув панелини ишга тушириш ойнаси
8. Серверни бошқарув панели ишга тушди. Буйруқлар сатрини ёнамиз.



1.14-расм. XAMPP ДТни бошқарув панели

Бошқарув панелидаги тугмаларнинг вазифалари:

Checkboxes (Svc) - Windows хизматчилари сифатида модулларни ўрнатади(агар сиз бирор бир модулни(Apache, MySQL ва ш.к.) хизматчи сифатида ишга туширмоқчи бўлсангиз, у ҳолда мос байроқчани белгилаймиз)

Admin – администратор модулларини ишга туширади (Apache модули қаршисидаги Admin – XAMPP ойнасини очади, тилни танлаймиз. XAMPP Web – интерфейсига кирамиз, бу ерда Demos бўлимида демонстрацион мисолларни текшириш ва модуллар ҳақида маълумот олиш мумкин. MySQL модули қаршисидаги Admin – phpMyAdmin ни ишга туширади)

Shell – сервер билан ишлаш учун буйруқлар сатрини ишга туширади;

Setup – XAMPP созлаш учун буйруқлар сатрини очади;

Port-Check – портларни текшириш;

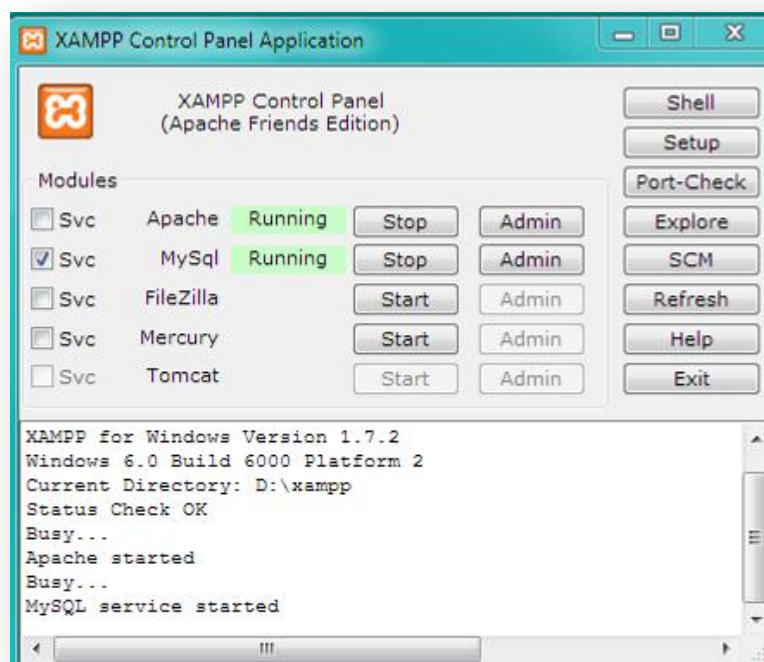
Explore – XAMPP ўрнатилган папкани очади, бизнинг ҳолда D:\xampp;

SCM – Windows хизматларининг бошқарув ойнасини очади;

Refresh – модуллер ҳолатини янгилаш(обновление);

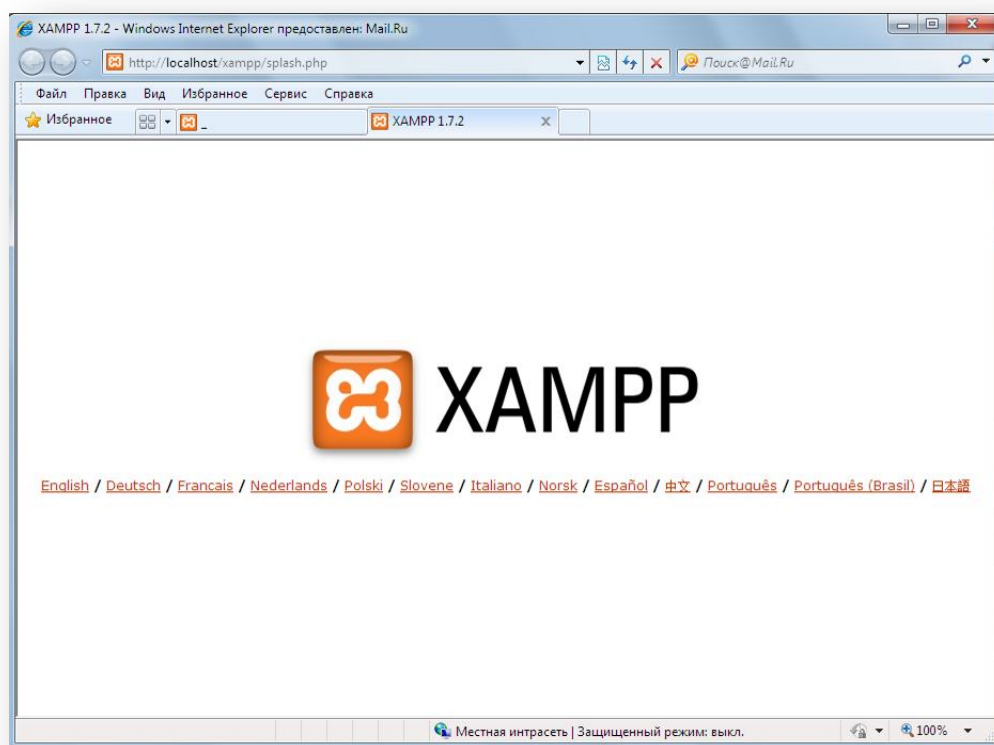
XAMPP ни иш столидаги ёки пуск менюсидаги ёрлик орқали ишга тушириш мумкин, агар сиз ёрлик яратишни бекор қилган бўлсангиз, XAMPP ни D:\xampp каталогидаги xampp-control.exe файли ёрдамида ишга туширишингиз мумкин.

9. *Apache* ва *MySQL* модуллерини ишга туширамыз(*Start* тугмаси), браузерда *localhost* ни терамиз (ёки *Apache* модули қаришисидаги *Admin* ни босамиз):



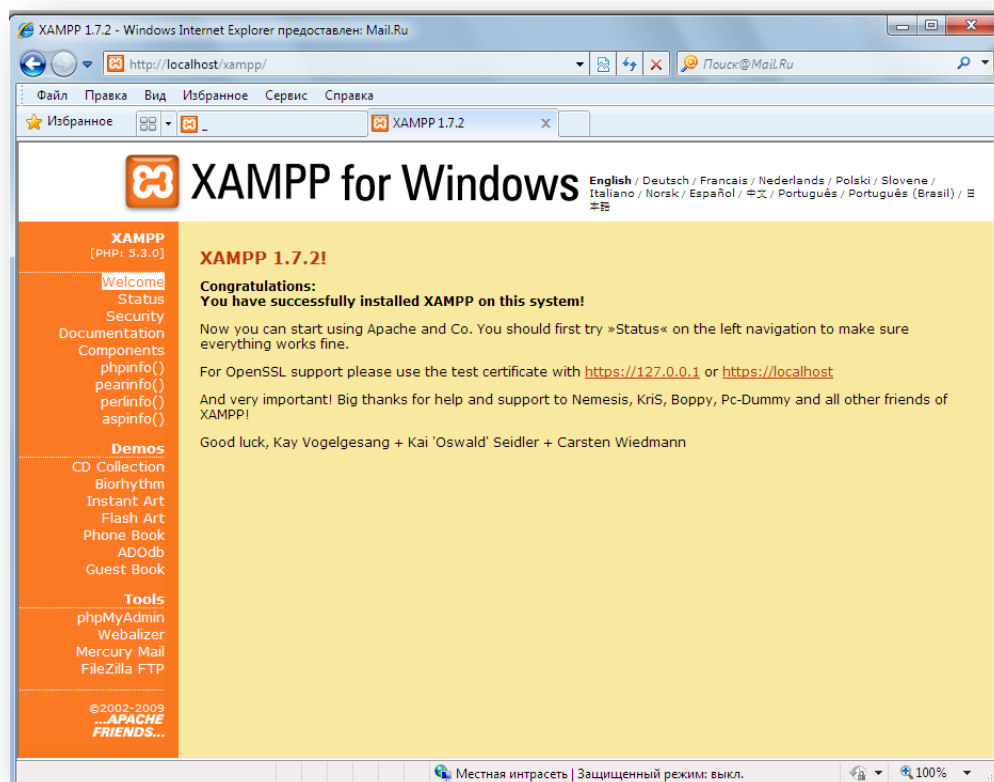
1.15-расм. *Apache* ва *MySQL* модуллерини ишга тушириш ойнаси

10. XAMPP нинг бош саҳифаси очилади, берилган тиллардан бирортасини танлаймиз:



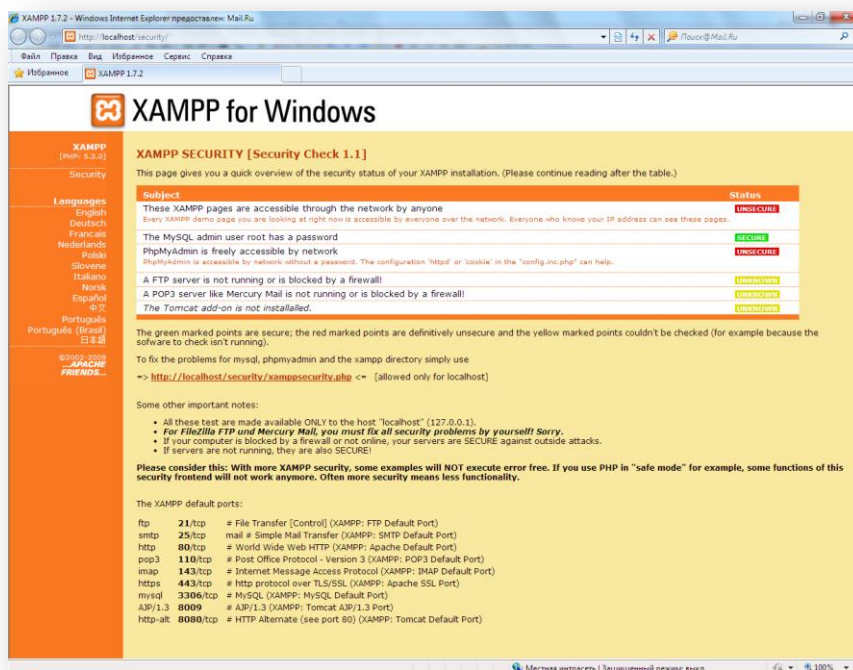
1.16-расм. XAMPP нинг бош саҳифаси

11. XAMPP нинг Web -интерфейси очилади, бу ерда Demos бўлимида турли хужжатлар ва уларнинг мисоллари билан танишишингиз мумкин ва б.:



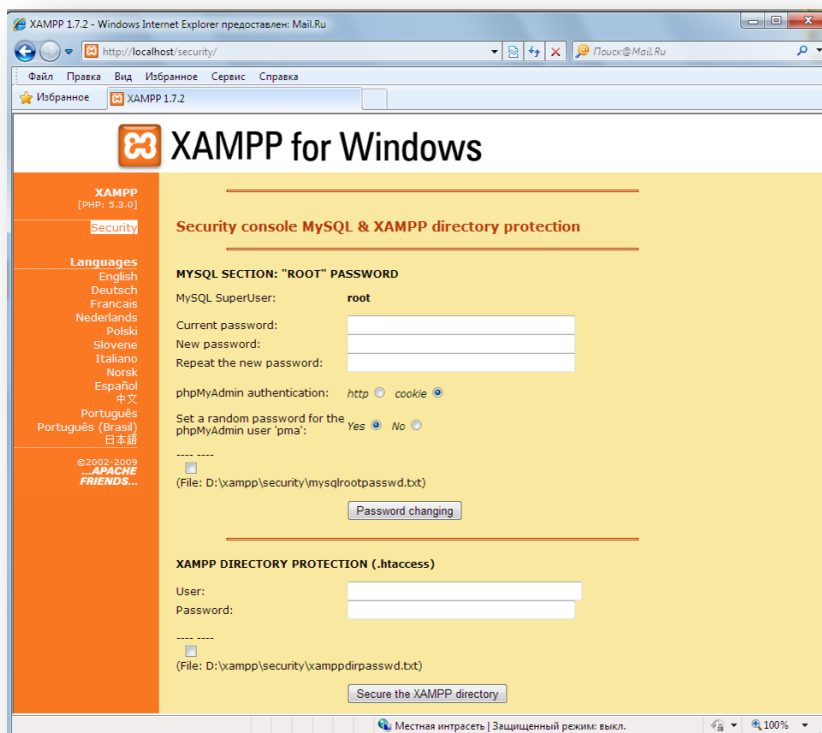
1.17-расм. XAMPP нинг Web интерфейси

12. Security бўлимида хавфсизликни созлаш қисмлари жойлашган:



1.18-расм. XAMPP нинг security бўлими

Ушбу `http://localhost/security/xamppsecurity.php`, кўрсатилган манзилга кириш, бу ерда MySQL маълумотлар базаси администратори учун парол бериш мумкин ва локал тармоқдан фойдаланишни чегаралаш учун XAMPP директориясига парол ўрнатиш мумкин:



1.19-расм. ХАМРР директориясига ва МБ си администраторига парол бериш ойнаси

Иккинчи босқич

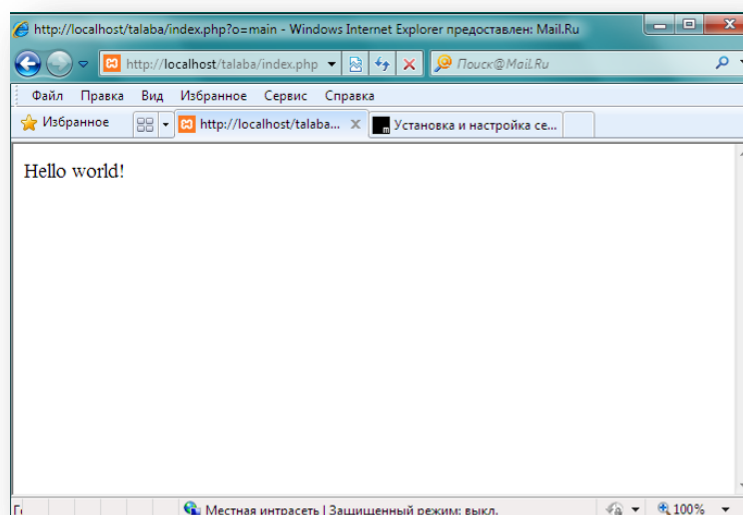
ХАМРР Web -сервери билан танишиб чикдик ва ўрнатдик, энди матнли саҳифа яратиш ва уни проектга жойлаштиришни кўриб ўтамыз.

1. Локал Web -серверда сайт яратиш учун, *D:\xampp\htdocs\ папкасида сайт номи билан боғлиқ бўлган ном билан папка яратамиз, масалан Talabalar portalі ни яратиш учун talaba папкасини яратамиз. Шу папкада матн муҳаррири ёрдамида php-скрипт ни ёзамиз:*

```
<?php
echo 'Hello world!';
?>
```

Ушбу файлни index.php деб сақлаймиз.

2. Браузерни ишга тушириб манзил қаторида `http://localhost/talaba/` манзилини ёзиб, саҳифани ишга туширамыз. Агар барча иш тўғри бажарилган бўлса, браузерда "Hello world!" ёзувини кўришимиз мумкин:



1.20-расм. Браузер ойнаси

1.4. Виртуал серверлар (хостлар) яратиш

Асосий конфигурацион файли `httpd.conf` - web-узелни бошқариш учун ишлатилади. У ёрдамида асосий амаллар (сервер локал ресурслар билан қандай ишлаш мумкин, сўровларга жавоб бериш, фойдаланувчи қайси файллар билан амалларни бажариши мумкинлиги) аниқланади.

У орқали Apache ни ишлашини бошқариш мумкин. Web-сервернинг конфигурацион файлини сошлаш – уни ўрнатишдаги энг муҳим ишлардан биридир. Сервер конфигурацион файлни ишга тушганда ўқийди. Агар сервер ишлаётган бўлса, у ҳолда файл конфигурациясини ўзгартиргандан сўнг уни қайта юклаш лозим.

`httpd.conf` файлида директивалар асосий учта бўлимга гуруҳланади:

Section 1. Global Environment – Apache нинг умумий ишини бошқарувчи директивалар бўлиб, Apache нинг асосий функцияларига таъсир кўрсатади;

Section 2. Main Server Configuration – сервернинг асосий параметрларини аниқловчи ва сўровларга жавоб берувчи директивалар. Бу директивалар виртуал серверлар учун ҳам аниқланиши мумкин;

Section 3. Virtual Hosts – виртуал серверларни ўрнатиш учун.

Виртуал серверлар (хостлар) – бу бир нечта web-серверларнинг турли IP-манзиллар билан битта нусхадаги Apache дастурини ишлатишидир.

Виртуал серверларни қўллаш доираси:

- турли ташкилотлар, фирмалар ва индивидуал фойдаланувчилар учун ўзларининг манзиллари билан алоҳида web-серверлар яратиш;

- фирма бўлимлари учун виртуал серверларни ташкил этиш, ҳар бир бўлим ўзининг домен номи ва web-серверига эга бўлсин;

- иккита виртуал сервер кўринишидаги умумий фойдаланувчили ва ички web-серверларни ташкил этиш;

- web-серверни қайта ишлаш ва текшириш учун виртуал серверларни ишлатиш.

Apache дастурининг httpd.conf файлида барча директивалар битта файлда httpd.conf нинг ўзида сақланади.

Xampp пакетининг httpd.conf файлидаги директивалар турли файлларга бўлиб сақланган бўлиб, улар httpd.conf файлига Include ёрдамида чақириб олинади.

```
# xampplite specific settings
Include conf/extra/httpd-xampp.conf

# Server-pool management (MPM specific)
# Include conf/extra/httpd-mpm.conf

# Multi-language error messages
Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
Include conf/extra/httpd-autoindex.conf

# Language settings
Include conf/extra/httpd-languages.conf

# User home directories
Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
Include conf/extra/httpd-info.conf

# Virtual hosts
Include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
# Include conf/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
Include conf/extra/httpd-dav.conf

# Various default settings
Include conf/extra/httpd-default.conf
```

```
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
```

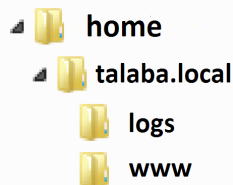
Қоидага асосан папкалар `D:\xampp\htdocs\` (масалан `talaba`) директорияси ичида яратилади ва шу папкага Web-саҳифа файллари жойлаштирилади, сўнг браузер ёрдамида `http://localhost/talaba/` мурожаатини ёзиб Web саҳифа ишга туширилади.

Бундан ҳам қулай йўли мавжуд бўлиб, Web саҳифалар `D:\xampp\htdocs\` да эмас, балки бошқа папкада сақланади ёки сайтнинг локал версиясига www.talaba.local манзили билан кирилади. Бунинг учун эса виртуал хостларни яратиш ва созлаш лозим.

1. *Виртуал хост яратиш учун аввал хост жойлашадиган папка яратилади:*

D: дискнинг home папкасида (D:\home) talaba.local номли папкани яратамиз, унинг ичида www ва logs папкаларини яратамиз.

Бу Windows файллар тизимида қуйидаги кўринишда бўлади:



2. *D:\xampp\htdocs\ да бўлган барча папкаларни D:/home/talaba.local/www га ўтказамиз.*

3. *apache нинг конфигурацион файлига ўзгартириш киритамиз.*

Матн муҳарририда D:\xampp\apache\conf\extra каталогда жойлашган жойлашган httpd-vhosts.conf файлини очамиз.

httpd-vhosts.conf файлининг таркибига ҳеч қандай ўзгартириш киритмаймиз, фақатгина янги хост қўшамиз (қуйидаги код нусхасини олинг ва файлдаги кодларнинг охирига қўйинг):

```
NameVirtualHost *:80

<VirtualHost *:80>
    DocumentRoot "D:\xampplite\htdocs"
    ServerName localhost
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@talaba.local
    DocumentRoot "D:\home\talaba.local\www"
    ServerName talaba.local
    ServerAlias www.talaba.local
    ErrorLog "D:\home\talaba.local\logs\error.log"
    CustomLog "D:\home\talaba.local\logs\access.log" combined
    <Directory "D:\home\talaba.local\www">
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Юқоридаги бошқа ранг билан белгиланган виртуал хост localhost орқали кириш учун имкон яратади. (XAMPP локал сайти).

4. Браузер виртуал хостларни аниқлаши учун *C:\Windows\System32\drivers\etc\hosts* манзилида жойлашган *hosts* файлига иккита қатор қўйиш талаб этилади (сервер папкасида бунинг ёрлигини ҳам яратишни тавсия этамиз):

```
127.0.0.1    talaba.local
127.0.0.1    www.talaba.local
```

5. Керакли файлларни *D:\home\talaba.local\www* манзилига кўчирамыз ёки *index.html* файлини яратамиз.

```
<html>
<head>
<title>Talabalar portali</title>
</head>
<body>
"Hello world!"
</body>
</html>
```

6. XAMPP бошқарув панелига (XAMPP Control Panel) кирамыз ва *Aracheni* ишга туширамыз. *www.talaba.local* манзили орқали ишлаши керак бўлган сайтимизни текиширамыз. Агар барчаси тўғри бажарилган бўлса, браузерда "Hello world!" хабарни кўришимиз мумкин.

Шунга каби бир қанча виртуал хостларни қўйиш мумкин. *httpd-vhosts.conf* файлига қўшилган сатрларни ёритиб чиқамиз:

- *ServerAdmin webmaster@talaba.local* - e-mail-Web-сервер администратори манзили;
- *DocumentRoot "D:\home\talaba.local\www"* - путь к папке с сайтом;
- *ServerName talaba.local* — браузерга киритиладиган сайт номи, ихтиёрий қийматни олади, масалан *talaba* ёки *talaba.local*;
- *ServerAlias www.talaba.local* — сайт очилиши керак бўлган манзил;
- *ErrorLog "D:\home\talaba.local\logs\error.log"* — хатолар журнали;
- *CustomLog "D:\home\talaba.local\logs\access.log" combined* — сайтдан фойдаланиш журнали, ким ва қачон сайтга кирди, қайси браузердан кабилар ёритилган.

II БОБ. HTML

Бу бобда HTML да Web саҳифа яратиш, матнлар билан ишлаш, матнларни форматлаш, график тасвирлар яратиш; фреймлар ёрдамида Web саҳифаларни ташкил этиш, гипермурувжаатлар, жадваллар, стиллар билан ишлаш ҳақида сўз боради. Бундан ташқари HTML нинг турли элементлари билан танишишингиз мумкин.

2.1. Web саҳифа яратиш ва матнлар билан ишлаш

Windows муҳитидаги матн муҳаррирлари: Notepad, TextPad, UltraEdit, EdutPlus. Кўрсатилган барча матн муҳаррирларини HTML кодларни ёзишда қўллаш мумкин.

Кўпгина матн муҳаррирларида HTML кодларни хаттоки, дастурлаш тилларида ҳам кодларни ёзиш мумкин, масалан, RHPEDIT. Баъзилари автоматик равишда операторлар, функцияларни таниш ва уларни ҳар-хил рангларда тасвирлаш имкониятига эга. Баъзи матн муҳаррирларида HTML ҳужжатни Web браузерда синаб кўриш тугмаси мавжуд.

HTML ҳужжатни яратишга мўлжалланган махсус дастурлар (HTML муҳаррирлар) ҳам мавжуд: FrontPage, Adobe GoLive, Macromedia Dreamweaver, Nestcape Composer. Муҳаррирлар 2 турга бўлинади:

- код муҳаррирлари;
- WYSIWYG технологияси(What You See Is What You Get – нимани кўрсанг шуни оласан) асосида ишлайдиган муҳаррирлар. Бу муҳаррирлар ёрдамида фойдаланувчи HTML командаси ва элементларини ёзмайди, оддий матн муҳаррирларидек матн ёзади, тасвирларни керакли жойга жойлаштиради, форматларни танлайди ва ҳ.

Web анимациялар турлича бўлади. масалан *GIF Animator* дан фойдаланиб, тасвирларни «Жонлантириш» мумкин. Бу Web рекламаларда кўп қўлланилади.

GIF анимацияларни Ulead GIF Animator (<http://www.ulead.com>) ва Animagic GIF Animator (<http://www.rtlsoft.com/animagic/index.html>) дастурлари ёрдамида ҳосил қилиш мумкин. GIF тасвирлардан юқорироқ ўринда Macromedia Flash ёрдамида яратилган анимациялар туради. Бу анимациялар интерактив бўлганлиги сабабли Flash технологиялар оммалашиб бораёпти. Бу технологияни айниқса автомобил ишлаб чиқарувчилар, web-бизнесчилар, умуман, мультимедиа – презентация яратмоқчи бўлганлар кенг қўллайдилар.

Macromedia (<http://www.macromedia.com/flash/>) дан ташқари CoffeeCup Firestarter (<http://www.coffeecup.com>) дан ҳам фойдаланиб анимациялар яратиш мумкин.

Агар фойдаланувчи мультимедиа саҳифасини яратмоқчи бўлса, қўшимча инструментлардан фойдаланишга тўғри келади, масалан, Quick Time, Windows Media ёки бошқа мультимедиа дастурлари.

Microsoft компанияси Windows ME ва ундан кейинги чиқарган ОС ларида Windows Movie Maker дастурини қўшиб чиқараяпти. Бу дастур

ёрдамида видео роликларни таҳрирлаш мумкин бўлади. Яна Sound Forge XP (<http://www.soniefoundry.com>) дастуридан товушларни таҳрирлаш ва конвертация қилишда фойдаланиш мумкин.

Web саҳифалар яратишда файлларни жойлаштиришнинг қуйидаги усуллари мавжуд:

- **Ҳамма файллар битта каталогда.** Бир нечта HTML саҳифали ва бир нечта тасвирларга эга кичик сайтларни яратишда ҳамма файлларни битта каталогда жойлаштириш мумкин бу усулнинг қулайлиги шундаки, ўзаро мурожаатларда ҳеч қандай йўл кўрсатиш шарт эмас. Ёмон томони эса, сайт ривожланиб, кенгайиб борса файллар кўпайиб ҳаммаси аралашиб кетади.

- **Функционал каталоглар.** Мураккаб сайтларни яратишнинг бир усулидир. Ҳар бир каталогда системанинг битта функцияли қисми жойлаштирилади. Энг юқорида бошланғич файл – index.htm, ва унга тегишли тасвирлар бўлади. Каталоглар номлари тахминан қуйидагича бўлиши мумкин: “Biz_haqimizda”, “Aloqalar”, “Mahsulotlar”. Ҳар бир каталогда тегишли HTML файллари ва тасвирлар сақланади.

- **Файл турлари бўйича каталоглар.** Бу усулда ҳар бир каталогда бир хил турдаги файллар сақланади. Бу ердаги файллар қайси функцияни бажариши муҳим эмас. Ташқи каталогда фақат index.htm файли жойлашади. Каталоглар номлари тахминан қуйидагича бўлиши мумкин: “images”, “aloqalar”, “downloads”,...

Ихтиёрий файлнинг кенгайтмаси унинг муҳим қисмидир. Чунки браузер файлларни унинг кенгайтмасига қараб “танийди”.

Одатда биринчи саҳифа ҳар доим index.htm, index.html каби сақланади. Кўпчилик браузерлар агар URL аниқ кўрсатилмаса бу файлларни автоматик юклашга ҳаракат қилади. Масалан, браузернинг адреслар панелига <http://www.microsoft.com> деб ёзилса <http://www.microsoft.com/index.htm> файлини юклайди.

Файлни сақлашга унга мантиқий ном танлаш керак бўлади. Файл номидан уни бошқа файллардан осон ажрата олиш керак. Масалан, about_inst.htm, fiz_mat.htm, rector.jpg каби ном қўйиш мақсадга мувофиқ. Notepad ni ochamiz. HTML faylni yaratishni boshlaymiz.

Сарлавҳа ёзиш учун:

```
<head>
</head>
```

теглари ёзилади.

Браузер бу теглар ўртасидаги матнни сарлавҳа деб тушунади ва браузернинг энг тепа қисмига шу матнни ёзади. Энди саҳифанинг танасини ҳосил қиламиз:

```
<body>
</body>
```

Бу теглар ўртасига ҳамма матн ва тасвирларни жойлаштириш керак. HTML шаблон бўлиши учун HTML саҳифанинг сарлавҳа ва танасини ўз ичига олган қуйидаги зарур теглар етишмаяпти:

```
<html></html>
```

Демак HTML шаблон қуйидаги кўринишга эга бўлди:

```
<html>
<head>
</head>
<body>
</body>
</html>
```

<head> ва </head> теглари ўртасига қуйидаги тегларни жойлаштириш мумкин:

<title>, </title> – ҳужжат номи. Масалан:

```
<title> TDPU ning Web saxifasi </title>
```

<meta /> – маълумотлар ҳақидаги маълумот. Яъни бу ерда асосий терминлар ёзилади. Қидирув системалари ишлаганда айнан шу терминлар бўйича кидириш ишини олиб боради, сайтларни топади. Масалан:

```
<meta name="keywords" content="TDPU, Nizomiy, Pedagogika,
Universiteti, Fakultet, Student"/>
```

content 50-200 та сўзни ўз ичига олиши мумкин.

name=("keywords", "autor", "copyright", "description") – махфий кидиришда қўлланилади, HTTP серверга доступ бериш учун http-equiv қўлланилади.

<body> ва </body> теглари ўртасига саҳифанинг танаси жойлаштирилади. Бу соҳада браузер ойнасида фойдаланувчи кўриши керак бўлган матн, гипермуружаат, тасвирлар, жадваллар ва ҳ. жойлаштирилади.

<p>, </p> теглари улар орасидаги матнни алоҳида абзац кўринишида ёзилишини таъминлайди. Масалан:

```
<p> Bu yangi abzats</p>
```

<p> </p> – бўш абзац жойини ташлайди.

Агар ёзувни янги абзацдан эмас, янги қатордан бошлаш керак бўлса,
 тегидан фойдаланиш мумкин. Масалан:

```
<p> Toshkent shaxri<br />
Yu.X.Xojib ko'chasi 103 uy<br />
Tel:252-74-38<br />
```

2.2. Матнларни форматлаш. Web саҳифаларда графика

Сарлавҳалар <p> теги каби алоҳида абзац кўринишида ифодаланади. Сарлавҳаларнинг 1 дан 6 гача даражалари мавжуд.

```
<h1 align="center">Eng katta sarlavxa</h1>
<h2>2-darajali kattalikdagi sarlavxa</h2>
<h3>3-darajali kattalikdagi sarlavxa</h3>
<p>Bu oddiy matn</p>
<h4>4-darajali kattalikdagi sarlavxa</h4>
<h5>5-darajali kattalikdagi sarlavxa</h5>
<h6>Eng kichik sarlavxa </h6>
```

Ҳар хил ажратувчи горизонтал чизиқлар <hr /> теги ёрдамида қўйилади. Бу чизиқ абзацлар оралиғида жойлашиб, браузер ойнасининг эни бўйлаб чизилади. Агар фойдаланувчи браузер ойнасининг ўлчамларини ўзгартирса, чизиқ ҳам мос равишда ўзгаради.

<hr /> теги
 теги каби очик тег ҳисобланади, яъни бу тегларни ёпувчи теги бўлиши талаб қилинмайди.

HR – “horizontal rule” (горизонтал чизиқ) дир. <hr /> тегининг атрибутлари:

Атрибут номи	Вазифаси
Size	Горизонтал чизиқнинг қалинлиги. Энг катта ўлчам 5 пикселни қўллаш тавсия этилади.
Width	Чизиқнинг энини белгилайди, фоизларда берилади.
Align	Чизиқнинг жойлашиши: браузер ойнасининг чап томонида, ўнг томонида, ўртасида.
noshade	Чизиқнинг ташқи кўринишини белгилайди.

масалан:

```
<hr size="4" align="center" width="50%" noshade="noshade" />
```

Стиллар икки хил бўлади: физик стиллар ва мантикий стиллар.

Матн муҳаррирлари билан ишлаш жараёнида биламизки, матнларни ҳар-хил кўринишда ифодалаш мумкин: қалинлаштирилган(полужирный), кўлёзма шаклида(курсив), таги чизилган(подчеркнутый) ... бу элементларни ихтиёрий график браузерлар бир хил кўринишда ифодалядилар. Баъзи бир физик стиллар:

Элемент	Вазифаси
, 	Қалинлаштирилган матн (полужирный)
<i>, </i>	Кўлёзма шаклидаги матн(курсив)
<tt>, </tt>	Ҳарфлар оралиғини кенгайтириш
<u>, </u>	Таги чизилган матн(подчеркнутый)
<big>, </big>	Катталаштирилган матн
<small>, </small>	Кичиклаштирилган матн
_,	Пастки индекс
[,]	Юқори индекс

```

<b> Qalinlashtirilgan matn (polujirniy)</b><br />
<i> Qo'lyozma shaklidagi matn (kursiv) </i><br />
<tt> Harflar oralig'ini kengaytirish </tt><br />
<u> Tagi chizilgan matn (podcherknutiy)</u><br />
<big> KATTALASH TIRILGAN MATN</big><br />
<small> kichiklashtirilgan matn</small><br />
C<sub>n</sub></sub><br />
ax<sup>2</sup>+bx+c=0<br />

```

Мантикий стиллар:

Элемент	Вазифаси
, 	<i> тегига аналог тег.
, 	 тегига аналог тег.
<cite>, </cite>	Мазкур хужжатга цитата келтириш.
<dfn>, </dfn>	Дастур коди
<samp>, </samp>	Дастурнинг ишлашига мисол. Олдингиси каби ишлайди.
<kbd>, </kbd>	Клавиатурадан киритиладиган матн
<var>, </var>	Ўзгарувчи ёки микдор
<abbr>, </abbr>	Аббревиатура
<acronym>, </acronym>	Акроним

Мантикий стиллар браузерга матнни қай тарзда экранга чоп этиш кераклигини билдиради. Мантикий стиллар физик стиллар ишламай қоладиган ҳолатлар учун ҳам ўринли бўлиши мумкин: уяли телефон интернетга уланганда тегі орқали қалинлаштирилган матн учраб қолса, уни ўқий олмайди. Бу ҳолатда элементи керакли натижани бера олади.

<blockquote> элементи физик атрибутлар қаторига киради. Бу элемент абзаци визуал ҳолда оддий матндан ажралиб туришини таъминлайди. масалан:

```

<blockquote>
Diqqat e`lon! Dushanba kuni kafedra majlisi bo`lib o`tadi
</blockquote>

```

<address> элементи саҳифа автори ҳақидаги маълумотни ўзида мужассамлаштирган матнни яратиш учун ишлатилади. Аксарият браузерларда бу матн қўлёзма кўринишига эга бўлади.

масалан,

```

<address>
web master: tdpu@tdpu.uz <br />
Toshkent shaxri, Y.X.Xojib kuchasi 103 uy.
Tel: 252-74-38
</address>

```


Рўйхат тури номерланган рўйхат ва чизиқли рўйхат бўлиши мумкин. Номерланган рўйхат ``, чизиқли рўйхат эса `` элменти билан берилади. Уларнинг ҳар бири `` элементи билан ифодаланади.

```
<ol>
<li>Birinchii punkt</li>
<li>Ikkinchi punkt</li>
<li>Uchinchi punkt </li>
</ol>
<ul>
<li>Birinchii punkt</li>
<li>Ikkinchi punkt</li>
<li>Uchinchi punkt</li>
</ul>
```

Кўпгина график браузерларда тартибланган рўйхатларнинг кўшимча атрибутларини қўллаш мумкин бўлади. Уларнинг ёрдамида тартиблашни кўринишини танлаш мумкин бўлади.

Тег номи	Рўйхат тури
<code><ol type="A"></code>	Катта ҳарфлардан тузилган рўйхат
<code><ol type="a"></code>	Кичик ҳарфлардан тузилган рўйхат
<code><ol type="I"></code>	Рим (Катта) ҳарфлардан тузилган рўйхат
<code><ol type="i"></code>	Рим (кичик) ҳарфлардан тузилган рўйхат
<code><ol type="I"></code>	Араб рақамларидан тузилган рўйхат
<code><ul type="disc"></code>	Доиралардан тузилган рўйхат
<code><ul type="square"></code>	Квадратлардан тузилган рўйхат
<code><ul type="circle"></code>	Айланалардан тузилган рўйхат

Бундан ташқари тартиблашни ихтиёрий **start** атрибути ёрдамида рўйхатни бошланғич қийматини танлаш мумкин бўлади. Масалан, рўйхат 5 номеридан бошланиши керак бўлса, `<ol start="5">` деб ёзиш керак.

Тармоқда тасвирлар ҳақида гапирилганда аниқланган маълум бир турдаги файллар ҳақида гап кетади. У цифровой камера ёки сканер ёрдамида олинган расм, тасвир, ҳаттоки, график форматда ифодаланган матн бўлиши ҳам мумкин. Бу ерда файлнинг аниқланган формати муҳим рол ўйнайди. Кўпчилик матн муҳаррирлари (Notepad, SimpleText) график файлларни ўзларида тасвирлай (кўра, ифодалай) олмайдилар. Бунинг учун махсус тасвирларни кўриш дастури, график муҳаррирлар мавжуд.

Тасвирларни web-саҳифага жойлаштириш жуда оддий. Бунинг учун `` элементи керак бўлади. Шунини тушуниш керакки, расм саҳифага жойлаштирилмайди, балки расмга саҳифадан туриб мурожаат (расмга йўл)

кўрсатилади. Расм файли эса компьютернинг каттиқ дискида ёки тармоқда жойлашган бўлиши мумкин.

Тармоқда тасвирларнинг асосан JPEG (Joint Photographic Experts Group «Dji-peg» deb o'qiladi), GIF (Graphics Interchange Format «gif» yoki «djif» deb o'qiladi) va PNG (Portable Network Graphics «ping» deb ўқилади) каби форматлари қўлланилади. Бундан ташқари PCX, TIFF ва PICT каби форматлар ҳам интернетда учрайди. Агар фойдаланувчида бошқа форматдаги тасвирлар бўлиб, уни web-саҳифага жойлаштирмоқчи бўлса, у ҳолда уни юқоридаги форматлардан бирида ифодалаб олиши мақсадга мувофиқдир.

 элементи Web саҳифага тасвир қўйиш имкониятини беради. Бунда тасвирнинг URL адресини кўрсатиш керак бўлади. Тасвир файли ҳамма файллар сақланаётган каталогларнинг бирида бўлади ёки Интернетда жойлашган бўлиши мумкин. элементи қуйидаги кўринишда ёзилади:

```

```

Агар тасвир интернетдан юкланиши режалаштирилган бўлса, у ҳолда унинг тўлиқ адресини ёзиш керак:

```

```

Агар тасвир ўзимизнинг каталогда жойлашган бўлса, унинг адресини тўлиқ кўрсатиш шарт эмас:

```


```

alt атрибутида саҳифага қўйилаётган тасвир экранга юклаш имконияти бўлганда шу тасвир нима ҳақида эканлигини билдирувчи матн жойлашади. Бу матн тасвир чиқиши керак бўлган, аммо баъзи бир сабабларга кўра экранда кўринмай турган жойда ёзилади. Масалан:

```

```

Матн ва тасвирларни саҳифага жойлаштиришда қуйидаги атрибутлардан фойдаланилади:

Номи	Вазифаси
align="top"	Матн тасвирнинг юқори чегараси билан тенглаштириб ёзилади.
align="middle"	Матн тасвирнинг ўртаси билан тенглаштириб ёзилади.
align="bottom"	Матн тасвирнинг қуйи чегараси билан тенглаштириб ёзилади.

Матнларни ўнгдан ва чапдан текислаш имконияти ҳам мавжуд. Бунинг учун **align** атрибутида **left** ёки **right** ёзувини ёзиш керак.

Қўйилаётган тасвирнинг ўлчамини ҳам бериш мумкин. Бунинг учун width ва height элементлари ишлатилади. Берилаётган ўлчам пикселларда ифодаланади. масалан:

```

```

2.3. Гипермуружаат (гиперссылкалар) лардан фойдаланиш

Гипермуружаатлар web-сайтлар бўйлаб ҳаракатнинг асоси ҳисобланади. Муружаатни танлаганда фойдаланувчи браузер ойнасига юкланувчи ёки ёрдамчи дастурни ишга туширувчи қанақадир URL билан боғланган адресга «тушиб» қолади. Баъзан гипермуружаат натижаси e-mail ёки FTP серверга йўлланма берувчи янги web-саҳифани очилишига олиб келади. Фойдаланувчи муружаатни танлаб олиши учун web-дизайнер уни яратиши керак.

Гипермуружаат яратиш учун `<a>` (anchor, yakor) элементида фойдаланилади. У ўзида йўлланма берувчи URL адресни кўрсатувчи **href** атрибути билан тўлдирилади. Шунинг учун гипермуружаатни яратиш учун URL адресни аниқлаб олиш керак.

URL бу браузер адреслар қаторида ёзиладиган web-саҳифа ёки хизматнинг тўлиқ адресидир. Кўпинча бу адрес гипермуружаатнинг устига курсор олиб келинганда браузернинг пастки қаторида кўрсатилади.

Гипермуружаатни яратишда агар Интернетдаги хизмат ёки адресдан фойдаланмоқчи бўлсак албатта унинг тўлиқ адресини кўрсатиш шарт. Агар ўзимизда бор бўлган web-саҳифалардан гипермуружаатлар яратмоқчи бўлсак баъзи бир ишни осонлаштирувчи ҳолатлар мавжуд:

1. Фараз қилайлик биз <http://www.tdpu.uz> асосли каталоглар структурасини яратмоқчимиз.

2. Бу каталогнинг ичида images, about, fakultet каби каталоглар жойлаштирмоқчимиз.

3. fizmat каталогида (fakultet каталоги ичида) kurs1.html саҳифасини яратдик.

4. Энди about каталогида жойлашган contact.html саҳифасига муружаат қўймоқчимиз. Бунинг учун қуйидагича ёзилади:

```
../..../about/contact.html
```

5. Афсуски, бу ёзув <http://www.tdpu.uz/about/contact.html> ёзуви каби қулай бўлмаган ёзувдир. Бундай узун муружаатларни ёзганда турли хил хатоликка йўл қўйиш мумкин. Бунинг олдини олиш учун `<base>` элементи ишлатилади. Бу элемент ёрдамида асосий база сифатидага каталог танланади:

```
<head>
<base href="http://www.tdpu.uz" />
</head>
```

Бу элемент ёрдамида юқоридаги муружаатни оддийгина қилиб «about/contact.html» кўринишида ёзиш мумкин. `<base>` элементи Интернетдаги адресларга қўйилган муружаатларга ҳалақит бермайди.

HTML ҳужжатда муружаатларни яратиш учун якор (`<a>`, ``) элементида фойдаланилади. Бу теглар муружаатни билдирувчи сўзни ўз ичига олади. Муружаат қуйидаги кўринишда ёзилади:

```
<a href="Murojaat_manzili">Murojaat matni</a>
```

Демак, «Ma'lumotlar» сўзи билан мурожаат about каталогидаги about.html саҳифага мурожаат қўйиш учун қуйидагича ёзиш керак бўлади:

```
<a href="http://www.tdpu.uz/about/about.html">Ma'lumotlar </a>
<a href="about/about.html" >Ma'lumotlar </a>
```

Ички мурожаатлар жорий саҳифанинг ўзига қўйилади. Масалан, бирор фандаги бирор мавзуга бағишланган саҳифа яратаётган бўлайлик. Мавзудаги мавжуд ҳамма режаларга мурожаатларни шундай қўйиш мумкинки, мурожаатни танлаганда режадаги маълумотлар экранда пайдо бўлади. Бу ишни қилиш оддий. Яъни ўтилиши керак бўлган ҳамма абзацларни белгилаб чиқиш, ундан кейин ўша белгиларга якор (<a>,) ни қўйиб чиқиш керак. Мисол қарайлик:

```
<h2 align="center">Reja </h2>
<p><a href="#q1">1. HTTP</a></p>
<p><a href="#q2">2. HTML</a></p>
```

Энди мурожаат қилинаётган абзацини белгилаб оламиз:

```
<p><a name=q1>HTTP bu...</p> yoki
<p><a name=q2>HTML bu...</p>
```

Бундан ташқари бошқа web-саҳифадаги бирор абзацга ҳам мурожаат ўрнатиш мумкин:

```
<p>Bu <a href=" ../index.html#q1">savol</a><p>ga javob bering
```

Тасвирларга ҳам гипермуурожаатларни қўйиш мумкин:

```
<a href="pedagog/index.html" ></a>
```

Тасвирларнинг ўлчами ҳар хил бўлиши мумкин, ҳар қандай тасвирга гипермуурожаатларни қўйиш мумкин.

URL шундай универсал ва мослашувчанки ундан Тармоқдаги ҳар хил мурожаатлар (e-mail, FTP, Gopher, Usenet yangiliklari, Telnet sessiyalari) дан фойдаланиш мумкин. Бу нарса битта саҳифанинг ўзида турли хил мурожаатлардан фойдаланиб, универсал сайт яратиш имкониятини беради. Масалан, дастурни юклаш билан бирга фодаланувчи ўша дастур ҳақидаги янгиликларни Usenet янгиликлар сервери мурожаатидан фойдаланиб танишиши мумкин.

mailto. mailto мурожаатини ҳосил қилиш оддий. Бунинг учун тўрт қисмдан иборат бўлган мурожаат қўйиладиган электрон адресни билиш кифоя: fodaluvchi_nomi, @ simvoli, kompyuter nomi, server nomi. mailto mуроjaati:

```
<p><a href="mailto:tdpu@tdpu.uz" title="TDPU mamuriyati">Ixtiyoriy
savol bilan bizga murojaat qilishingiz mumkin</a></p>
```

FTP. FTP протоколи компьютерлар ўртасида файллар алмашишда қўлланилади. FTP мурожаатларини ҳосил қилиш учун керакли сервер номини билиш кифоя:

```
<a href="ftp://ftp.microsoft.com/">Microsoft
kompaniyasining FTP sayti</a>
```

```
<a href="ftp://ftp.jspi.uz/downloads/program.zip"> Zip langan  
programma</a>
```

Иккинчи мисолдан кўриниб турибдики, керакли дастурни биратўла юклаб олиш мумкин. Бунда браузер олинаётган дастурни каттиқ дискка ҳақиқатан ҳам сақлаш керак ёки керакмаслигини фойдаланувчидан сўрайди.

Gopher. Gopher матнлар ёрдамида иерархик кўринишдаги маълумотларни браузерда ифодалашдир. Ҳозирги кунда Gopher сайтлари жуда ҳам кам миқдорда ишлатилади. Уларга бўлган мурожаатларни қўйиш:

```
<a href="gopher://marvel.loc.gov">Kongress kutubxonasi </a>
```

Usenet. Usenet янгиликлар гуруҳи – Интернет конференциялар гуруҳидир. Унинг номи янгиликлар бўлгани билан бу гуруҳнинг асоси ҳар бир қатнашувчи иштирок этиши мумкин бўлган савол-жавоблар, муҳокамалардан иборатдир. Бу гуруҳга сизнинг сайтингизда қандайдир муаммо муҳокама қилиниши керак бўлганда мурожаат қилиш мумкин. Ёки фойдаланувчилар томонидан сизга шунақа кўп саволлар бериладики, сизнинг уларга жавоб беришга имкониятингиз йўқ, шунда бу гуруҳга мурожаат қилишингиз мумкин, қаердадир сизга ёрдам берадиган инсон топилади:

```
<a href="news:alt.tx.startrek">Starrek ga bagishlangan  
Usenet konferentsiyasi</a>
```

Telnet. Telnet серверига мурожаат – Дастурий Таъминот билан таъминланган узоқдаги компьютер билан алоқа ўрнатиш учун мўлжалланган мурожаатдир. Telnet технологиясини бирорта ҳам замонавий браузерлар қўлламайди, шунинг учун бу мурожаатни танлаганда қўшимча дастурлардан фойдаланишга тўғри келади:

```
<a href="telnet://mac1.fakecorp.com/">Telnet serveri bilan aloqa  
o'rnatish</a>
```

Мурожаатларни яратишда баъзи бир кичик айёрликлардан фойдаланиш ҳам мумкин. Масалан мурожаатни танлаганда янги ойна очилиши учун **target** атрибутидан фойдаланилади. Бу атрибут қўлланилганда натижа янги ойнада ҳосил бўлади.

```
<a href="http://www.tdpu.uz" target="_blank"> TDPU sayti</a>
```

Агар сайтдаги барча мурожаатлар янги ойнадан очилишини ҳоҳласангиз уни **<base>** элементи ичида жойлаштирганингиз маъкул:

```
<base href="http://www.tdpu.uz" target="_blank" />
```

Натижада сизнинг web-саҳифангиздага ҳамма гипермурожаатлар янги ойнада ҳосил бўлади.

2.4. Фреймлар ёрдамида сайт яратиш.

Узоқ вақтлар web-саҳифалар яратувчилари бутун сайт хаттоки портал саҳифалари бўйлаб ҳаракатнинг умумий системасини топишга ҳаракат қилдилар. Фреймлар кириб келиши билан боғлиқ энг асосий муаммо ҳар доимдагидек браузерларнинг ўзаро туғри келмаслиги бўлди. Кўпгина браузерлар фреймларни «танямасди». Фреймлар спецификацияси HTML стандартларига 1997 йилда яратилган HTML 3.2 версиясидан бошлаб қўлланила бошланди. У вақтларда фреймлар Netscape фойдаланувчилари орасида оммавийлашган эди. HTML 4.0 дан бошлаб браузерлар фреймларни «тушуна» бошладилар.

Фреймлар битта браузер ойнасига бир нечта турли URL адресларга эга бўлган мустақил web-саҳифаларни юклаш имкониятини беради. Бу вазифани бажариш жуда оддий. Бунинг учун **<body>** элементи **<frameset>** элементи билан алмаштирилади. Бу контейнор саҳифада фреймлар ҳосил қилувчи **<frame />** учун мўлжалланган. Фреймлар ичида қайси фреймга чиқиши кўрсатилган алоҳида гипермуружаатлар бўлиши мумкин.

Демак саҳифада фреймлар ҳосил қилиш учун иккита **<frameset>** ва **<frame />** элементлари ишлатилар экан:

```
<html>
<head>
</head>
<frameset>
</frameset>
</html>
```

<frameset> контейнори бир ўзи ҳеч нарса қила олмайди. Экранда фреймлар ҳосил бўлиши учун **<frame />** ва **<noframes>** элементлари ишлатилиши керак. Биринчиси экранда фреймлар ҳосил бўлиши учун ишлатиласа, иккинчиси аксинча фреймларни олиб ташлаш учун ишлатилади:

```
<frameset>
<noframes>
```

<p> Uzoq vaqtlar web-sahifalar yaratuvchilari butun sayt hattoki portal sahifalari bo'ylab harakatning umumiy sistemasini topishga harakat qildilar. **</p>**

```
</noframes>
</frameset>
```

<noframes> элементи ёрдамида умуман фреймларсиз саҳифа яратиш ҳам мумкин.

<frame /> элементига мисол:

```
<frameset>
<frame src= "index.html" />
<frame src= "viewer.html" />
</frameset>
```

<frame /> элементи бирор бир фреймга итиёрий web-саҳифани чақириш учун ишлатилади. Саҳифанинг URL адреси **src** атрибути ёрдамида берилади.

Адреслар гипермуружаатлардагидек тўлиқ ёки қисқартирилган ҳолда берилиши мумкин.

Бирор бир фреймларга янги саҳифаларнинг юкланиши икки босқичдан иборат бўлади:

- ❖ Биринчидан, фреймга аниқ ном берилиши керак;
- ❖ Иккинчидан, кўрсатилган фреймга юкланиши учун юқорида берилган номдан фойдаланиш керак.

Масалан:

```
<frameset>
<frame src= "index.html" />
<frame src= "viewer.html" name= "doc_viewer">
</frameset>
```

Биринчи фрейм номланмади, чунки унга янги web-саҳифа юкланиши шарт эмас. Иккинчи фреймга янги web-саҳифани юклаш учун одатдаги **<a>** элементининг **target** атрибути ишлатилади:

```
<a href= "story.html" target= "doc_viewer">Тарих</a>
```

Энди чап томондаги гипермуружаат натижаси янги ойнага эмас, ўнг томондаги фреймга чиқади. **<a>** элементига ихтиёрий URL адресни қўйиш мумкин:

```
<a href= http://www.yandex.ru target= "doc_viewer">Yandex qidiruv
sistmasi </a>
```

<frame /> элементи қуйидаги параметрларга эга:

- ❖ **Noresize**. Фреймлар ўлчамини ўзгартиришни ман қилади.
- ❖ **Frameborder**. Фақат 1 ёки 0 қийматлардан биттасини қабул қилади. Агар 1 қийматни қабул қилса, фрейм чегараси мажуд, 0 қиймат қабул қилса, фрейм чегараси йўқ.

❖ **Scrolling**. **yes**, **no** ёки **auto** қийматларидан фақат биттасини қабул қила олади. **Yes** – прокрутка бор, **no** – прокрутка йўқ, **auto** – агар эҳтиёж бўлсагина прокрутка қўйилади.

❖ **marginwidth, marginheight** – бу атрибутлар горизонтал ва вертикал майдонларга мўлжалланган.

❖ **Longdesc**. Мазкур фрейм ҳақидаги маълумотлар сақланувчи URL адресни ўзида сақлайди.

Ичма-ич фреймлар. Агар экранда горизонтал ва вертикал фреймларни бирданига ҳосил қилмоқчи бўлсак, ичма-ич фреймлардан фойдаланишимиз мумкин.

<frameset> элементи иккита **cols** ва **rows** атрибутларига эга. Улар ёрдамида **<frameset>** саҳифаларни бир нечта қаторлар ёки устунлардан иборат фреймларга ажратади:

```
<frameset cols= "25%, 75%">
</frameset>
```

Натижада web-саҳифа иккита устунли фреймга ажрайди.

<frameset> элементи ёрдамида саҳифада ҳоҳлаганча устун ва қатор фреймларини ҳосил қилиш ҳам мумкин. Уларнинг ўлчамини доим ҳам

процентларда ифодалаш шарт эмас. Ўлчов бирлиги сифатида пикселларни ҳам олиш мумкин :

```
<frameset rows= "10, 200, 300, 100">
```

Натижада тўртта қатор фреймлари ҳосил бўлади.

Бундан ташқари ўлчамларни ифодалашда «*» ни ҳам ишлатиш мумкин. Бунда «*» га қанча ўлчам қолса ҳаммаси тегишли бўлади:

```
<frameset cols= "200, 300, *">
```

Процентларда ифодаланган ўлчамлар учун ҳам «*» ни ишлатиш мумкин:

```
<frameset rows= "25%, 35%, *">
```

Масалан тепада тўлиқ сатрли фрейм, унинг пастида икки устунга ажратилган фреймларни ҳосил қилайлик. Биринчидан бизга икки қаторли фрейм керак бўлади. Иккинчидан иккинчи қатор фреймни икки устунга ажратиш керак:

```
<frameset rows= "100, *">
<frame src= "banner.html" scrolling= "no" noresize = "noresize" />
<frameset cols= "25%, 75%">
<frame src= "index.html" />
<frame src= "viewer.html" marginwidth= "5" marginheight= "5"/>
</frameset>
```

Фреймларнинг классик вариантыни кўриб ўтамиз: браузер ойнаси учта қисмдан ташкил топган бўлсин: logo.html, menu.html, asosiy.html. Уларнинг жойлашиш ўрни турлича бўлиши мумкин, масалан,

LOGO	LOGO		L O G O	M E N U	ASOSIY	LOGO	ASOSIY
ASOSIY	MENU	ASOSIY				MENU	
MENU							

Қуйидагича вариантдаги фреймни яратамиз:

LOGO
ASOSIY
MENU

```
<html>
<head>
</head>
<frameset rows="100,*,150">
<frame src="logo.html">
<frame src="asosiy.html">
<frame src="menu.html">
```

```
</frameset>
</html>
```

Коддаги `rows="100,*,150"` ёзуви, биринчи қаторнинг узунлиги 100 пиксел, учинчи қаторнинг узунлиги 150, иккинчи қатор эса қолган майдонни эгаллашини англатади. `rows` ўрнида `cols` ишлатилса браузер ойнасини қаторларга эмас, балки устунларга бўлиш кераклигини билдиради.

Қатор ва устунларнинг ўлчамлари нафақат пикселлардан балки, фоизларда ҳам берилиши мумкин:

```
<frameset rows="15%,75%,10%">
```

Шуни ёдда тутиш керакки, барча ўлчамларнинг йиғиндиси 100% га тенг бўлиши керак.

Энди бошқа кўринишдаги фреймларни кодини ёзамиз. Масалан,

LOGO	
MENU	ASOSIY

```
<html>
<head>
</head>
<frameset rows="100,*">
<frame src="logo.html">
<frameset cols="150,*">
<frame src="menu.html">
<frame src="asosiy.html">
</frameset>
</frameset>
</html>
```

Охирги вариантдаги фрейм кодини ёзамиз:

LOGO	ASOSIY
MENU	

```
<html>
<head>
</head>
<frameset cols="100,*">
<frameset rows="100,*">
<frame src="logo.html">
<frame src="menu.html">
```

```

</frameset>
<frame src="asosiy.html">
</frameset>
</html>

```

Фреймларни юқорида келтирилганлардан бошқа қўшимча имкониятлари ҳам мавжуд. Масалан, бирор фреймга саҳифани юклаш ёки фреймларни янги ойнага очиш.

Target атрибути қуйидаги қийматларни қабул қилиши мумкин:

❖ **_self**. Мурожаат қўйилган фрейм ўзида ҳужжат очилишини таъминлайди.

❖ **_top**. Бу қийматни қўллаб, шу ойнанинг ўзида фреймларни ўчириб ҳужжатни юклаш мумкин.

❖ **_blank**. Ҳужжатни янги ойнада очилишини таъминлайди.

<iframe> элементи <frame /> элементи билан тўғридан-тўғри боғланган, лекин <frameset> элементига умуман алоқаси йўқ. <iframe> элементи HTML ҳужжатда ички мустақил фреймлар яратиш имкониятини беради. <iframe> элементи ихтиёрий саҳифада <body> нинг ичига жойлашиши мумкин. Унинг вазифаси саҳифада бошқа бир ҳужжатни <frame /> <frameset> да кўрсатгани каби кўрсатишдир.

2.1 - мисол:

```

<iframe src="jad.html" width="300" height="300" frameborder="0"
scrolling="auto">
  Mana sizga freym!!!
</iframe>

```

<iframe> элементи <frame /> элементининг **frameborder**, **marginwidth**, **marginheight**, **scrolling** каби ҳамма атрибутларини қабул қилиши мумкин. Булардан ташқари унинг **width** ва **height** атрибутлари ҳам мавжудки, улар қўйилаётган фреймнинг бўйи ва эни ўлчамларини пикселларда белгилайди. Яна <iframe> элементи **align** атрибутига эга бўлиб, у одатдагидек, **right** ва **left** қийматларини қабул қилади. <iframe> элементининг яна бир хусусияти шундаки, у ўз ичига олган матнни қачонки фойдаланувчи браузер ички фреймларни қўлламасагина экранга чиқаради.

2.5. Жадвал яратиш

Жадвални ҳосил қилишнинг асосий элементи `<table>` ҳисобланади. Ишлатилиши шарт бўлмаган элемент `<caption>` дир. `<table>` нинг ичида `<tr>` – жадвал қаторини билдирувчи ва `<td>` жадвал устунини билдирувчи элементлар жойлашади. Устун ёки қатор сарлавҳасини `<th>` элементи ёрдамида бериш мумкин.

Жадвални яратиш технологияси:

1. Жадвал ҳосил қилишни бошлаш учун `<table>` ва `</table>` элементлари ёзилади.

2. Бу элементлар ўртасига керакли миқдорда қаторлар сонига мос `<tr>` ва `</tr>` элементлари ёзилади.

3. Энди биринчи `<tr>` дан сўнг керакли миқдорда `<td>` ва `</td>` элементлари ёзилади. Ҳар бир `<td>` ва `</td>` элементлари ўртасида жадвалнинг мос қийматини киритилади. `<td>` ва `</td>` элементлари ўрнига `<th>` ва `</th>` элементларини ишлатиш ҳам мумкин. Бу элементлар одатда фақат биринчи қатор элементларини ёзишда ишлатилади. Кўпгина график браузерлар `<th>` ва `</th>` лар ўртасидаги матнни қорайтирилган (жирный) курунишда тасвирлайдилар.

4. Ҳар бир `<tr>` элементида кейин юқоридаги учинчи қадам такрорланади.

Оддий жадвалга мисол:

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

Учта устун ва иккита қатордан иборат жадвал яратилиш керак бўлсин:

```
<table border="1">
<caption>2ta satr va 3ta ustunli jadval</caption>
<tr>
<td>1x1</td>
<td>1x2</td>
<td>1x3</td>
</tr>
<tr>
<td>2x1</td>
<td>2x2</td>
```

```
<td>2x3</td>
</tr>
</table>
```

Жадвал номини `<caption>` элементи ёрдамида берилиши айтиб ўтилди. Бундан ташқари бу элементни ИЗОҲ сифатида ишлатиш ҳам мумкин. Бунинг учун `<caption align="bottom">` деб ёзилса жавдалнинг пастки қисмида чиқувчи изоҳга айланади.

Қатор элементларида иккита муҳим атрибутлар мавжуд: **align** va **valign**. Улар ёрдамида жадвал ячейкасидаги матнни горизонтал ва вертикал ҳолатда ихтиёрий жойлаштириш мумкин. **align** элементи **center**, **left**, **right** (ячейкадаги матнни ўртадан, чапдан, ўнгдан текислаш), **valign** атрибути эса **top**, **bottom**, **center** (ячейкадаги матнни тепадан, пастдан, ўртадан текислаш) каби атрибутларга эга:

```
<table border="1" align="center">
<tr align="center">
<th>Yillar</th>
<th>Choy</th>
<th>Kofe</th>
</tr>
<tr align="left">
<td>2001</td>
<td>35%</td>
<td>65%</td>
</tr>
<tr align="right">
<td>2002</td>
<td>45%</td>
<td>55%</td>
</tr>
<tr align="center">
<td>2003</td>
<td>55%</td>
<td>45%</td>
</tr>
</table>
```

Баъзи ҳолларда, масалан, жадвал элементлари бир хил, ўхшаш маълумотларга эга бўлса **valign** атрибутининг натижаси кўринмай қолиши мумкин. Лекин ячейкаларда ҳар хил ҳажмдаги маълумотлар сақланадиган бўлса, унинг натижасини кўриш мумкин.

Бундан ташқари `<th>` элементининг `<scope>` атрибути мавжуд бўлиб у сарлавҳа қатор ёки устунга тегишли эканлигини билдиради. Масалан:

```
<table border="1" align="center">
<tr align="center">
<th scope="row">Yillar</th>
```



```

<td>Choy</td>
<td>Kofe</td>
</tr>
<tr align="left">
<th scope="row">2001</th>
<td>35%</td>
<td>65%</td>
</tr>
<tr align="right">
<th scope="row">2002</th>
<td>45%</td>
<td>55%</td>
</tr>
<tr align="center">
<th scope="row">2003</th>
<td>55%</td>
<td>45%</td>
</tr>
</table>

```

Шундай ҳоллар бўладики, жадвалнинг бир нечта ячейкаларини (бир нечта устунни ёки бир нечта сатрни) бир хил ўлчамда ифодалашга тўғри келади (бир-бирига яқин информациялар ёки жадвалнинг чиройли чиқиши ...). Ҳар қандай ҳолатда ҳам <th> ва <td> лар учун: **colspan** ва **rowspan** атрибутлари қўлланилади. Улардан бири ячейкани горизонтал ўлчамини иккинчиси вертикал ўлчамини билдиради.

Жадвал орқасидаги фонни рангини танлаш имконияти ҳам мавжуддир. Бунинг учун **bgcolor** атрибути қўлланилади:

```

<table border="1" align="center">
<tr align="center" bgcolor="red">
<th>Yillar</th>
<th>Choy</th>
<th>Kofe</th>
</tr>
<tr align="left" bgcolor="brown">
<td>2001</td>
<td>35%</td>
<td>65%</td>
</tr>
<tr align="right" bgcolor="orange">
<td>2002</td>
<td>45%</td>
<td>55%</td>
</tr>
<tr align="center" bgcolor="yellow">

```

```
<td>2003</td >
<td >55%</td>
<td>45%</td>
</tr>
</table>
```

<table> элементи **align** ва **border** атрибутларидан бошқа атрибутларга ҳам эга. Улар куйидагилардир:

1. **width**. Жадвалнинг энини ўрнатади. Икки ҳолатда ўлчамини бериш мумкин: процентларда (`width="50%"`) ва пикселлар (`width="250"`) да. Бу атрибут берилмаганда жадвалнинг ичидаги маълумотлар ўлчамига қараб жадвал чизилади. Ўлчамларни пикселларда бериш тавсия этилмайди. Чунки ҳамма фойдаланувчилар ҳам экраннинг ўлчамини ҳар доим ҳам бир хилда ифодаламайдилар (800x600, 1024x768 каби ўлчамлар мавжуд).

2. **border**. Жадвал рамкаси қалинлигини танлаш имкониятини беради. Ўлчамлар пикселларда белгиланади (`border="3"`). Бу атрибут берилмаганда жадвал чегараси оқ рангда ифодаланади (кўринмайди).

3. **cellspacing**. Жадвал устунлари орасидаги масофани билдиради. Ўлчамлар пикселларда берилади. Атрибут қўлланилмаганда 1 га тенг деб қабул қилинади.

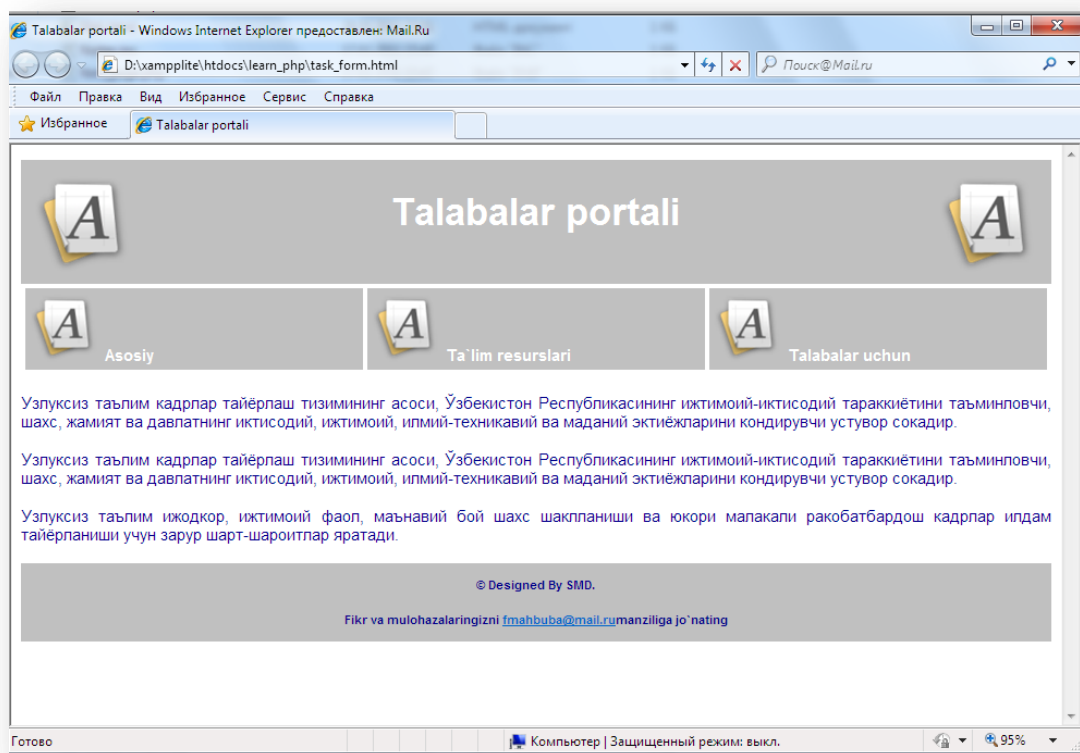
4. **cellpadding**. Жадвал қаторлари орасидаги масофани билдиради. Ўлчамлар пикселларда берилади. Атрибут қўлланилмаганда 1 га тенг деб қабул қилинади.

5. **rules** ва **frames**. Жадвалнинг ички (`rules`) ва ташқи (`frames`) чегаралари қандай чизилиши кераклигини билдиради.

Жадваллар HTML стандартларига киргандан бошлаб дизайнлар яратиш профессионализм даражаси бир неча маротаба ортди. Жадвал ячейкаларини бошқариб, уларга ихтиёрий элементларни – тасвирларни, матнларни, рўйхатларни, гипермуножаатларни қаерда керак бўлса ўша ерда ишлатиш имконияти мавжуд бўлади. Шу тариқа бутун web саҳифаларни қамраб олувчи жадвалларни яратиш яхши натижа беради. Ячейкаларни умумлаштириш, керакли ўлчамгача кенгайтириб, тораитириш, улар ораликларини белгилаш орқали юқорида айтилган натижага эришиш мумкин.

Умумий ҳолда жадваллар устунлар ва қаторлардан иборат бўлиб, улар устида ишлаш имконияти мавжуддир. Газета ёки журналларни варақлаб кўплаб катакларга бўлинганлигини кўриш мумкин. Худди шундай қилиб web-саҳифаларни ҳам катакларга бўлиш мумкин. Агар саҳифа бир нечта бўлакларга бўлинса у ўқувчи томонидан осон тушунилади. Масалан, куйидаги кўринишдаги web-саҳифани яратмоқчи бўлайлик:

2.2 – мисол.



2.1 – расм. Жадвал яратишга мисол

Бунинг учун қуйидаги кодни ёзишга тўғри келади:

```
<!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "logo.png "></td>
<td>
<h1>Talabalar portali</h1>
</td>
<td align = right><img src = "logo.png "></td>
</tr>
</table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
<tr >
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Ta`lim resurslari</span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Talabalar uchun</span></td>
</tr>
</table>
<!-- page content -->
```

<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Республикасининг ижтимоий-иқтисодий тараккиётини таъминловчи, шахс, жамият ва давлатнинг иқтисодий, ижтимоий, илмий-техникавий ва маданий эктиёжларини кондирувчи устувор сокадир. </p>

<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Республикасининг ижтимоий-иқтисодий тараккиётини таъминловчи, шахс, жамият ва давлатнинг иқтисодий, ижтимоий, илмий-техникавий ва маданий эктиёжларини кондирувчи устувор сокадир.

</p>

<p>Узлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шаклланиши ва юкори малакали ракобатбардош кадрлар илдам тайёрланиши учун зарур шарт-шароитлар яратади.</p>

<!-- page footer-->

<table width = "100%" bgcolor = silver cellpadding = 12 border =0>

<tr>

<td>

<p class=foot>© Designed By SMD.</p>

<p class=foot>Fikr va mulohazalaringizni fmahbuba@mail.rumanziliga jo`nating </p>

</td>

</tr>

</table>

Худди шунингдек, менюлар учун учта устун ҳосил қилиб, барча устунларга 25% дан бериб, бўлиш мумкин. Асосий саҳифани ўзини ҳам иккига бўлиб, чап меню ҳосил қилиш мумкин.

Жадвалнинг **colspan** атрибути қаторидаги устунларни умумлаштириш имкониятини беради. Бу ерда colspan=«n» ёзуви нечта қаторни умумлаштириш керак бўлса, n шунга тенг деб ёзилади.

Жадвалнинг **rowspan** эса бир нечта қаторларни умумлаштириш имкониятини беради. Нечта қатор умумлаштирилиши керак бўлса rowspan=«n» деб ёзилади.

Жадвалнинг яна қуйидаги атрибутлари мавжуд: **frame** ва **rules**. Биринчи атрибут **border>0** бўлган ҳолатда ташқи чегарасининг қандай бўлиши кераклиги, иккинчи атрибут эса ячейкаларни ажратувчи чегаралар қандай бўлиши кераклигини билдиради.

frame атрибути қабул қилиши мумкин бўлган қийматлар қуйидаги жадвалда келтирилган:

Атрибут номи	Вазифаси
Void	Чегара йўқ
Above	Юқори чегара
Below	Пастки чегара
Nsides	Чап ва ўнг чегаралар йўқ

Vsides	Юқори ва пастки чегаралар йўқ
lhs	Фақат чап чегара мавжуд
Rhs	Фақат унг чегара мавжуд

Rules атрибути қабул қилиши мумкин бўлган қийматлар қуйидаги жадвалда келтирилган:

Атрибут номи	Вазифаси
None	Чизик йўқ
Groups	<thead>, <tfoot>, <tbody> элементлари иштирок этганда фақат қаторлар оралиғида чизик бўлади, <colgroup>, <col> элементлари иштирок этганда фақат устунлар оралиғида чизиклар бўлади.
Rows	Чизиклар фақат қатор оралиқларида
Cols	Чизиклар фақат устун оралиқларида
All	Ҳамма чизиклар бўлади

2.6. Стиллар жадвали, шрифтлар, махсус белгилар. Тасвирлар ва мурожаатлар картасининг қўшимча хусусиятлари

HTML нинг илк версияларида web-саҳифаларни яратишда бироз қийинчиликлар бор эди. Улар асосий шрифт сифатида Times, ундан ташқари сарлавҳалар, рўйхатлар, тасвир ва гипермурожаатлар қўйиш элементлари каби ҳамма асосий элементлар ҳам қўлланилар эди. Маълумки Интернетдан тижорат мақсадида фойдаланила бошлангандан кейин Web нинг оммавийлиги ошиб кетди. Дизайнерлар ўзларининг ишланмалари билан Интернетнинг ривожланишига ҳиссаларини қўшдилар. Браузерлар ишлаб чиқарувчи йирик компаниялар улар эҳтиёжларини қондирадиган маҳсулот етказиб беришга ҳаракат қилдилар. Шундай қилиб, web-саҳифаларни яратишда янги, имконияти юқорирок элементлар ҳосил бўлди. Бу элементлар каторига <center>, , <blink> каби элементларни киритиш мумкин.

```
<font face= "Arial, Helvetica, Times New Roman" size= "+2"> Mening
saxifamga xush kelibsiz </font>
```

каби ёзувлар пайдо бўлди.

Кўпгина браузерлар элементини тушунмасди ва унинг натижасини экранга чиқармасди. Агар <h1> элементи ишлатилса график ва график бўлмаган ҳамма браузерлар унинг сарлавҳа эканлигини тушунардилар. (График бўлмаган браузерлар тасвирларни, ҳар-хил ўлчамдаги матнларни экранда ифодалай олмайдилар. Чўнтак компьютерлари, уяли телефон аппаратлари, ... Улар элементини «танимайдилар» лекин, <h1> ва <h2> элементларини ажрата оладилар.

Web-саҳифалар қисмлари билан ишлаганда уларни информатив бўлакка айлантириш мақсадида стиллар жадвали қўлланилади.

2.3 - мисол:

```
<p><font face= "Times New Roman" size= "+2" color= "red"> Web-
sahifalar qismlari bilan ishlaganda ularni informativ bulakka aylantirish maqsadida
stillar jadvali qo'llaniladi.</font></p>
```

ва ҳар бир абзацда ... такрорланиши керак. Стиллар жадвали ёрдамида бу ишни осонгина ҳал қилса бўлади:

```
<style>
p { font-family: Times New Roman; font-size: 16 pt; color:red }
</style>
```

Энди қаерда <p> элементи ишлатилса ҳам улар бир кўринишга эга ва уларга ортиқча элементини ишлатиш шарт эмас.

Агар қачондир абзацларнинг рангини ёки ўлчамини ўзгартириш керак бўлса, ҳар бир абзацни битталаб эмас, бошидаги <style> ўзгартирилади холос. Демак бу инструмент web-саҳифалар билан ишлаш вақтини камайтиради ва уларга бир хил кўриниш ато этади.

HTML тилидек стиллар жадвали ҳам бир нечта спецификацияларга эга. Биз CSS2 (Cascading Style Sheets 2) стандартидан фойдаланамиз. У жуда кўплаб web-публикацияларда қўлланилиб, ҳозирги кунда ҳамма замонавий график браузерлар у билан ишлай оладилар.

`` элементи **face**, **size**, **color** каби атрибутларга эга. У атрибутлар уз ичига олган матнга тегишли булади.

```
<font face= "Arial, Helvetica" size= "+2" color="red"> Xush kelibsiz
...</font>
```

face атрибутида шрифтлар рўйхати келтирилади. Агар фойдаланувчида биринчи шрифт бўлмаса, иккинчисидан у бўлмаса, учинчисидан фойдаланилади.

size атрибути 1 дан 7 гача қиймат қабул қилади. Бундан ташқари, охирга ўлчамдан катта ёки кичик ўлчамда ифодалаш имконияти ҳам мавжуддир. Бунинг учун `size="-2"` каби ёзилиб, унинг натижаси энг охирги ўлчамдан икки ўлчам кичик ўлчамда ёзилади.

color атрибути шрифт рангини билдиради. У ранг номи (red, green, yellow, blue, ...) ёки 16 лик ўлчамда ҳам ифодаланиши мумкин.

`<center>` контейнеридан фойдаланиб саҳифанинг ўртасига матн, тасвир, гипермуружаат, ... лар каби ихтиёрий элементни жойлаштириш мумкин, ҳатто мултимедиаларни ҳам.

Фойдаланувчи стиллар жадвалини қуйидаги турларини яратиши мумкин:

- Стиллар атрибути. Стиллар HTML элементлар ва контейнерларининг ҳар бирига қўлланилади.

- Элемент стилини аниқлаш. Стиллар жадвали орқали бирор бир элемент (`<p>`, `<h1>`, ...) ни стилини аниқлаш мумкин.

- Элементлар синфини аниқлаш. Стиллар жадвали орқали бир нечта элемент (`<p>`, `<h1>`, ...) лар стилини аниқлаш мумкин.

Стилларни икки усулда яратиш мумкин.

1. Стиллар ҳақидаги маълумотларни `<head>`, ... `</head>` секцияси ичига жойлаштириш мумкин.

2. Саҳифани алоҳида стиллар аниқланадиган файл билан боғлаб қўйиш мумкин.

Style атрибути.

```
<a href= "index.htm" style=text-decoration: none"> Бош саҳифа </a>
ёки
<table>
<tr> <td style= "background: red">100</td>
  <td style= "background: yellow">200</td> </tr>
</table>
ёки
<p style= "align:right; font style:italic; font-weight: bold"; background:
yellow>...</p>
```

каби ишлатиш мумкин.

Style элементи.

Агар яратилаётган web-саҳифа 2-3 саҳифадан иборат бўлса, style атрибути кўплаб қўлланилишига тўғри келади. Бутун саҳифага бир хил стил қўлламоқчи бўлсак, **style** элементини қўллаш қулайроқдир:

```
<head>
<style type= "text/css">
p {font-style: italic}
</style>
</head>
```

style элементини қўллаш умумий ҳолда қуйидагичадир:

```
<head>
<style type= "text/css">
element_nomi {xususiyati: qiymati}
</style>
</head>
```

Натижада ҳамма элементлар web-саҳифада бир хил стилга ега бўлади ва бир хил ифодаланади.

Бундан ташқари бир нечта элемент хоссаларини ҳам бирданига бериш мумкин:

```
<head>
<style type= "text/css">
p {font-style: italic}
h1 {color: blue}
ul {list-type: disc}
</style>
</head>
```

Бир элементнинг бир неча хоссаларини ҳам бирданига бериш мумкин:

```
<head>
<style type= "text/css">
p {font-style: italic;
background: yellow;
padding-left: 12 px}
</style>
</head>
```

тушуниш осонроқ бўлиши учун ҳамма хусусиятларни алоҳида қаторларга ёзиш тавсия этилади.

Агар бир неча элементга бир хил стил бериш керак бўлса қуйидагича ёзиш мумкин:

```
<head>
<style type= "text/css">
p, h1, h2, ul, ol {font-family: Arial, Helvetica}
</style>
</head>
```

Бундан ташқари бутун саҳифада ҳам ҳар хил стиллардан фойдаланиш мумкин:

```
<head>
<style type= "text/css">
```

```
body {font-family: Arial, Helvetica; font size: 12 pt}
ul { font-family:Times New Roman, Times; font size: 10 pt}
</style>
</head>
```

Махсус синфларни яратиш. Стиллар жадвали ёрдамида алоҳида элементларнинг хусусиятларини эмас, балки, махсус синфларни яратиш ҳам мумкин. Қуйидаги мисол билан танишинг:

```
<head>
<style>
h1.qizil {color: red}
</style>
</head>
<body>
<h1> Oddiy sarlavha </h1>
<h1 class= “qizil”> Oddiy bo’lmagan, qizil sarlavha</h1>
</body>
```

Бу мисолда **h1.qizil** селектори одатий **h1** элементи ўрнига ишлатилди. Бу ерда кўрсатилган стиллар фақатгина **class** атрибути қўлланилган жойда ишлайди. Бунинг афзаллиги шундаки, бир элементнинг ўзини бир неча хил стилларда ифодалаш имконияти пайдо бўлади.

```
<style>
p.body {font-family: Arial, Helvetica; font-size: 15 pt}
p.ajratilgan { font-family: Times New Roman, Times; font-size: 12 pt }
</style>
```

Энди web-саҳифанинг танасида абзац элементининг қайси стилини ишлатиш керак бўлса **class** атрибути ёрдамида керакли жойда кўрсатиб кетилади.

Бундан ташқари биз универсал синфларни ҳам яратишимиз мумкин. Web-саҳифанинг танасида ихтиёрий элементни унга улаб ўша стилда ифодалашимиз мумкин бўлади. Масалан:

```
<style>
.kichik {font-family: Arial, Helvetica; font-size: 15 pt}
</style>
```

Энди ҳоҳлаган элементни ўша синф ёрдамида стилини бериш мумкин бўлади:

```
<u1 class= “kichik”> ёки <p class=”kichik”>...
```

** элементи.** Стиллар жадвали билан ишлаш жараёнида яна битта - **** элементидан фойдаланиш ҳам мумкин. Уни қўллаб, ихтиёрий элемент стилини ўзгантириш мумкин. Бошқача айтганда **** элементи стиллари, токи ёпувчи теггача бўлган муълумотларга тегишли бўлади. Масалан:

```
<span style= “ font-variant: small-caps”>
Mana stili uzgartirilgan matn
</span>
```

**** элементи фақатгина матнларни ўз ичига олганлиги сабабли унга расмлар, мультимедиа элементлари ... ни қўллаш мумкин эмас.

Юқорида айтилагиндек, стиллар жадвалини ихтиёрий элементга қўллаш мумкин, худди шунингдек **** га ҳам:

```
<style>
span {font-family: Arial, Helvetica; font-size: 15 pt}
</style>
```

Энди web саҳифада ишлатилган ҳамма **** лар юқоридаги стилда ифоланадилар:

```
<p><span>Mana shunaka gaplar </span></p>
```

**** элементи учун ҳам бир нечта синфларни яратиш мумкин:

```
<style>
span.bir {font-family: Times New Roman, Times; font-size: 28 pt}
span.ikki { font-family: Arial, Helvetica; font-variant: small-caps }
</style>
```

**** элементи стиллар class атрибути ёрдамида берилади:

```
<p><span class="bir">Birinch sinf</span>
<span class="ikki"> Ikkinchi sinf</span></p>
```

**** элементини ҳам universal синфларга улаш мумкин:

```
<style>
.universal {font-family: Arial, Helvetica; font-size: 15 pt}
</style>
```

Ишлатишда:

```
<span class="universal">
```

деб ёзиб ишлатиб кетавериш мумкин.

<div> elementi. Энди стилларга тегишли бўлган яна битта элементни қараймиз. У **<div>** элементи (division) бўлиб, web-саҳифаларни секцияларга ажратиш учун қўлланилади. Бу элемент **** элементидан фарқли равишда ўз ичига ихтиёрий элементни олади ва унинг устида ихтиёрий таҳрирлаш ишларини бажариш мумкин.

<div> элементи **** элементи каби ишлатилади. Унинг учун жадвал стиллари аниқланган бўлиши мумкин, у **class** атрибутига эга булиши мумкин:

2.4 – мисол.

```
<html>
<head>
<style>
p {font-family: Arial, Helvetica; font-size: 25 pt}
</style>
</head>
<body>
<p>Stillar jadvali, shriftlar,maxsus belgilar. Tasvirlar va murojaatlar
kartasining kushimcha xususiyatlari </p>
<div style="background-color: gold; font-size=15 pt; font-family: Courier
```

```

New" >
  <ul type="A">
    <li> Stillar jadvali. </li>
    <li> Stillar jadvalini yaratish. style elementi.</li>
    <li> Maxsus sinflarni yaratish. span elementi. div elementi.</li>
    <li> Matn stillari. Shriftlar xususiyatlari. Fon va rang. </li>
    <li> Murojaatlar va ob'ektlar stillari. Birinchi xarf, birinchi kator. </li>
    <li> Jadval stillari. Maxsus belgilar.</li>
    <li> Veb-tasvir optimallashtirish. </li>
    <li> Animatsiyali tasvirlar.</li>
  </ul>
</body>
</html>

```

Мисолдан кўриниб турибдики, **<div>** элементи ўзида **<p>**, **** каби элементларни ҳам ифодалаш мумкин.

<div> элементи **align** атрибутига эга бўлиши ҳам мумкин. Агар
<div align="center">... </div>

ёзуви ёзилса, у ҳолда бу элемент ичида қўлланилган ҳамма элементлар (абзацлар, гипермурожаатлар, рўйхатлар, тасвирлар, ...) ҳаммаси web-саҳифанинг ўртасида ифодаланади.

Стилларни жадвалини алоҳида файлда ёзиб олиб кейин ундан ихтиёрий web-саҳифадан мурожаат орқали фойдаланиш ҳам мумкин. Бунинг учун `faul_nomi.css` файли ҳосил қилиниб унда ҳамма қўлланилиши керак бўлган стиллар ёзиб қўйилади (одатда блокнотда бу файлни яратиш қулай):

/ Birinchi darajali sarlavha */*

```

h1 {  font-family: Arial, Helvetica;
      font-size: 35 pt;
      font-weight: bold;
      word-spacing: 2 pt;      }

```

Энди бу нарсани web-саҳифага улаш керак бўлади. Бунинг учун **<head>** бўлимидаги **<link>** элементида фойдаланилади:

```

<head>
<link rel="stylesheet" href="d:/1.css" />
</head>

```

Матнлар хусусиятлари. Матн стиллари ёрдамида матнларни экранга қандай тасвирланиши ва жойлашишини белгилаш мумкин.

Қуйидаги жадвалда баъзи-бир матн стиллари келтирилган:

Хусусияти	Қиймат тури	Мисол
word-spacing	Сонли қиймат	1pt, 4em, 1in
letter-spacing	Сонли қиймат	3pt, 0.1em, +1
line-height	Сонли қиймат	14pt
Text-decoration	Аниқ қиймат	Underline, line-through, box, blink
Text-transform	Аниқ қиймат	Capitalize, lowercase, uppercase,

		none
Text-indent	Сонли қиймат ёки процент	1in, 5%, 3em
Vertical-align	Аниқ қиймат ёки процент	Baseline, sub, sup, top, middle, 50%
Text-align	Аниқ қиймат	Left, right, center, justify

word-spacing, letter-spacing, line-height ва **text** хусусиятлари бирор матннинг узунлигини белгилайди ва сонли қиймат қабул қилади. Сонлар мусбат ва манфий бўлиши мумкин. Ўлчов бирлиги сифатида пикселларни (px), дюймларни (in), миллиметрларни (mm), сантиметрларни (cm), пунктларни (pt), жорий матннинг баландлигини (em), ва жорий шрифт «х» харфининг баландлигини (ex) олиш мумкин.

Text-decoration фақат юқоридаги қийматларни қабул қилиб, шрифтни тузилишини ўзгартиради.

Vertical-align ва **Text-align** web-дизайнерга матнни ҳужжатда вертикал ёки горизонтал ҳолда жойлаштириш имкониятини беради.

Shriftlar xususiyatlari. элементинини ишлатмасдан стиллар жадвалини қўллаб ҳам шрифтнинг хусусиятларини (шрифтлар оиласи, ўлчами, ранги, ...) танлаш мумкин. Бунинг учун аввал <body> учун хусусиятни аниқлаб олиб, кейин маълум бир элементлар учун стиллар хусусиятлари берилади. Қуйидаги жадвалда баъзи бир хусусиятлар берилган:

Хусусияти	Қиймат тури	Мисол
Font-family	Шрифт номи	Helvetica, Arial, Times
Font-size	Сон/процент	12pt, +1, 120%
Font-weight	Сон/қиймат	Normal, bold, bolder, 100, 900
Font-style	Стил номи	Italic, oblique, normal
Font-variant	Стил номи	Normal, small-caps
Font	Комбинация	12pt Serif
Color	Ранг номи/16 лик қиймат	Red, green, #FF00AA

Font-family элементида шрифтлар оиласи берилади. Бу ерда умумий шрифтлар номлари келтирилиши тавсия етилади чунки, бизда мавжуд баъзи бир стандарт бўлмаган шрифтлар ҳар доим ҳам бошқаларда бўлмаслиги мумкин.

2.5 - мисол:

```
<style type = "text/css">
  p.standart { font-family: Helvetica, Arial, Sans Serif }
</style>
```

Font-size ни процентларда, пунктларда ёки аниқ қийматларда (larger, smaller) ифодалаш мумкин. **Font-weight** шрифтнинг қалинлигини билдириб 100 дан 900 гача қиймат қабул қилади (стандарт алинлик 400 ҳисобланади). **Font-style** қийшайиш даражасини билдиради ва **Italic, oblique, normal** қийматлардан бирини қабул.

Font-variant фақатгина оддий ёзув (**нормал**) ва кичик бош ҳарфлар (**small-caps**) дан бир-бирига ўтиш учун хизмат қилади.

`` хусусияти юқорида айtilган ҳамма хусусиятларни ўзида мужассам эта олади. Муаммо шундаки ҳар-хил браузерлар унинг натижасини ҳар-хил кўринишда ифолалашлари мумкин. Шунинг учун ҳамма матн хусусиятларини алоҳида-алоҳида ўрнатган маъқул.

Фон ва ранг. `<body>` тегида фон хусусиятини ўрнатиш мумкин, бу ҳолатда бутун web-саҳифага тегишли бўлиб қолади. Қуйидаги жадвалда фоннинг ҳамма хусусиятлари келтирилган:

Хусусияти	Қиймат тури	Мисол
background-color	Ранг номи/16 лик қиймат	White, blue, #0000FF
background-image	url()	url(image.gif), url(http://www.tdpu.uz/title.jpg)
background-repeat	Режим номи	Repeat, repeat-x, repeat-y, no-repeat
background-attachment	Режим номи	Scroll, fixed
background-position	Йўналиш ёки процент	Top, left, center, 20%, 65%
background	Комбинация	White ul(image.gif) repeat-x fixed

Рангларнинг қабул қилиши мумкин бўлган қийматлари: black, red, maroon, white, green, olive, lime, aqua, teal, blue, navy, yellow, brown, gray, silver, orange, purple, fuchsia ва ҳоказо.

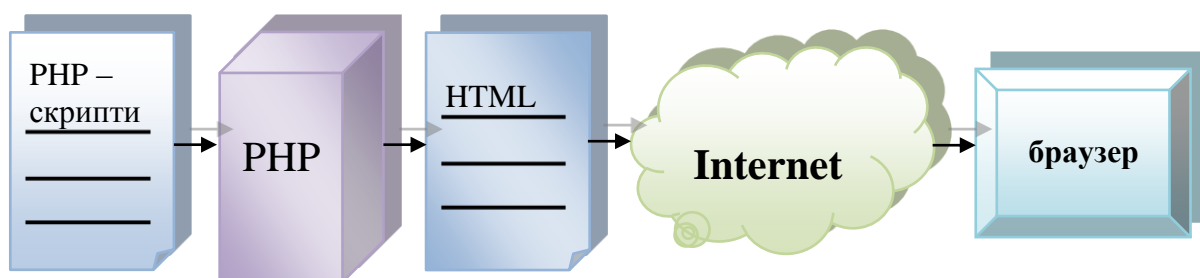
2.5 – мисол. 2.1 – расмда тасвирланган Талабалар портали учун стил ёзамиз.

```
<style>
<!--
h1 {color: white; font-size :24p; text-align : center ; font-family : arial , sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-family : arial , sans-serif; font-weight : bold}
td {background: silver}
p {color :darkblue; font-size: 12pt; text-align: justify;font-family: arial , sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-family : arial , sans-serif ; font-weight : bold}
a: link , a : visited, a : active {color : white }
--></style>
```

III БОБ. PHP ТЕХНОЛОГИЯСИ

3.1. PHP технологияси ҳақида қисқача

PHP — тили(технологияси) Интернетда ишлаш учун мўлжалланган бўлиб, Си тили каби универсал ва аниқ синтаксисга эга ва ўзида Perl ва Си тили имкониятларини мужассамлаштириб олган. Бу тил яқинда пайдо бўлган бўлсада (аниқроғи уни интерпретатори), у миллионга яқин серверларга ўрнатилган ва ўрнатилишда давом этмоқда. PHP тили HTML файлига динамик таркибни яратувчи PHP тили инструкцияларини қўйиш имкониятини беради. PHP дастурлари фақатгина PHP тилининг конструкцияларидан ёки PHP ва HTML тилининг конструкцияларидан ташкил топган бўлиши мумкин.



PHP технологиясининг асосий вазифаси - Webда ишловчиларга html-саҳифаларни осон ва тез ҳосил қилиш, уларни динамик ҳолда ишлаши ва ўзгартириш учун имкониятлар яратишдан иборат. PHP тили бу вазифаларни аъло даражада бажара олади. PHP га хос бўлган яна бир асосий жиҳат - унинг коди HTML га кадам-бакадам ўгирилади, тўғридан – тўғри ҳеч қандай чегаралар, тўсиқларсиз кўчади ва браузерга қандай бўлса ўша ҳолда чиқарилади. Бунда асосий шарт - PHP кодларини чегараловчи тегларни очиш ва ёпиш ёддан чиқмаслиги зарур. Шундай яратилган файл серверда сақланади. Фойдаланувчи саҳифага мурожаат қилганида, PHP у юборган буйруқларни қайта ишлаб браузерга у қандай юборган бўлса ўша ҳолатда қайтаради, худди CGI дастури C ва Perl да ёзилган дастурларни серверга узатганидек.

CGI (Common Gateway Interface - умумий шлюзли интерфейс) – HTTP сервер қўлланмаларини яратиш учун ташкил қилинган стандарт бўлиб, CGI сценарийлари ёки шлюз деб номланувчи бундай қўлланмалар www-серверини жорий вақтда ишга туширади. У Web серверига серверда ишлайдиган бошқа дастурлар (CGI сценарийлари) билан боғланишга имкон беради. CGI сценарийлари ёки шлюз одатда C/C++, Fortran, Perl, TCL, Unix Shell, Visual Basic, Apple Script ва бошқа дастурлар ёрдамида ёзилади. CGI сценарийлари қидирув, сервер томонидан карта тасвирларини бошқариш, ўйин кабилар ва бошқа турли функцияларни бажариш учун ишлатилади. Бироқ сценарийлар орасида энг ўзига хос фойдаланиш – бу формалар яратиш ва улар билан ишлашдир. CGI да аналогик скриптлар PHP га қараганда кўпроқ бўлади.

Саҳифада HTML кодидан ташқари PHP кодлари мавжуд бўлса, сервер буни тушуниши учун файл кенгайтмасини стандарт кўринишини ***.htm** ёки ***.html** ўрнига ***.phtml**, ***.php3** ёки ***.php** га ўзгартириш керак. Баъзида, PHP ни умуман кераги йўқ, ахир SSI бор-ку каби фикрларни ҳам эшитиб қолиш мумкин. Лекин, SSI бажара олган нарсани PHP ҳам бажара олиши мумкин ва бунга қўшимча у ундан ҳам кўп имкониятларга эга.

SSI дан фойдаланиш. SSI – Server Side Includes ёки ўзбекчада сервер томонидан қамраб олиш. SSI – бу тўғридан-тўғри HTML кодига қўйиладиган ва кўрсатмаларни Web серверига узатиш учун хизмат қиладиган директивалар (йўл-йўриқ дастурлар) дир.

Интерпретаторми ёки компилятор? Эхтимол, PHP 4 аввалги версияларидан фарқли равишда компилятор эканлигини эшитгандирсиз. Бу унчалик тўғри эмас. Терминлардаги тушунмовчиликларни ойдинлаштириш учун нима компилятор, нимани интерпретатор деб аташни келишиб оламиз. Кўпинча компилятор деб интерпретацияловчи транслятор дастурни, яъни асосий функцияси интерпретатор бўлган дастурни аташади. PHP ни 4 версиясида ҳам худди шундай.

Келинг, PHP нинг 4 - версияси қандай ишлашини кўриб чиқамиз. У дастурни кодини олиб, биринчи навбатда уни таҳлил қилади (хусусан, синтаксисини текширади) ва махсус *ички кўринишга трансляция* қилади. Бу кўриниш махсус байт – коддан иборат бўлиб, уни кўз билан кўриб тушуниб олиш имкони йўқ. Бироқ уни PHP амалга ошириши учун жуда қулай. Мана шу жараёни кўпинча кўпчилик нотўғри холда компиляция деб аташади. Кейин PHP ҳосил бўлган байт – кодни бажаради (интерпретациялайди). Бу жараён эса одатдаги интерпретатор ишини акс эттиради.

Юқорида таъкидлаганимиздек, PHP бир – биридан деярли мустақил бўлган 2 та блок — транслятор ва интерпретатордан ташкил топган. Нима учун бундай тузилган? Албатта, бу тезликни оширишни кўзлаб шундай қилинган. Ўзингиз таҳлил қилиб кўринг: синтактик таҳлил фақатгина трансляция пайтида бир марта амалга оширилади, тайёр бўлган «яримфабрикат» — байт-код кўп марта бажарилади.

Фараз қилайлик, дастурда кўп марта такрорланувчи цикл мавжуд бўлсин. PHP ни трансляция фазаси йўқ бўлган 3 версиясида такрорлашни навбатдаги ҳар бир қадамида цикл коди таҳлил қилиниши, сатрлари таҳлил қилиниши, синтаксиси текширилиши ва х.к. лар амалга оширилиши зарур. PHP 4 версиясида бу *бир марта* амалга оширилади ва циклни ҳар бир итерациясида тайёр байт – коддан фойдаланилади.

Хусусан, юқорида келтирилган PHP ни ишлаш схемаси амалдагига ҳамма вақт ҳам тўғри келмайди. Чунки, бу тилда шундай конструкцияларни яратиш мумкин, уларни трансляция пайтида (бунга масалан, дастур кодига номи фойдаланувчи томонидан кўрсатиладиган ташқи файлдаги кодни киритиш инструкцияси мисол бўлади) ички тасвирга ўтказишни сира имкони йўқ. Бундай пайтда PHP бу конструкциясини қолдириб кетади ва бу конструкцияси бажариш пайтида трансляция қилади. Албатта бу дастур бажарилишини секинлатади.

PHP интерпретаторини қуйидаги устунликлари мавжуд.

- Дастурни бажариш пайтида хатони аниқлаш соддалашади. Агар дастур бажарилишида узилиш бўлса интерпретатор бу ҳақида хабар қилади.
- Эгалланган хотирани бўшатиш ҳақида ўйламаса ҳам бўлади. Интерпретаторни ўзи дастурдаги у ёки бу ўзгарувчи керак ёки керакмаслигини аниқлайди ва ўзгарувчи учун ажратилган хотирани бўшатади.
- Ўзгарувчи типлари ҳақида ўйламаса ҳам бўлади.

Анъанавий равишда дастурлаш ҳақидаги барча китоблар «Hello world!» дастуридан бошланади. Биз ҳам ушбу анъанани давом эттирган ҳолда мисол келтирамыз:

```
<?
echo "Hello world!";
?>
```

Сценарийни браузерда ишга туширинг. Уни ишлаётганлигига гувоҳ бўласиз. Кўриниб турганидек, PHP – дастурининг структураси СИ ва Бейсик аралашмасини эслатади ва HTML таркибига ҳам киритилади. Бироқ биз юқорида PHP тилидаги оддий дастур мисолини кўриб чикдик. Бу тил ҳақидаги умумий тассаввурни бермайди. Бу бўлимда PHP тили конструкцияларини ўрганамиз, яъни тил асосларини ўрганамиз.

3.2. PHP дескрипторларини ишлатилиши

PHP кодлари “<?” белгиси билан бошланиб, “?” белгиси билан тугатилади, худди HTML дескрипторлари «кичик» (<) белгиси билан бошланиб ва «катта» (>) белгиси билан тугаганидек. Бу белгилар PHP дескрипторлари деб аталади ва улар WEB серверга PHP кодлари қаерда бошланиб, қаерда тугаганлиги ҳақида хабар беради. PHP кодлари блоки WEB серверда қайта ишланиб браузерга узатилади.

Бу дескрипторларнинг турли хил кўринишлари мавжуд бўлиб, кўрсатиб ўтилгани қисқартирилган формасидир.

PHP дескрипторларни тўрт хил тури мавжуд:

- Қисқартирилган тури:

<? ?>

Бу тур энг содда тур бўлиб, SGML инструкциясига мос келади.

- XML тури:

<?php ?>

Дескрипторнинг бу туридан XML хужжатларда фойдаланиш мумкин.

- SCRIPT тури:

<SCRIPT LANGUAGE='php'> </SCRIPT>

Дескрипторнинг бу тури JavaScript ва VBScript дастурлари билан ишлаганлар учун жуда яхши таниш.

- ASP тури:

<% %>

3.3. PHP операторлари

PHP интерпретатори бажариши керак бўлган бирор амал PHP оператори ёрдамида кўрсатилади. Қуйидаги мисолда PHP нинг маълумотларни чиқазиш операторларини кўриб чиқамиз. PHP да маълумотларни экранга чиқазиш оператори **echo** ва **print** операторларидир.

echo конструкцияси жудаям оддий ишлайди. У браузер ойнасига кўштирноқ ичидаги ёзувни чиқазади. Амал бажарилгандан сўнг албатта нукта-вергул кўйилиши керак, агар нукта-вергул унутиб қолдирилса, у ҳолда дастурда хатолик юзага келади. Масалан,

```
<?
echo "WEB sayt yaratamiz!";
?>
```


3.4. Изоҳлар

Изоҳлар фойдаланувчиларга дастурни кодларини ўқишда, дастурда ишлатилаётган код нимани аниқлашнинг изоҳлаш, унинг қисмларини номлашга, яъни бу код блоки дастурнинг қайси қисми эканлигини ажратишга ёрдам беради. PHP саҳифасида изоҳларнинг бир неча хилини ишлатиш мумкин: C, C++, Unix даги каби кўринишларини. Изоҳларнинг турлари:

// бир сатрли изоҳлар

бир сатрли изоҳлар

/* кўп сатрли изоҳлар */

// ва # белгиси билан бошланувчи изоҳлар сатрнинг охиригача тугатилиши керак, яъни кейинги сатрга ўтмаслиги керак.

3.1 – мисол. PHP да изоҳларнинг ишлатилиши.

```
<? php
/* bu ko`p satrli izoh.
   Bu yerda bir nechta satrlarni yozish mumkin*/
echo "Hodimlar ro'yhati"; // bu C++ stilidagi
// bir satrli izoh
echo "Hodimlar ro'yhati"; # bu Unix stilidagi
# bir satrli izoh
?>
```

IV БОБ. ЎЗГАРУВЧИЛАР ВА МАЪЛУМОТЛАР ТИПИ

Бу бобда ўзгарувчилар қандай яратилади ва уларга қандай ном бериш мумкин ва мумкин эмас; ўзгарувчиларга қийматларни тенглаштириш оператори; ўзгарувчи типлари кабилар билан танишишингиз мумкин.

4.1. Ўзгарувчи типлари

Ўзгарувчилар ҳақида сиз мактаб курсларидан тушунчага эгасиз. Ўзгарувчи бу ўзида маълумотни, яъни бирор бир сўзни, бирор бир сонни сақловчи белги.

Ўзгарувчилар дастур бажарилаётган вақтда, устида амаллар бажариш мумкин бўлган маълумотларни ўзида жамлаган хотира майдони номини ифодалайди. PHP да ўзгарувчилар \$(доллар) белгиси ва ўзгарувчи номини ёзиш орқали яратилади. Ўзгарувчи номи бирор бир ҳарф ёки сўз билан ифодаланади, лекин сон билан ифодаланиши мумкин эмас. **Масалан:** \$name

Ўзгарувчига бирор қиймат бериш учун уни ўша қийматга тенглаштириш лозим. Тенглаштириш “=” амали ёрдамида амалга оширилади. Бир нарсани ёддан чиқарманг, PHP да “=” амали айнан тенг маъносини эмас, балки тенглаштириш маъносини билдиради. Агар бу икки тушунчани аралаштирилиб юборилса, дастур кодида хатоликлар юзага келади.

Энди ўзгарувчини бирор бир қийматга тенглаштирамиз. **Масалан,** белгили қийматга, у ҳолда белгили қиймат “” ичида ёзилиши керак. Ва ҳар бир сатр “;” белгиси билан якунланиши керак.

```
<?
$name="Lola";
?>
```

Агар ўзгарувчига сон қиймат берадиган бўлсак, у ҳолда “” белгиси қўйилмаса ҳам бўлади.

```
<?
$name="Lola";
$age=22;
?>
```

4.1 - мисол. Ўзгарувчига қиймат бериш

```
<?php
$first = ' Matn '; // $first ga
// ' Matn ' qiymatini tenglaymiz
$second = $first; // $second ga
// $first o'zgaruvchisi qiymatini tenglaymiz
$first = ' Yangi matn '; // $first ning qiymatini
// ' Yangi matn ' ga almashtiramiz
echo " first nomli o'zgaruvchi " .
"$first qiymatiga teng <br>";
// $first ning qiymatini chiqazamiz
echo " second nomli ozgaruvchi" .
" $second qiymatiga teng";
// $second ning qiymatini chiqazamiz
?>
```

Натижа:

first nomli o'zgaruvchi Yangi matn qiymatiga teng
second nomli o'zgaruvchi Matn qiymatiga teng

Ўзгарувчи типи ўзида сақланувчи маълумотларни типига боғлиқ. PHP қуйидаги маълумот типларини қўллайди:

Скаляр типлар:

- `boolean` (мантикий)
- `integer, int`(бутун)
- `float, real, double` (ҳақиқий)
- `string` (сатрли)

Аралаш типлар:

- `array`(массив)
- `object`(объект)

Махсус типлар:

- `resource`(ресурс)
- `NULL`

Скаляр типдаги ўзгарувчилар

Скаляр типдаги ўзгарувчиларни маълумотларни сақловчи бир қути десак ҳам бўлади. У ўзига сонли, белгили ва матнли қийматларни қабул қилади. Скаляр типдаги ўзгарувчилар ҳақида янада аниқроқ тасаввур ҳосил қилиш учун қуйидаги схемани кўриб чиқамиз.

\$a=

12

\$b=

Bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim

Boolean (булев ёки мантикий) типи

Бу оддий тип қийматни ростлигини аниқлайди, бу типдаги ўзгарувчи фақатгина иккита қиймат қабул қилади- рост (TRUE) ёки ёлғон (FALSE).

Мантикий типни аниқлаш учун, иккита калит сўз TRUE ёки FALSE ишлатилади.

4.2 – мисол. Мантикий тип

```
<?php
$test = True;
?>
```

integer ёки int (бутун) типи

Integer (бутун) типи бутун сонлар тўпламида ($Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$) ги қийматлар учун ишлатилади. Бутун типлар ўнлик, ўн олтилик ва саккизлик санок системаларидаги қийматларни «-» ёки «+» белгиларини ишлатган ҳолда ҳам қабул қилади.

Агар саккизлик санок системаси ишлатилган бўлса, рақамдан аввал 0(ноль) сонини, ўн олтилик санок системасида эса рақамдан аввал 0x қўшиб ёзиш керак.

```
<?php
# o`nli sanoq sistemasidagi son
$a = 1234;
# manfiy son
$a = -123;
# sakkizlik sanoq sistemasidagi son(o`nlik sanoq sistemasidagi
# sistemasidagi 83 soniga ekvivalent)
$a = 0123;
# o`n oltilik sanoq sistemasidagi son (o`nlik sanoq
# sistemasidagi 26 soniga ekvivalent)
$a = 0x1A;
?>
```

Бутун типнинг энг катта қиймати 2 миллиардга яқин (бу 32 бит белги). Белгисиз бутун қийматларни PHP қабул қилмайди.

Float, real, double (нуқта билан ажратилган сон қиймати) типи

Нуқта билан ажратилган сон қиймати (ҳақиқий сон қиймати) типи куйидагича синтаксисларга эга.

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Double(иккилик қиймат) float, real типлари бир-бирига ўхшаш бўлиб– ҳақиқий сон қийматлари учун ишлатилади.

String (сатрли) типи

Сатр – бу белгилар тўплами. PHP да бошқа дастурлардаги каби байт бу турли 256 та белгини қабул қилади. String типи сатрли қийматлар учун ишлатилади.

PHP да сатр турли хил усуллар билан берилади:

- битталиқ кўштирноқ ёрдамида
- иккиталиқ кўштирноқ ёрдамида
- heredoc – синтаксиси ёрдамида

Битталиқ кўштирноқ

Сатрларни аниқлашни оддий усули – бу уни битталиқ кўштирноқ «'» ичига олиб ёзиш керак. Сатр ичида битталиқ кўштирноқ ишлатиш учун бошқа дастурлаш тилларидаги каби, унинг олдида тескари чизик белгиси кўйилиши керак. Агар тескари чизик белгисини сатрнинг охирида битталиқ кўштирноқ олдида куйишга тўғри келса, у ҳолда уни яна биттага ошириб кўйиш керак «\\'».

Агар битталиқ кўштирноқ ичига олинган сатр ичида тескари чизик «\» белгиси ихтиёрий белги олдида («\» ва «'» дан фарқли) учраса, у ҳолда бу оддий белги сифатида қаралади ва экранга шу кўринишда чиқади. Шунинг учун тескари чизик сатр охирида келган бўлса, унга яна битта тескари чизик кўшиб ёзиш керак. PHP нинг тескари чизик билан бошланувчи белгилар тўплами мавжуд бўлиб, улар бошқарувчи кетма-кетликлар деб аталади ва махсус қийматларга эга(/n, :// ва х). Шунга кўра, иккита турли синтаксислар билан фарқ қилувчи ўзгарувчилар ва махсус белгилар учун бошқарувчи кетма-кетликлар сатрларда битталиқ кўштирноқ ичида учраса, улар қайта ишланмайди.

4.3 - мисол. Бошқарувчи кетма-кетликларни ишлатилиши

```

<?php
echo ' Satrda qo`shtirnoq shu ko`rinishda ishlatiladi';
echo Agar satrda
Warning: Unexpected character in input:  '\' (ASCII=92) state=1 in
D:\xampplite\htdocs\learn_php\code.php on line 2
' u holda ' .
' uni oldidan quyidagi belgi qo`yiladi \\' ;
// Chiqadi: Agar satrda ' ishlatilsa, u holda
// uni oldidan quyidagi belgi qo`yiladi \
echo 'Ushbu diskni o`chirasizmi C:\\*. *?';
// Chiqadi: Ushbu diskni o`chirasizmi C:\\*. *?
echo 'Yangi satrni: \n qo`ymaydi';
// Chiqadi: Yangi satrni : \n
// qo`ymaydi
echo ' O`zgaruvchi $expand $either' .
'kabi qo`yilmaydi';
// Chiqadi: O`zgaruvchi $expand
// $either kabi qo`yilmaydi
?>

```

Иккиталик қўштирноқ

PHP да иккиталик қўштирноқ «"» билан чегараланган сатрларда махсус белгилар учун бир қанча бошқарувчи кетма-кетликлар ишлатилади. Улардан баъзилари 4.1 - жадвалда келтирилган

4.1 - жадвал. Сатрлар учун ишлатиладиган бошқарувчи кетма-кетликлар

Бошқарувчи кетма-кетликлар	
Кетма - кетлик	Қиймати
\n	Янги сатр (ASCII да LF ёки 0x0A (10))
\r	Курсорни қайтиши
\t	Горизонтал табуляция (ASCII да HT ёки 0x09 (9))
\\	Тескари қийшиқ чизик
\\$	Доллар белгиси
\"	Иккиталик қўштирноқ

Heredoc синтаксиси

Сатрлар катта ҳажмдаги маълумотлар (жадваллар, сатрлар, мурожаатлар) ҳам бўлиши мумкин. Бундай сатрларни echo операторидан фойдаланиб экранга чиқазиш бир қанча муаммолар келтириши мумкин. Ушбу ҳолатларда сатрий ўзгарувчиларни **heredoc**-синтаксиси усулидан фойдаланиб аниқлаш мумкин. Бу усулда сатрлар <<< маркери билан бошланиб, сўнг идентификатор сўзи ёзилади ва сатр яна шу идентификатор сўзи билан тугатилади.

Синтаксиси:

```
echo <<<MATN
```

```
... ..
```

```
MATN;
```

Heredoc матни қўштирноқ ичида жойлашган матн каби бўлади ва heredoc синтаксиси ишлатилганда қўштирноқ ишлатилишига ҳожат йўқ.

Heredoc матни ичида ўзгарувчиларни ҳам бериш мумкин ва бу ҳолда heredoc ичидаги ўзгарувчилар ҳам қайта ишланади.

4.4 - мисол. heredoc синтаксисининг ишлатилиши

```
<?php
$str = <<<EEE
<table cellspacing="1" width="100%" cellpadding="1" border="0" >
<tr><td>
<div class="body_txt">
<a href="#"><li>$fak fakultetining tarixi</a>
</div>
</td></tr>
</table>
EEE;
// EEE - bu yerda identifikator.
?>
```

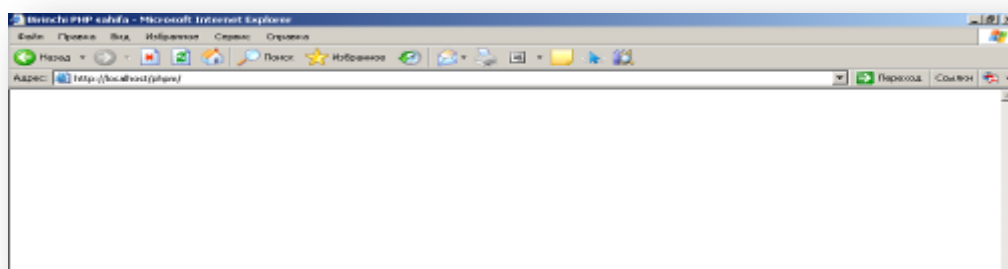
Скаляр типдаги ўзгарувчиларни boolean, integer, float, string турларини кўриб ўтдик.

4.5 - мисол. Ўзгарувчиларнинг қийматларини чиқазиш

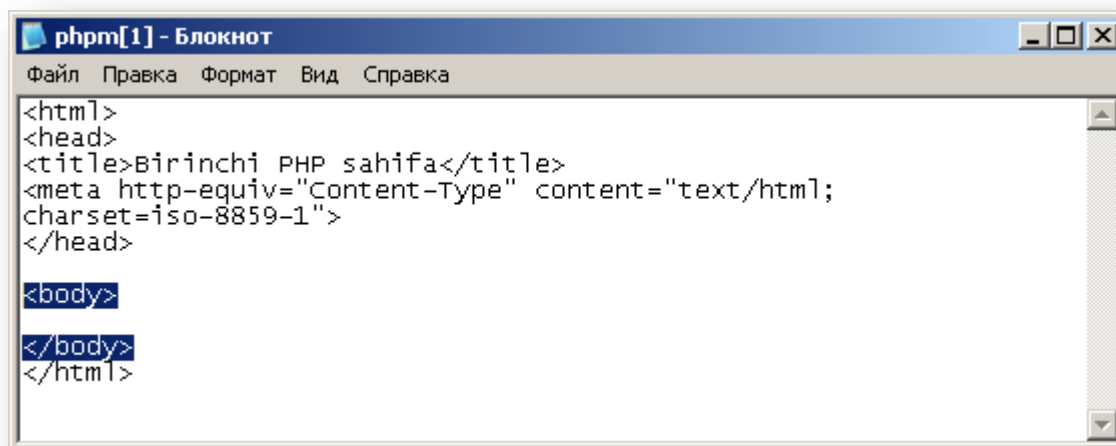
```
<?
$name="Lola";
$age=22;
$price=2.11;
?>
```

Бу ерда, \$name ўзгарувчиси string типдаги, \$age ўзгарувчиси integer типдаги, \$price ўзгарувчиси эса double типдаги қийматларни қабул қилади.

Энди саҳифани сақлаб, браузерда кўрамиз.



Саҳифада ҳеч қандай ёзув ёки маълумот йўқ. Ички (html) кодини кўрадиган бўлсак.



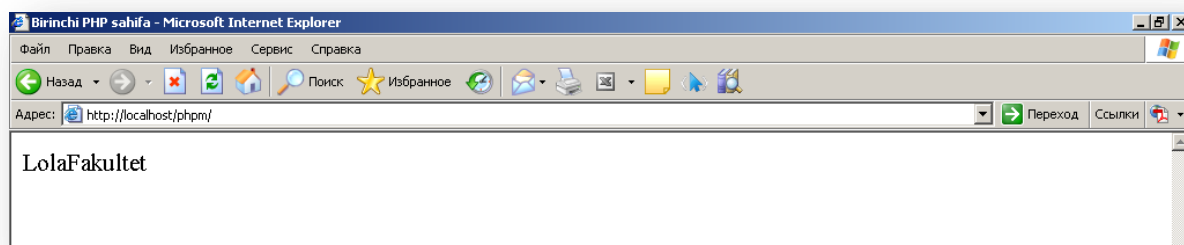
Биз ёзган PHP кодлар браузерда ҳам, ички кодда ҳам чиқмади. Нимага? Чунки биз фақатгина ўзгарувчиларга қиймат бердик холос, браузерга чиқаришга эса буйруқ бермадик.

echo операторидан фойдаланиб ўзгарувчиларнинг қийматларини браузерга чиқазамиз.

```
<?
$name="Lola";
$age=22;
echo "$name";
?>
```

Fakultet сўзини ҳам браузерга чиқазмоқчимиз, у ҳолда

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "Fakultet";
?>
```



Бу ерда қийматлар кетма-кет ҳолда чиқди. Уларни ажратиш учун бизга таниш бўлган
 тегидан фойдаланишимиз мумкин.

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "<br>Fakultet";
?>
```

Аралаш типдаги ўзгарувчилар

Array типи (Массивлар)

Скаляр **типдаги** ўзгарувчи — бу ягона *қиймат* сақладиган катакча хотирасини номи. Массив **типидаги** ўзгарувчи эса *қийматлар тўплами* сақладиган катакча хотирасини номи. Бошқача қилиб айтганда, **бир нечта типдаги** маълумотларнинг бир типдаги ўзгарувчида **сақланиши array (массив)** деб аталади. Массивда сақланаётган қийматлар массив элементлари деб аталади. Битта массив бир нечта элементлардан ташкил топган **бўлиб**, ҳар битта элемент ўзида ягона қиймат(матн, сон ёки бирор бошқа массив)ни сақлайди. Массивнинг ҳар бир элементи массив билан калит орқали боғланади. Бошқа типдаги ўзгарувчиларда битта қиймат сақланса, массив типдаги ўзгарувчиларда бир нечта қийматлар сақланади. Элемент идентификацияси усули билан фарқ қилувчи иккита типдаги массив мавжуд.

Биринчи типдаги массив элементи калитлари кетма-кет келувчи **сонли** индекс сифатида аниқланган сонли индексли массивлар. Массивнинг **иккинчи** тип **ассоциатив** шаклда бўлиб, элементга мурожаат этиш учун қиймат билан мантиқий боғланган калитлар ишлатилади. Ўзида бошқа массивларни сақлаган массив **кўп ўлчовли массив** деб аталади.

Массивлар `array()` конструкцияси ёрдамида эълон қилинади.

Синтаксиси:

```
array([калит] => қиймат,  
[калит1] => қиймат1, ... )
```

`array()` массиви жуфтлик *калит* => **қиймат** параметрларини вергул билан ажратиб қабул қилади. => белгиси қиймат ва уни калити орасидаги боғланишни ўрнатади. Калит тип бутун ва сатрли, қиймат эса PHP нинг ихтиёрий типда бўлиши мумкин.

Масалан, `$name=array("Lola", "Karim", "Vasila", "Akmal", "Botir");`

Сонли индексли массивлар



`$a` ўзгарувчисини нашр қилишга берсак, битта қийматни экранга чиқазади. `$name` ўзгарувчиси массив бўлганлиги сабабли, калитни кўрсатишимиз керак. Яъни `Akmal` ни экранга чиқазмоқчи бўлсак, `$name[3]` деб калитини кўрсатишимиз керак. Сонли индексланган массивлар нолдан бошланади.

```
$name=array(0=>"Lola", 1=>"Karim", 2=>"Vasila", 3=>"Akmal",  
4=>"Botir");
```

ёки

```
$name=array("Lola", "Karim", "Vasila", "Akmal", "Botir");
```

ёки

```
$name[0]= "Lola";  
$name[1]="Karim";  
$name[2]= "Vasila";  
$name[3]= "Akmal";  
$name[4]= "Botir";
```

Ёки

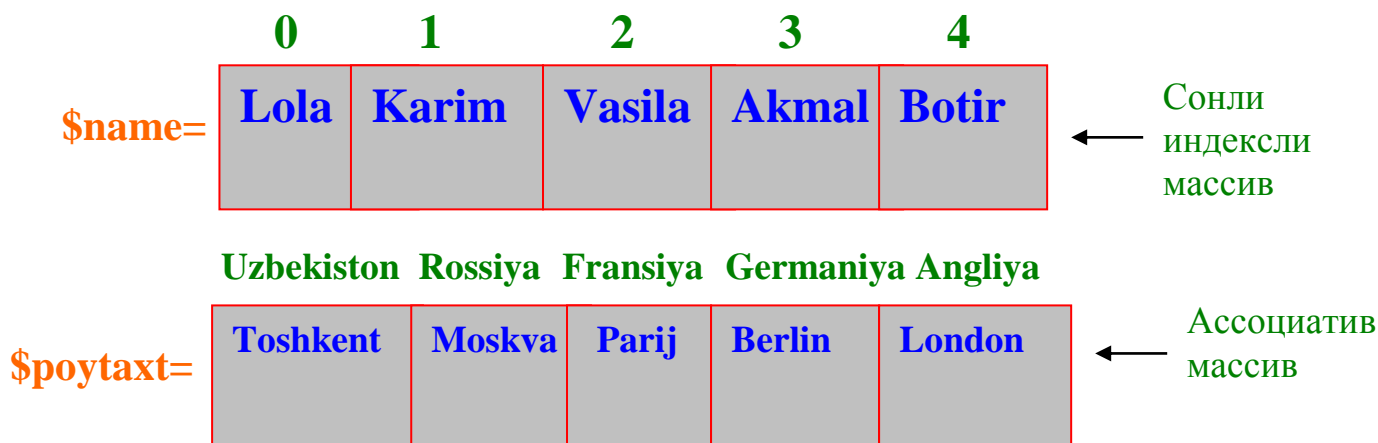
```
$name[ ]= "Lola";  
$name[ ]="Karim";  
$name[ ]= "Vasila";  
$name[ ]= "Akmal";
```

\$name[]= "Botir";

Ассоциатив массивлар

Кўпгина дастурлаш тилларида массив индекслари 0 ва 1 дан бошланувчи сонлардан иборат. Бу дастурлаш тилларидан фарқли равишда PHP ассоциатив массивларни ҳам қўллайди. Ассоциатив массивларни сонли индексланган массивлардан фарқи ассоциатив массивларда калит ихтиёрий қийматлар(матн, сон ва ҳ.)ни қабул қилади. Бу массивда массив элементлари калит сифатида берилган сонлар билан эмас, балки сатрлар билан боғланади.

4.6 - мисол. PHP да ассоциатив массивларни ишлатилиши



```
<?
$poytaxt["Uzbekiston"]="Toshkent";
$poytaxt["Rossiya"]="Moskva";
$poytaxt["Fransiya"]="Parij";
$poytaxt["Germaniya"]="Berlin";
$poytaxt["Angliya"]="London";
echo $poytaxt["Fransiya"];
?>
```

Натижа: Parij

Шу давлатларни аҳолиси сонини ҳам массив сифатида беришимиз мумкин.

4.7 - мисол. PHP да ассоциатив массивларни ишлатилиши

```
<?
$poytaxt["Uzbekiston"]="Toshkent";
$poytaxt["Rossiya"]="Moskva";
$poytaxt["Fransiya"]="Parij";
$poytaxt["Germaniya"]="Berlin";
$poytaxt["Angliya"]="London";
$aholisi=array ("Uzbekiston"=>"25", "Rossiya"=>"150", "Fransiya"=>"40",
"Germaniya"=>"35", "Angliya"=>"35");
echo "Uzbekistonning poytaxti $poytaxt["Uzbekiston"], aholisi $aholisi["Uzbekiston"] mln kishidan iborat.";
?>
```

Натижа: Uzbekistonning poytaxti Toshkent, aholisi 25 mln kishidan iborat.

Кўп ўлчовли массив

Массив калит ва қийматларни ўзида мужассамлаштирган оддий рўйхатдан иборат бўлиши шарт эмас, балки массив ячейкаси ўзида бошқа массивларни ҳам сақлаши мумкин. Бундай массивлар кўп ўлчовли массивлар дейилади. Бу массивда аралаш (сонли ва ассоциатив) индекслашга рухсат берилади.

Кўп ўлчовли массив сифатида, икки ўлчовли массивларни ҳам яратиш мумкин. Икки ўлчовли массив кенглиги ва баландлиги сатрлар ва устунларга аксланувчи матрица ёки жадвал кўринишида бўлади.

4.8 -мисол. Кўп ўлчовли массивларни ишлатилиши

Ассоциатив массив

	Uzbekiston	Rossiya	Fransiya	Germaniya	Angliya
\$poytaxt=	Toshkent	Moskva	Parij	Berlin	London

Кўп ўлчовли массив

	Nexia	Damas	Matiz	Tiko				
\$auto=	yili	rangi	yili	rangi	yili	rangi	yili	rangi
	1999	oq	2003	qora	2005	oq	1999	qizil

\$auto массивини қарайдиган бўлсак, Nexia индексли массив қиймати бошқа бир массивга тенг. Демак бу кўп ўлчовли массивни PHP кодини ёзиб чиқамиз.

```
<?
$auto["nexia"] = array("yili" => "1999", "rangi" => "oq");
$auto["damas"] = array("yili" => "2003", "rangi" => "qora");
$auto["matiz"] = array("yili" => "2005", "rangi" => "oq");
$auto["tiko"] = array("yili" => "1999", "rangi" => "qizil");
?>
```

Кўп ўлчовли массив элементларини киритдик. Энди шу массив элементларини экранга чиқазиб кўрамиз.

```
<?
$auto["nexia"] = array("yili" => "1999", "rangi" => "oq");
$auto["damas"] = array("yili" => "2003", "rangi" => "qora");
$auto["matiz"] = array("yili" => "2005", "rangi" => "oq");
$auto["tiko"] = array("yili" => "1999", "rangi" => "qizil");
echo "Matiz avtomobilini ishlab chiqarilgan yili - ". $auto["matiz"]["yili"];
?>
```

Натижа.
Matiz avtomobilini ishlab chiqarilgan yili - 2005

Массив калитлари

Агар калит берилмаган бўлса, у ҳолда калит ўрнига калитнинг энг катта қийматига 1 сонини қўшиб ёзиб кетилади. Калит бирор- бир қийматга тенг бўлса, у ҳолда ўша қиймат ўзи ёзилади. PHP нинг 4.3.0 дан кейинги версияларида калитнинг энг катта қиймати манфий сон бўлса, массивнинг кейинги қиймати нол(0) деб олинади.

Агар калит ўрнида TRUE ёки FALSE қийматлари ишлатилса, у ҳолда калит типи integer бўлган 0 ёки 1 қийматни беради. Калит ўрнида NULL

ишлатилса, бўш сатрни беради. Калит ўрнида бўш сатр ҳам ишлатилиши мумкин, фақат уни қўштирноқ ичига олиб ёзиш керак ёки алоҳида ёзилганда бўш тўртбурчак қавс ишлатилади. Калит ўрнида объект ва массивлар ишлатилмайди.

4.9 - мисол. PHP да массивларни ишлатилиши

```
<?php
// $arr va $arr1 massivlari ekvivalent
$arr = array(-3 => 43, 32, 56, 86, 97, "b" => 12);
$arr1 = array(-3 => 43, 0 => 32, 1 => 56, 2=> 86,
3 => 97, "b" => 12, "c" => true, " " => 28);
Echo $arr1[null];
?>
```

Агар янги калит ва янги қиймат берилса (\$name["new_key"]="new_value"), у ҳолда массивга янги элемент қўшилган ҳисобланади. Агар биз калитни кўрсатмасак, фақатгина қийматга тенгласак, \$name[]="new_value", у ҳолда массивнинг янги элементи сонли калит бўлиб, мавжуд калитнинг энг катта қийматидан бир қийматга ошган бўлади.

```
<?
$name=array(2=>5, 3=>6,4=>15);
$name[] = value1;
/* qiymati value1 kaliti 5 (bizda kalitning
eng katta qiymati 4 bo'lganligi uchun)ga
teng massiv kiritdik */
echo $name;
?>
```

Массивнинг элементини ўзгартириш учун, калитини янги қийматга тенглаштириш етарли. Массив калитини ўзгартириш мумкин эмас, фақатгина элемент(калит/қиймат жуфтлигини)ни ўчириб юбориш ва янгисини киритиш мумкин. Массив элементини ўчириш учун, **unset()** функциясида фойдаланилади.

```
<?php
$name=array(2=>5, 3=>6,4=>15);
$name[] = value1;
/* qiymati value1 kaliti 5 (bizda kalitning
eng katta qiymati 4 bo'lganligi uchun)ga
teng massiv kiritdik */
echo $name;
$name[] = "Sobir";
// kaliti(indeksi)5 ga teng
// $name[5] = "Sobir" ga ekvivalent
// element yaratildi
Echo $name.'  
';
$name["starosta"] = "Komila";
/* Kaliti "starosta", qiymati "Komila"
bo'lgan yangi massiv elementini yaratadi */
unset($name[4]);
// massivdan kaliti 4 ga teng bo'lgan
//elementni o'chiradi
Echo $name.'  
';
unset ($name);
// massivni to'liq o'chiradi
Echo $name;
?>
```

Массив элементи киритилаётганда бўш тўртбурчак қавс ишлатилса, массив элементига калит қўйиш учун охирги қайта индексланган массив

элементлари орасидан калити энг каттасини қидиради. Массивни қайта индекслаш учун **array_values()** функциясидан фойдаланилади.

4.10 - мисол. Массивларни қайта индекслаш.

```
<?php
$arr =
array ("a","b","c");
/* qiymati "a", "b" va "c" ga teng.
Kalit ko`rsatilmaganligi uchun
mos ravishda 0,1,2 ga teng bo`ladi.*/
print_r($arr);
// massiv(kalit, qiymat)ni ekranga chiqazamiz
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
// massivni barcha qiymatlarini o`chiramiz.
print_r($arr); // massiv(kali, qiymat)ni ekranga chiqazamiz
$arr[] = "aa"; // massivga yangi element qo`shamiz,
// uni indeks (kaliti) nol emas, 3 bo`ladi
print_r($arr);
$arr = array_values($arr); // massivni qayta indekslaymiz
$arr[] = "bb"; // bu element kaliti 1 bo`ladi
print_r($arr);
?>
```

Натижа:

```
Array ( [0] => a [1] => b [2] => c ) Array ( ) Array ( [3] => aa ) Array ( [0]
=> aa [1] => bb )
```

Object tipi (объектлар)

Объектлар – объектга йўналтирилган дастурлашдан(ОЙД) олинган маълумотлар типи. ОЙД принципига асосланиб, класс - бу хусусияти ва ишлаш методи аниқланган объектлар тўплами, объект эса классдан олинган бир нусха.

Бирор бир вазифани бажарувчи амаллардан ташкил топган классни эълон қиламиз. Класс эълон қилиб бўлингандан сўнг, классни *аъзоси бўлган алоҳида аниқ элемент* – объектни яратамиз. Бу жараён классни нусхасини яратиш деб номланади. Объект new калит сўзи ёрдамида яратилади. Бунинг учун, объект қайси классни нусхаси эканлигини ва барча параметрларни кўрсатиш керак. Объектлар билан ишлашда классларни чақириш учун → операторидан фойдаланилади.

Мисол учун, дастурчилар класс - бу дастур туза оладиган, компьютерда ишлай оладиган ва бундан ташқари бошқа инсонлар каби исм ва фамилияга эга инсонлар класс. Энди, агар бирорта аниқ бир дастурчи, масалан, Одил Каримовни оладиган бўлсак, у ҳолда у бошқа дастурчилар каби дастур туза оладиган, исм - фамилияга ва бошқа хусусиятларга эга дастурчилар классининг объекти ҳисобланади.

4.11 – мисол. PHP да объектлар.

```
<?php
//insonlar klassini e`lon qilamiz
class person
{ function prog_learn($name)
{ echo "$name PHP da dastur tuza oladi"; }
function lang_learn($name1)
{echo "$name1 ingliz tilidi gaplasha oladi";}
}
```



```
// klassni obyektlarini yaratamiz
$a = new person();
$aa=$a->prog_learn("Lola");
$b=$a->lang_learn("Karim");
$c= $a-> prog_learn();
?>
```

Klass har bitta obyekt yaratilganda quyidagi natijani chiqaradi:

```
Lola PHP da dastur tuza oladi
Karim ingliz tilida gaplasha oladi
PHP da dastur tuza oladi
```

Махсус тундаги ўзгарувчилар

resource tinni (ресурслар)

Ресурс – бу ташқи ресурсга (масалан, маълумотлар базаси билан боғланишга) мурожаатни таъминловчи махсус ўзгарувчи. Ресурслар махсус функциялар(масалан, `mysql_connect()`, `pdf_new()` ва бошқалар) билан яратилади ва ишлатилади.

Null tinni

Махсус қиймат *NULL* ўзгарувчи қийматга эга эмаслигини англатади.

Ўзгарувчи NULL ҳисобланади, агар:

у *NULL* константасига тенглаштирилган бўлса (`$var = NULL`);

у ҳали бирор – бир қийматга тенглаштирилмаган бўлса;

unset() ёрдамида ўчирилган бўлса.

Фақатгина битта регистрга боғлиқ бўлмаган *NULL* калит сўзли *NULL* типдаги қиймат мавжуд.

4.2. Идентификаторлар

Умумий термин «идентификатор» фойдаланувчи томонидан аниқланган ўзгарувчилар, функциялар ва бошқа объектлар учун қўлланилади. PHP идентификаторлари бир нечта шартларни қониқтириши керак:

- идентификаторлар бир ёки бир нечта белгилардан ташкил топган бўлиб, ҳарфлар ёки остки чизик белгиси билан бошланади;
- идентификаторлар фақатгина ҳарфлар, сонлар, остки чизик белгилари ва 127 дан 255 гача бўлган ASCII коди белгиларидан иборат бўлиши мумкин.

Масалан:

Рухсат этилган идентификаторлар	Рухсат этилмаган идентификаторлар
my_f	Thi&th
Siz	!coun
_somew	4wa

Идентификаторларда белгилар регистрга сезгир. Мисол учун, \$rec номли ўзгарувчи \$Rec номли ўзгарувчидан фарқ қилади. Идентификатор узунлиги чегараланмаган. Идентификатор PHP нинг бирорта стандарт калит сўзига мос келиши мумкин эмас.

4.3.Ўзгарувчилар билан ишловчи функциялар

PHP нинг ўзгарувчилар билан ишловчи функциялари билан танишамиз. Бу функциялар кутубхонаси ўзгарувчиларни текшириш ва уларни турли усуллар билан ўзгартириш учун ишлатилади.

Ўзгарувчиларнинг типларини текшириш ва ўрнатиш

Синтаксиси:

string gettype(mixed var);

int settype(string var, string type);

gettype() функцияси унга узатилган ўзгарувчининг типини аниқлаб, тип номини ёки агар ўзгарувчи типни стандарт типлардан (integer, double, string, array ёки object) бирортаси ҳам бўлмаса, "unknown type" ни қайтаради.

settype() функциясига параметрлар сифатида ўзгарувчи ва стандарт типлардан бири берилади. Бу функция параметр сифатида берилган ўзгарувчи типини параметр сифатида берилган янги типга ўзгартиради.

Мисол:

```
<?
$a = 56;
echo gettype($a). "<br>";
settype ($a, "double");
echo gettype($a). "<br>";
?>
Natija:
integer
double
```

gettype() функциясига мурожаат этганда, **\$a** ўзгарувчисининг типини **integer** эканлигини чиқазади. **settype()** функциясига мурожаат этгандан сўнг, унинг типини **double** га ўзгаради.

PHP нинг типларни текшириш учун бир қанча функциялари мавжуд. Бу функцияларнинг ҳар бири параметр сифатида ўзгарувчини қабул қилади ва true ёки false қийматларини қайтаради.

- **is_array()**
- **is_double(), is_float(), is_real()** (бир хил функция)
- **is_long(), is_int(), is_integer()** (бир хил функция)
- **is_string()**
- **is_object()**

Ўзгарувчи ҳолатини текшириш

PHP нинг ўзгарувчи ҳолатини текширувчи бир қанча функциялари мавжуд. Улардан бири – **isset()**.

Синтаксиси:

int isset(mixed var);

бу функция параметр сифатида ўзгарувчи номини олади ва натижа сифатида агар ўзгарувчи мавжуд бўлса, true, акс ҳолда false ни қайтаради.

Ўзгарувчини **unset()** функцияси ёрдамида ўчириш мумкин. Унинг синтаксиси қуйидагича:

int unset(mixed var);

бу функция ўзгарувчини ўчиради ва натижа сифатида true қийматини қайтаради.

empty() функцияси қиймати бўш ва нол бўлмаган ўзгарувчининг мавжудлигини аниқлайди ва мос ҳолда true ёки false қийматларини қайтаради.

Синтаксиси:

int empty(mixed var);

Ушбу учта функцияга мисол кўрамыз. Вақтинчалик яратилган саҳифага ушбу кодларни киритамиз:

```
<?
echo isset($bor);
echo isset($yuq);
echo empty($bor);
echo empty($yuq);
?>
```

isset() функцияси **\$bor** ўзгарувчиси учун форма майдонида қиймат киритилган ёки киритилмаганлигига боғлиқ бўлмаган ҳолда, true қийматини қайтаради. **empty()** функцияси эса киритилган қийматга боғлиқ ҳолда натижани қайтаради.

\$yuq ўзгарувчиси мавжуд эмас, шунинг учун бу ўзгарувчи учун **isset()** функцияси false қийматини, **empty()** функцияси эса – true ни қайтаради.

Кўриб ўтилган функциялар формадаги мавжуд майдонлар фойдаланувчи томонидан тўлдирилганлигини текшириш учун ишлатилади.

Ўзгарувчиларни қайта интерпретациялаш

Синтаксиси:

int intval(mixed var);

double doubleval(mixed var);

string strval(mixed var);

ушбу ҳар бир функция параметр сифатида ўзгарувчини қабул қилади ва мос типга ўзгартирилган ўзгарувчи қийматини қайтаради.

V БОБ. АМАЛЛАР

Бу бобда сатрлар билан ишлаш(сатрларни сатрлар билан бирлаштириш, ўзгарувчилар билан сатрларни бирлаштириш), арифметик амаллар(ўзгарувчи қийматлари устида қўйиш, айириш, кўпайтириш ва бўлиш амалларини тўғри бажариш), мантиқий амаллар, тенглик амаллари, муносабат амаллари, хатоларни яшириш амаллари билан танишиш мумкин.

5.1. РНР амаллари

Амал белгилари – булар ўзгарувчилар ва қийматлар устида амаллар бажариш учун ишлатиш мумкин бўлган белгилар.

5.1 - жадвал. РНР амаллари

Амаллар	Мазмуни
()	Қавс
new	Объект нусхасини яратиш, янги объект ҳосил қилиш
!~	Мантиқий рад этиш
++,--	Битта сонга ошириш(инкремент), камайитириш(декремент)
@	Хатоларни яшириш
/,*,%	Бўлиш, кўпайтириш, қолдиқ
+, -, .	Қўйиш, айириш, конкатенация
<<, >>	Чапга суриш, ўнгга суриш(разрядли)
<, <=, >, >=	Кичик, кичик ёки тенг, катта, катта ёки тенг
==, !=, ===, <>	Тенг, тенг эмас, айнан тенг, тенг эмас
&, ^,	Разрядли амаллар AND, XOR ва OR
&&,	Мантиқий амаллар AND ва OR
?:	Тернар амал
=, +=, *=, /=, ., %=, ?=, /=, :=, <<=, >>=,	Тенглик амаллари
AND, XOR, OR	Мантиқий амаллар AND, XOR ва OR

5.2. Сатрлар билан ишлаш амаллари

Ўзгарувчилар ва литераллар. Ўзгарувчилар маълумотлар белгиси. \$name – ўзгарувчи ёки бирор бир маълумотни ўзида сақловчи белги. Сатрлар ҳам маълумотлар. “talaba”- бу литерал, у қандай бўлса шундай қабул қилинади.

Сатрларни бирлаштириш

PHP да сатрлар билан ишлаш учун энг қулайи конкатенация(сатрларни бирлаштириш) амали. Конкатенация икки ёки ундан ортиқ объектларни бир бутун қилиб бирлаштириш деган маънони билдиради. Сатрлар билан ишлаш учун иккита амал мавжуд: конкатенация амали(.) ва конкатенация билан тенглик амали(=).

5.2 - жадвал. Сатрлар билан ишлаш амали

Белгиси	Номи	Мисол	Натижа
.	Конкатенация	<code>\$a="abc"."def"</code>	\$a ўзгарувчиси abc ва def ларнинг бирлаштириш натижасига тенг.
.=	Конкатенация билан тенглик	<code>\$a.="ghijkl"</code>	\$a ўзгарувчиси унинг мавжуд қиймати билан “ghijkl” сатрини бирлаштириш натижасига тенг.

Конкатенация амали ёрдамида сатрлар билан сатрларни, ўзгарувчилар билан сатрларни бирлаштириш мумкин. Бу амалдан браузерга матнларни чиқазишда тез-тез фойдаланилади ва echo буйруғини бир неча маротаба беришга зарурат туғилмайди.

Сатрлар билан ишлаш амалларига мисоллар келтирамиз:

5.1- мисол. Сатрлар билан ишлаш.

```
<? $a="Olma"." va Nok";
// $a "Olma va Nok" satrli qiymatga teng
$a.=" mevalar";
// $a "Olma va Nok mevalar" satrli qiymatga teng
?>
```

5.2-мисол. Сатрлар билан ишлаш.

```
<?
$name="Lola";
$age=22;
echo "$name";
echo $name." talaba<br>";
?>
```

Ёки

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "$name talaba<br>";
?>
```

деб ҳам ёзиш мумкин. Иккала ҳол ҳам бир хил натижа беради.

Сонли ўзгарувчилар устида ҳам конкатенация амалини бажариш мумкин.

5.3- мисол. Сатрлар билан ишлашда конкатенация амали

```
<?
$a=39;
```



```
$b=10;
$n=$a.$b;
echo $n;
?>
Natija:3910
```

Демак, (.) конкатенация амали \$a ва \$b ўзгарувчиларни қийматларини бир-бирига қўшиб қўймайди, балки уларни сатрий катталиқ сифатида кетма-кет жойлаштиради.

Энди ўзгарувчи қийматлари, матнларни ва HTML тегларини биргаликдаги комбинациясини чиқаришини кўрамиз.

5.4- мисол. Сатрлар билан ишлаш

```
<?
$name="Lola";
$age=22;
echo "<br>$name Fizika-
matematika fakultetida o'qiydi. <br>Uni yoshi $age da."
?>
Natija:
Lola Fizika-matematika fakultetida o'qiydi.
Uni yoshi 22 da.
```

Ўзгарувчиларга баъзи ҳолларда қиймат берилаётганда қуйидаги кабиларга дуч келиш мумкин.

```
<?
$a= "Lola "Ulug`bek stipendiyasi" sovrindori";
echo $a;
?>
```

Агар ўзгарувчига шу тартибда қиймат бериладиган бўлса, у ҳолда браузерда хатолик ҳақида хабар чиқарилади, яъни иккиталик қўштирноқ ичида иккиталик қўштирноқ бу ҳолатда ишлатилмайди. Агар иккинчи қўштирноқнинг очилиши ва ёпилиши олдиан \ белгиси қўлланилса, у ҳолда хатолик юзага келмайди.

```
<?
$a= "Lola \"Ulug`bek stipendiyasi\" sovrindori";
echo $a;
?>
```

PHP саҳифада HTML тегларини қўлланилишини кўриб чиқамиз.

```
<?
echo "<p align=\"center\">Fizika - matematika fakulteti</p>";
?>
```

HTML тегини PHP саҳифада бундай ҳолда қўллашимиз хатоликка олиб келади. Бу ерда юқоридаги сингари иккинчи қўштирноқ очилиши ва ёпилиши олдиан \ белгиси ишлатилса бўлади, лекин уни ўрнига қуйидаги ҳолда ёзиш ҳам мумкин.

```
<?
echo "<p align='center'>Fizika - matematika fakulteti</p>";
?>
```

5.3. Арифметик амаллар.

Биз иккита амал билан танишимиз: (=) тенглик амали ва (.) сатрларни бирлаштириш амали. Энди арифметик амаллар билан танишиб чиқамиз. Ўзгарувчилар устида барча арифметик амалларни бажариш мумкин.

5.3 - жадвал. Арифметик амаллар

Белгиси	Кўриниши	Номи
+	$\$a+\b	қўшиш
-	$\$a-\b	айириш
*	$\$a*\b	кўпайтириш
/	$\$a/\b	бўлиш
%	$\$a\%\b	модул олиш

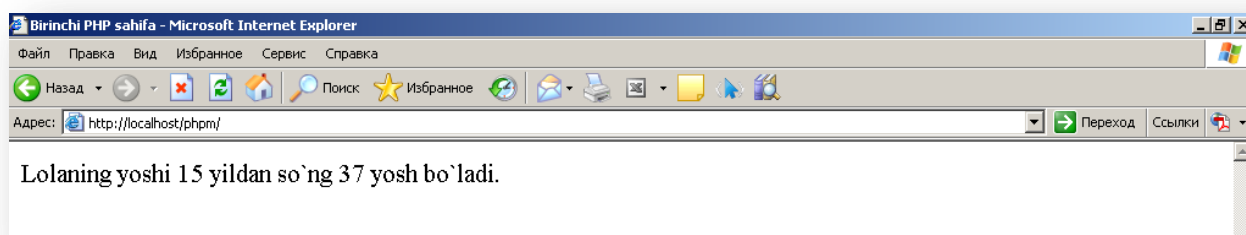
Қўшиш ва айириш амаллари оддий бажарилади. Уларни натижаси $\$a$ ва $\$b$ ўзгарувчиларда сақланаётган қийматларнинг йиғиндиси ва айирмасига тенг. Айириш амалидан унар амал сифатида ҳам фойдаланилса бўлади. Бунда ўзгарувчига манфий қиймат берилади, $\$a=-1$.

Кўпайтириш ва бўлиш амаллари ҳам оддий ишлатилади. Эътибор беринг, белгилар анъанавий математикадаги белгиларни, яъни юлдузча кўпайтириш амалини белгисини ва эгилган таёқча бўлиш амалини белгисини беради.

5.5 – мисол. Қўшиш амали.

```
<?
$name="Lola";
$age=22;
$n=15;
$sum=$age+$n;
echo $name. "ning yoshi $n yildan so'ng $sum yosh bo'ladi."
?>
```

Натижа:



5.6 – мисол. Модул олиш амали.

```
<?
$a=39;
$b=10;
$natija=$a%$b;
/* Bu yerda $natija o'zgaruvchisi 39 ni 10 ga bo'lingandagi qoldiq 9 ga teng bo'ladi. */
?>
Натижа:
9
```

5.6 – мисол. Арифметик амалларнинг ишлатилиш

```
<?
$a=10;
$b=17;
$qushish=$a+$b;
$ayirish=$a-$b;
echo $qushish." - $a va $b ning yig'indisi.<br>";
echo $ayirish." - $a va $b ning ayirmasi. <br>";
$a=20;
$b=5;
$kupay=$a*$b;
$bulish=$a/$b;
echo $kupay."- $a va $b ni kupaytmasi.<br>";
echo $bulish."- $a ni $b ga bulinmasi.";
?>
```

Натижа:
 27 - 10 va 17 ning yig'indisi.
 27 - 10 va 17 ning ayirmasi.
 100 - 20 va 5 ni kupaytmasi.
 4- 20 ni 5 ga bulinmasi.

Арифметик амалларни тенглаштириш амали билан бирга бошқача кўринишда ҳам ишлатиш мумкин. **Масалан,**

$b=12+(a=4);$

Бу ҳолда b ўзгарувчиси 16 га тенг бўлади. Чунки, a ўзгарувчиси 4 қийматни қабул қиляпти ва 12 га қўшиляпти, шунинг учун $12+4=16$ га тенг.

Оддий тенглаштириш амалларидан ташқари комбинацияланган тенглаштириш амаллари ёки қиймат узатиш амаллари ҳам мавжуд.

Қиймат узатиш амали ҳар бир арифметик амал ва конкатенация амали учун ҳам мавжуд.

5.4 - жадвал. Қиймат узатиш амали

Белгиси	Ишлатилиши	Эквиваленти
$+=$	$a+=b$	$a=a+b$
$-=$	$a-=b$	$a=a-b$
$*=$	$a*=b$	$a=a*b$
$/=$	$a/=b$	$a=a/b$
$\%=$	$a\%=b$	$a=a\%b$
$.=$	$a.=b$	$a=a.b$

Масалан,

$a+=8$

бу ёзув

$a=a+8$ ёзувига эквивалент.

инкремент(++) ва декремент(--) ёрдамчи амаллари дастурда ўзгарувчи қийматини 1 га камайтириш ёки ошириш учун хизмат қилади.

5.5 - жадвал. Инкремент ва декремент амаллари

Мисол	номи	Натижа
$++a, a++$	Инкремент	a ўзгарувчисини 1 га оширади
$--a, a--$	декремент	a ўзгарувчисини 1 га камайтиради

Префиксли ва суффиксли инкремент(++), ва декремент(--), амаллари += ва -= амалларига ўхшаш. **Масалан,**

```
<?
$a=4;
echo ++$a;
?>
Natija:5
```

Иккинчи сатрда префиксли инкремент амали ишлатилляпти, чунки \$a дан олдин (++) белгиси мавжуд. Натижада \$a нинг қиймати 1 га оширилади ва оширилган натижа чиқазилади. Бу ҳолда, \$a нинг қиймати 5 га оширилади ва 5 экранга чиқазилади, яъни \$a ўзгарувчининг қиймати аввал оширилиб, кейин натижага берилади.

Агар (++) белгиси \$a дан кейин ишлатилса, демек суффиксли инкремент амали ишлатилляпти, у ҳолда натижа бошқача бўлади. **Масалан,**

```
<?
$a=4;
echo $a++;
?>
Natija:4
```

бу ҳолда амал тескари тартибда бажарилади. Аввал \$a қийматни қабул қилади ва натижага чиқзади, шундан сўнг уни қиймати 1 тага оширилади. Бунда натижага 4 сони чиқазилади, кейин \$a нинг қиймати 5 га тенг бўлади.

Инкремент ва декремент амаллари бир-бирига ўхшаш. Фақатгина, Префиксли ва суффиксли инкремент амалида \$a нинг қиймати 1 тага оширилса, декремент амалида эса \$a нинг қиймати 1 тага камайтирилади.

5.4. Мантикий бинар амаллар.

Мантикий амаллар PHP да ихтиёрий дастур тузиш жараёнида, ўзгарувчи қийматлари билан боғлиқ бўлган ҳолларда муҳим рол ўйнайди. Мантикий амаллар дастурда буйруқлар бажарилиш тартибини бошқариш имкониятини беради ва бошқарувчи конструкциялар (шартли if оператори ва for, while цикллари) билан тез-тез ишлатилади.

Мантикий шартларни натижаларини бирлаштириш учун мантикий амаллар ишлатилади. Мисол учун \$a ўзгарувчиси 0 ва 100 орасидаги қийматларни қабул қилади дейлик. Бу ҳолда AND(BA) амалини ишлатиб, \$a>=1 ва \$a<=100 шартларни текширамыз.

\$a>=1 && \$a<=100

PHP технологияси AND(BA), OR(ЁКИ), XOR(ёки мустасно) ва NOT(ЭМАС) каби мантикий амалларни ишлатади.

5.6 - жадвал. Мантикий амаллар.

Белгиси	номи	ишлатилиши	натижа
&&	Ва	\$a&&\$b	\$a ва \$b ўзгарувчиларнинг иккаласи ҳам рост қиймат қабул қилса, рост, акс ҳолда ёлғон
AND	Ва	\$a AND \$b	&& каби мантикий амал
	Ёки	\$a \$b	\$a ёки \$b ўзгарувчиларнинг иккаласидан бирортаси рост қиймат қабул қилса, рост, акс ҳолда ёлғон
OR	ёки	\$a OR \$b	каби мантикий амал
XOR	ёки	\$a XOR \$b	Ўзгарувчиларнинг иккаласидан фақатгина биттаси рост бўлса, рост
!	Эмас	!\$a	Агар \$a ёлғон бўлса, натижа рост ва аксинча

5.5. Тенглик амали

Тенглик амали $==$ (иккита тенг белгиси) иккита қийматни бир-бирига таққослайди ва уларни тенглигини текширади. масалан,

$\$a == \b ;

$\$a$ ва $\$b$ ўзгарувчиларда сақланаётган қийматларнинг тенглигини текшириш учун ишлатилади. Агар иккала ўзгарувчининг қиймати бир-бирига тенг бўлса натижа true ни, акс ҳолда false ни беради.

Бу амални тенглаштириш амали билан адаштириш мумкин. Бу вақтда дастурда хатолик юзага келганлиги ҳақида хабар чиқмаслиги мумкин, лекин исталган натижага эришиб бўлмайди. Умумий ҳолларда нол бўлмаган қиймат true ни, нол қийматлар эса – false ни беради. Қуйидаги ҳолатни кўриб чиқамиз:

$\$a = 5$;

$\$b = 7$;

$\$a = \b ни текширадиган бўлсак, натижа true ни беради. Нима учун?

Чунки $\$a = \b қийматга эга, яъни ўнг томондаги операнднинг қиймати айни вақтда 7 га тенг, бу эса нол бўлмаган қиймат. Шунинг учун ифода натижаси true ни беради.

Агар $\$a == \b ифодани текширадиган бўлсак, бу ҳолда натижа false ни беради, чунки $\$a$ ва $\$b$ нинг қиймати бир-бирига тенг эмас.

Демак, $=$ ва $==$ белгилари икки ҳил натижа берди. Шунинг учун дастур кодлари ёзаётган вақтда қайси таққослаш амали кераклигини аниқлаб, кейин бу белгиларни ишлатиш керак.

5.7 - жадвал. Тенглик амаллари.

Амал белгиси	Номи	Ишлатилиши	Натижа
$==$	Тенг	$\$a == \b	$\$a$ ва $\$b$ тенг бўлса, рост
$===$	Айнан тенг	$\$a === \b	$\$a$ ва $\$b$ айнан тенг(тенг ва бир хил типли) бўлса, рост
$!=$	Тенг эмас	$\$a != \b	$\$a$ ва $\$b$ тенг бўлмаса, рост
0	Тенг эмас	$\$a 0 \b	$\$a$ ва $\$b$ тенг бўлмаса, рост

5.6. Муносабат(таққослаш) амали.

Таққослаш амали иккита қийматни бир-бири билан солиштириш учун ишлатилади. Таққослаш амаллари худди мантиқий амаллар каби ишлайди, дастур мантиғини бошқаради ва икки ёки ундан ортиқ ўзгарувчиларни таққослаб хулоса қабул қилади. Бу амаллар ишлатилганда таққослаш натижаси true(рост) ёки false(ёлғон) мантиқий қийматларга тенг бўлади.

5.8 - жадвал. Таққослаш амаллари

Амал белгиси	Номи	Ишлатилиши	Натижа
<	Кичик	$a < b$	a ўзгарувчиси b дан кичик бўлса, рост
>	Катта	$a > b$	a ўзгарувчиси b дан катта бўлса, рост
<=	Кичик ёки тенг	$a \leq b$	a ўзгарувчиси b дан кичик ёки тенг бўлса, рост
>=	Катта ёки тенг	$a \geq b$	a ўзгарувчиси b дан катта ёки тенг бўлса, рост
$(a=12)?5:-1$	Тернар амал		Агар a ўзгарувчиси 12 га тенг бўлса, 5 қийматини, акс ҳолда -1 қийматини чиқазади

Тернар амали

Тернар амали (?:) худди C даги каби ишлатилади.

Синтаксиси:

шарт ? шарт рост бўлгандаги қиймат: шарт ёлғон бўлгандаги қиймат

Тернар амали if-else операторига ўхшаш.

5.7 – мисол. Тернар амалининг ишлатилиши

```
<?
($natija>50 ? "Imtihondan o'tdingiz" : " Imtihondan o'tmadizingiz")
?>
```

Бу ифода талабаларнинг имтиҳондан ўтган ёки ўтмаганлигини аниқловчи шарт сифатида ишлатилади.

5.7. Хатоларни яшириш амали

Хатоларни яшириш амали @ бирор қийматга эга ёки ишлаб чиқиладиган ёзув, ихтиёрий ифода олдидан ишлатилиши мумкин.

5.8 – мисол. Хатоларни яшириш амали

```
<?
$a=@ (57/0)
?>
```

Бу қатор @ белгисиз ёзилса, нолга бўлиш мумкин эмаслиги ҳақидаги огоҳлантиришни беради. Агар @ белгиси ишлатилса, у ҳолда бу хатолик ҳақидаги хабар яширилади. Агар PHP track_errors функцияси билан ўрнатилган бўлинса, хатолар ҳақидаги хабарлар \$php_errormsg глобал ўзгарувчисига сақланади.

VI БОБ. PHP ёрдамида сўровларни қайта ишлаш

Бу бобда серверга маълумотларни жўнатиш методлари ва уларни PHP ёрдамида қайта ишлаш ўрганилади. Мижоз-сервер технологиясининг асосий тушунчалари кўрилади. Post ва Get методларининг қисқача тавсифи берилади. HTML-формада маълумотларни олиш механизми ва уларни PHP ёрдамида қайта ишлаш билан танишилади.

6.1. Мижоз-сервер технологияси асослари

Биз маърузанинг бошида PHP- сервер ёрдамида қайта ишланадиган скрипти тил эканлигини таъкидлаб ўтгандик. Ҳозир биз сервер нима у, қандай функцияларни бажаради ва умуман қанақа серверлар бўлишини аниқлаб оламиз. Гап сервер ҳақида кетаётганида ўз-ўзидан мижоз тушунчаси эсга тушади. Чунки бу икки тушунча бир-бири билан узвий боғлиқ. Уларни мижоз-сервер компьютер архитектураси бирлаштириб туради. Одатда сервер ҳақида гапирилатганида мижоз-сервер архитектураси назарда тутилади, мижоз ҳақида гапирилатганида ҳам шу архитектура назарда тутилади. Унинг мазмуни шундаки, икки ички система функциясини ажратиш қандайдир ҳаракатларни бажарётган сўровни жўнатаётган мижоз ва шу сўровни бажарётган сервер ёрдамида амалга оширилади. Мижоз ва сервер ўртасида ўзаро алоқа TCP/IP ва z39.50 каби махсус стандарт протоколлар иштирокида амалга оширилади. Аслида протоколлар жуда ҳам кўп, улар даражасига қараб ажратилади. Бизнинг дастурли масалаларимизда фақат HTTP протоколи керак бўлгани учун сал кейинроқ уни кўриб чиқамиз. Ҳозирча мижоз-сервер архитектурасига қайтамиз ва мижоз - сервер қандай тушунча эканлигини аниқлаб оламиз.

Ҳар хил жараёнларни бажарилишини назорат қилиб турадиган дастурлар тўпламига сервер дейилади. Ушбу дастурлар мос равишда қандайдир компьютерга ўрнатилган бўлади. Кўпинча сервер ўрнатилган компьютерни сервер деб аташади. Мижоз сўрови бўйича қандайдир аниқ жараённи ишга тушириш ва уни ишини натижасини мижозга жўнатиш компьютер-сервернинг асосий функциясига киради.

Серверни исталган хизматидан фойдаланилган жараёнга мижоз дейилади. Фойдаланувчи ҳам дастур ҳам мижоз бўла олиши мумкин. Мижознинг асосий вазифаси илова талаб қилаётган вақтда сервер билан алоқада иловани бажариш ва уни амалга оширишдир. Шу билан бирга мижоз фойдаланувчига ишини мантиғини ва керак бўлганда уни масаласини серверга жўнатишни, ҳамда илова билан ишлаш учун интерфейс яратиб беришни таъминлаши керак.

Мижоз ва сервер ўртасидаги ўзаро алоқа мижоз ташаббусига кўра бошланади. Мижоз хизмат турини танлайди, сеансни ўрнатади, ўзига керак бўлган натижани олади ва иш тугагани ҳақида хабар қилади.

Кўпинча бир нечта мижозлар бир вақтнинг ўзида битта сервер хизматларидан фойдаланади. Шунинг учун ҳар бир сервер маълумотлар хавфсизлигини таъминлаши ва катта ишлаб чиқаришга эга бўлиши керак.

Сервер компьютерга уланаётган вақтда локал ёки глобал тармоқ бўлса мақсадга мувофиқроқ бўлади. Серверни алоҳида турадиган компьютерга ҳам улаш мумкин (унда у бир вақтнинг ўзида ҳам сервер ҳам мижоз вазифасини бажаради).

Серверларнинг ҳар-хил турлари мавжуд. Қуйида уларнинг айримлари келтирилган:

- **Видеосервер.** Бундай сервер тасвирларни қайта ишлашга, видеоматериалларни, видео ўйинларни ва ҳ.к.ларни сақлаш учун мўлжалланган. Видеосервер ўрнатилган компьютер катта хотирага ва юқори ишлаб чиқаришга эга бўлиши керак.

- **Қидирув** сервери - интернетда маълумотни қидириш учун мўлжалланган.

- **Почта** сервери электрон почта орқали жўнатилган сўровларга жавоб бериш хизматларига эга.

- **WWW** сервери – интернетда ишлаш учун мўлжалланган.

- **Маълумотлар базаси** сервери - маълумотлар базасидаги сўровларни қайта ишлайди.

- **Маълумотлар хавфсизлиги** сервери маълумотлар хавфсизлигини таъминлайди (масалан, паролларни идентификациялаш муҳитини мужассамлайди)

- **Иловалар** сервери амалий (прикладной) жараёнларни бажариш учун мўлжалланган. Бир томондан масалани олиш учун мижоз билан ўзаро алоқада бўлади, иккинчи томондан маълумотларни қайта ишлаш учун кераклисини олиб маълумотлар базаси билан ишлайди.

- **Масофадан туриб фойдаланиш** сервери маълумотларга жамоавий удалённий доступни таъминлайди.

- **Файл** сервери қидирув, сақлаш, маълумотларни архивлаш ва бир вақтнинг ўзида уларга бир нечта фойдаланувчи ишлай олиш хизматларини ва ресурсларни функциялашни таъминлайди.

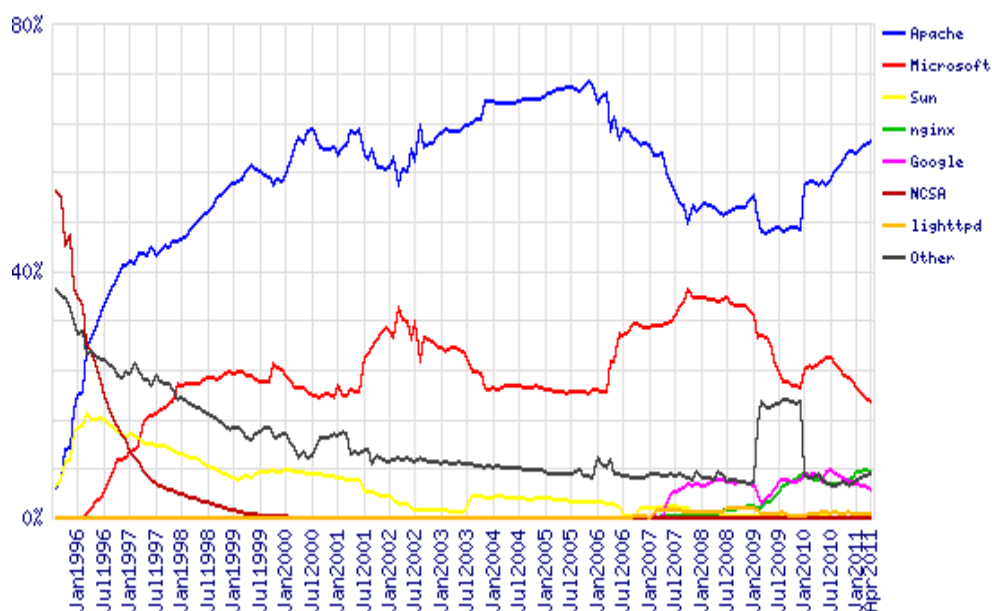
Одатда компьютер-серверда бир вақтнинг ўзида бир нечта сервер дастурлари ишлайди. Бири электрон почта билан, иккинчиси файлларни тақсимлаш билан, учинчиси web саҳифалар билан ишлайди. Ҳамма серверлар туридан бизни кўпроқ WWW сервери қизиқтиради. Одатда уни http-сервер, web-сервер ёки оддий қилиб сервер деб аташади. Веб-сервер ўзида нимани мужассамлаштиради? **Биринчидан**, бу информацион ресурслар омбори. **Иккинчидан**, бу ресурслар фойдаланувчиларга интернет стандартларига мос ҳолда берилади ва сақланади (масалан, HTTP маълумотларини узатиш протоколи сингари). Бу протокол билан боғлиқ ҳолда маълумотларни қандай тасвирлаш мумкинлигини кейинроқ кўриб ўтамыз. Веб - сервердаги саҳифалар билан ишлаш браузер дастурлари (масалан, IE, Opera, Mozilla) ёрдамида амалга оширилади. Бунда браузер дастури HTTP протоколи мос

равишда яратилган сўровларни серверга жўнатади. Сервер эса топшириқ, бошқача айтганда суровни қайта ишлайди ва бажаради, ўз навбатида қайта ишлаш жараёнида бошқа серверлар (бу ерда дастур серверлари назарда тутиляпти, масалан МББТ сервери, файл сервери, почта сервери кабилар) билан алоқа ўрнатиши мумкин.

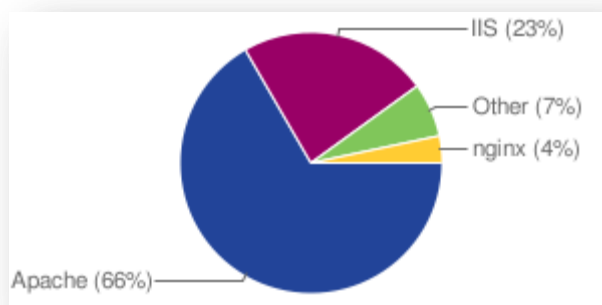
Бундан кейин “сервер” деганда web-сервер ни тушунамиз.

Веб-серверга мисол қилиб Apache гуруҳининг Apache серверини, Microsoft компаниясининг Internet Information Server (IIS), Sun Microsystems фирмасининг Sun, Rambler компанияси учун Игорь Сысоев томонидан яратилган nginx(engine x), Google компаниясининг Google, NCSA(National Center for Supercomputing Applications) ташкилотининг NCSA HTTPd, lighttpd(lighty) ва бошқаларни келтириш мумкин.

6.1 – 6.2 - расмда ва 6.1 - жадвалда Интернет ва NetCraff нинг барча доменлари ичида Веб-серверлардан фойдаланишнинг статистик кўрсаткичи келтирилган.



6.1- расм. Веб-серверлардан фойдаланиш кўрсаткичи



6.2- расм. Веб-серверлардан Интернетда фойдаланиш кўрсаткичи

6.1 - жадвал. Веб-серверларни қайта ишловчилар

Ишлаб чиқувчи	Март 2011	фоизда	Апрель 2011	фоизда	Ўзгаришлар
Apache	179,720,332	60.31%	191,139,966	61.13%	0.82
Microsoft	57,644,692	19.34%	58,867,097	18.83%	-0.52
nginx	22,806,060	7.65%	23,463,669	7.50%	-0.15
Google	15,161,530	5.09%	14,690,422	4.70%	-0.39
lighttpd	1,796,471	0.60%	1,862,963	0.60%	-0.01

Келтирилган жадвалдан Apache сервери олдинги ўринларни эгаллагани кўриниб турибди. web-серверлар ҳақида сўз борганда бошқа серверлар кўрсатилмаган бўлса, Apache назарда тутилади. Уни шахсий компьютерга қандай қилиб юклаш ҳақида айтиб ўтгандик.

«Мижоз – сервер» технологияси

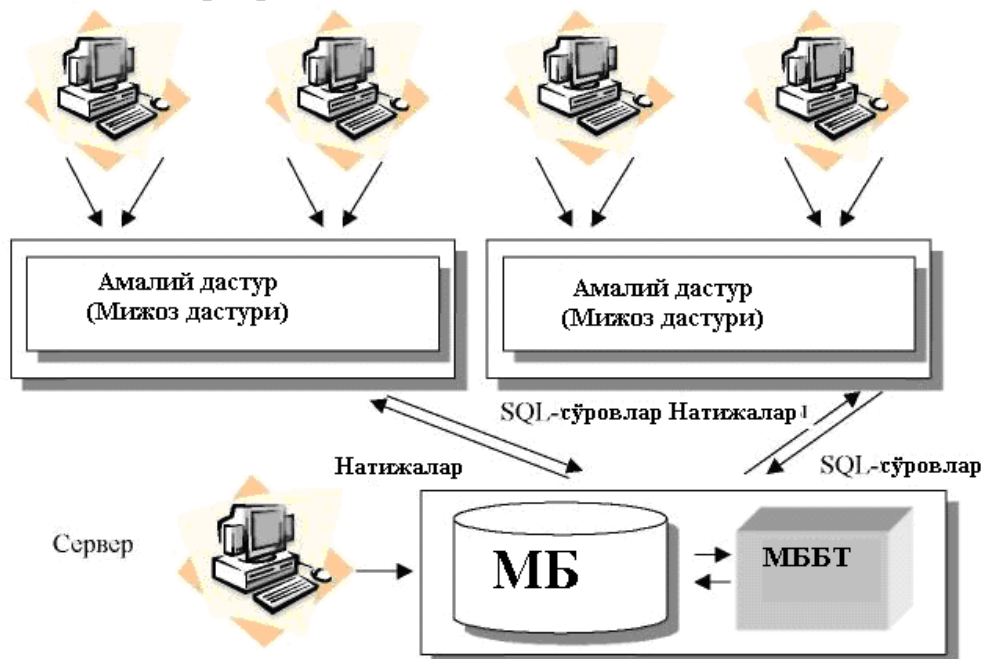
«Мижоз – сервер» технологиясини қўллаш тармоққа бирлаштирилган компьютерларга асосланади, бу компьютерлардан бири махсус бошқарув функцияларини бажаради (тармоқ сервери бўлади).

«Мижоз – сервер» архитектураси функцияларни фойдаланувчи дастури (мижоз деб аталувчи) ва сервер функцияларига ажратади. Мижоз–дастур МБ сақланаётган серверга SQL (Structured Query Language) структуралаштирилган сўровлар тилидаги сўровни жўнатади. SQL реляцион МБ ларнинг халқаро стандарти ҳисобланади. Масофадаги сервер сўровни қабул қилади ва фаол бўлган SQL-серверга беради. SQL-сервер – бу масофадаги маълумотлар базасини бошқарувчи махсус дастур. SQL-сервер сўровни талқин қилади, бажаради, сўров натижаларини расмийлаштиради ва мижоз –дастурига узатади. Бу жараёнда мижоз компьютерининг ресурслари сўровни бажаришда иштирок этмайди. Мижоз компьютер фақат сервер МБ га сўров жўнатади ва натижани қабул қилади. Кейин натижани зарур шаклда талқин қилади ва фойдаланувчига тасвирлаб беради. Мижоз дастурга сўровни бажариш натижаси жўнатилади, яъни тармоқ орқали фақат мижозга зарур бўлган маълумотлар жўнатилади. Натижада тармоқ юкламаси пасаяди. Қолаверса, сўров маълумотлар базаси сақланаётган жойда (серверда) бажарилади ва катта ҳажмли маълумотлар пакетини тармоқ орқали жўнатиш зарур бўлмайди. Бундан ташқари, SQL-сервер агар имкони бўлса минимал вақт ва сарф-харажатларда бажарилиши учун сўровни оптималлаштиради. Бундай тизим архитектураси 6.2-расмда тасвирланган.

Буларнинг барчаси тизимни тезлигини оширади ва сўров натижасини кутиш вақтини қисқартиради. Сервер томонидан сўровлар бажарилганда маълумотларни хавфсизлик даражаси анча юқори бўлади. Чунки маълумотларни бутунлик коидаси сервердаги маълумотлар базасида аниқланади ва ушбу маълумотлар базасидан фойдаланувчи барча дастурлар учун ягона ҳисобланади. Шу тарзда бутунликни қўллаб-қувватлашда қарама-қарши коидаларни вужудга келишини олди олинади. SQL –серверлар томонидан қўллаб-қувватланадиган ишончли транзакциялар аппарати битта

маълумотни бир вақтда турли фойдаланувчилар томонидан ўзгартирилишига йўл қўймайди ва МБ да аварияли тугаган ўзгаришларни бекор қилиш имконияти мавжуд.

Мижоз компьютерлари



6.3 – расм. «мижоз– сервер» архитектураси

«Мижоз – сервер» архитектураси қуйидагича қурилган:

- Маълумотлар базаси файллар тўплами шаклида махсус компьютернинг (тармоқ сервери) каттиқ дискида жойлашади.
 - МББТ ҳам тармоқ серверида жойлашади.
 - Мижоз компьютерларидан иборат бўлган локал тармоқ мавжуд бўлиб, ҳар бир компьютерда МБ билан ишловчи мижоз дастури ўрнатилган.
 - Ҳар бир мижоз компьютерда фойдаланувчилар дастурини ишга тушириш имконига эга. Дастурнинг фойдаланувчи интерфейси ёрдамида фойдаланувчилар маълумотларни танлаш/янгилаш учун сервердаги МБ га мурожат қилади. Мурожат қилиш учун махсус SQL сўровлар тили ишлатилади, яъни серверга фақат сўров матни жўнатилади.
 - МББТ да сервердаги МБ нинг физик структураси ҳақидаги маълумотлар сақланади.
 - МББТ сервердаги маълумотларга мурожатни қайд қилади ва серверда маълумотларни қайта ишлаш амаллари бажарилади ва натижа мижоз компьютерига юборилади. Шу тарзда МББТ натижаларни мижоз дастурига юборади.
 - Дастур эса ўзидаги фойдаланувчи интерфейси ёрдамида натижаларни тасвирлаб беради.
- Бажариладиган функциялар сервер ва мижоз ўртасида қандай тақсимланганини кўриб чиқамиз:
- мижоз дастури функциялари:
 - Сўровларни серверга жўнатиш.
 - Сервердан олинган сўров натижаларини талқин қилиш.

– Натижаларни бирор шаклда фойдаланувчига кўрсатиш (фойдаланувчи интерфейси).

•Сервер функциялари:

- Мижоз –дастурлардан сўровларни қабул қилиш.
- Сўровларни талқин қилиш.
- МБ даги сўровларни оптималлаштириш ва бажариш.
- Натижаларни мижоз-дастурга жўнатиш.
- Хавфсизлик тизимини ва мурожаатни чегаралаш.
- МБ бутунлигини бошқариш.
- Кўп фойдаланувчили иш режими стабиллигини таъминлаш.

6.2. HTTP протоколи ва маълумотларни серверга жўнатиш методлари.

Интернет фойдаланувчи ва тармоқ ўртасидаги интерфейсни амалий даражагача таъминлайдиган, физик даража билан иккиламчи маълумотни узатишда физик аспектларга боғлиқ кўп даражали принцип асосида қурилган.

HTTP - бу, Интернетда гиперматнли маълумотни алмаштириш учун ишланган амалий даража протоколи.

HTTP серверга жўнатадиган тўлиқ сўровлар кўрсатмалари методлари йиғиндисидир. Бу методлар ресурсни кўрсатиш учун жўнатмалар (ссылкалар) тартибига асосланади, ушбу метод қўлланилиши керак бўлганда ресурсни жойлашган жойини(Universal Resource Locator, URL) ёки уни универсал номини (Universal Resource Name, URN) топадиган универсал идентификаторлари (Universal Resource Identifier, URI) қўлланилади.

HTTP протоколи қўлланилаётган вақтда тармоқ ичида хабар жўнатиш худди Интернет (RFC-822)почта хабари форматига ёки MIME(Multipurpose Internet Mail Exchange) хабарлар форматига ўхшаш форматда юборилади.

HTTP SMTP(электрон почта протоколи), NNTP(хабарлар жўнатиш протоколи), FTP(файллар жўнатиш протоколи), Gopher ва WAIS каби мавжуд Интернет протоколларига рухсат берувчи ҳар-хил фойдаланувчи дастурлари ва шлюз дастурлари коммуникацияси учун қўлланилади. HTTP шундай шлюзларга оралиқ сервер-дастурлари (proxy) да маълумотларни йўқотишларсиз жўнатиш учун яратилган.

Протокол сўров/жавоб принципини амалга оширади. Мижоз сўраётган дастур жавоб бераётган дастур-сервердан қуйидагиларни сўрайди:

- Рухсат методи;
- URI манзили;
- Протокол версияси;
- Узатилаётган маълумотлар типи ҳақидаги хабарни (MIME даги формага ўхшаш), сўров жўнатган мижоз ҳақида маълумотни;

Сервер жавоби:

- Протокол версияси ва қайтиш коди(муваффақиятли ёки хатоли) чиқувчи қатор ҳолати;
- Сервер ҳақидаги маълумот, метамаълумот (хабар мундарижаси ҳақида маълумот) ва хабар танасини мужассамлаштирган хабар.

Мижоз ва сервер ўртасида алоқа ўрнатилаётганда ким очиши ёки ёпиши протоколда кўрсатилмайди. Қоидага асосан алоқа ўрнатишда мижоз очади, сервер бўлса жавоб жўнатилгандан кейин ўртадаги алоқани узади.

Серверга сўров қандай шаклда жўнатилишини тўлиқроқ кўриб чиқамиз.

6.3. Мижоз сўрови шакли

Мижоз серверга икки хил: тўлиқ ёки қисқартирилган шаклда сўров жўнатади. Ушбу сўровлар мос ҳолда тўлиқ сўров ва оддий сўров деб айтилади.

Оддий сўров рухсат методи ва ресурс манзилини мужассамлаштиради. Буни тўлиқроқ шундай ёзиш мумкин:

< Оддий-сўров > := <Метод> < пробел белгиси>

<Сўралаётган -URI> <янги қатор белгиси>

Метод тариқасида GET, POST, HEAD, PUT, DELETE ва бошқаларни кўрсатиш мумкин. Улар ичида кенгроқ тарқалганлари ҳақида кейинроқ айтиб ўтамиз.

Оддий сўровга мисол:

GET <http://phpbook.info/>

Бу ерда GET – рухсат методи, яъни сўралаётган ресурсга қўлланилиши керак бўлган метод, <http://phpbook.info/> эса сўралаётган ресурснинг URL-манзили.

Тўлиқ сўров эса қатор ҳолати, бир нечта сарлавҳалар(сўров сарлавҳаси, умумий сарлавҳа ва мундарижа сарлавҳаси) ва сўров танасидан иборат бўлиши мумкин. Тўлиқ сўровнинг умумий кўринишини қуйидагича ёзиш мумкин:

< тўлиқ сўров > := < қатор ҳолати >

(<умумий сарлавҳа> < сўров сарлавҳа>

<мундарижа сарлавҳа>)

<янги қатор белгиси>

[<сўров мундарижаси>]

Бу ерда тўртбурчак қавслар муқобил вариантнинг вертикал чизиқлар устида кўрсатилган сарлавҳанинг кераксиз элементларини англатади. Бу ерда <қатор ҳолати> Сўров методи ва URI ресурсини(худди оддий сўров каби) ва бундан ташқари HTTP протоколининг қўлланилган версиясини ўз ичига олади. Масалан, ташқи дастурни чақириш учун қуйидаги қатор ҳолатига таъсир қилиш мумкин:

POST <http://phpbook.info/cgi-bin/test HTTP/1.0>

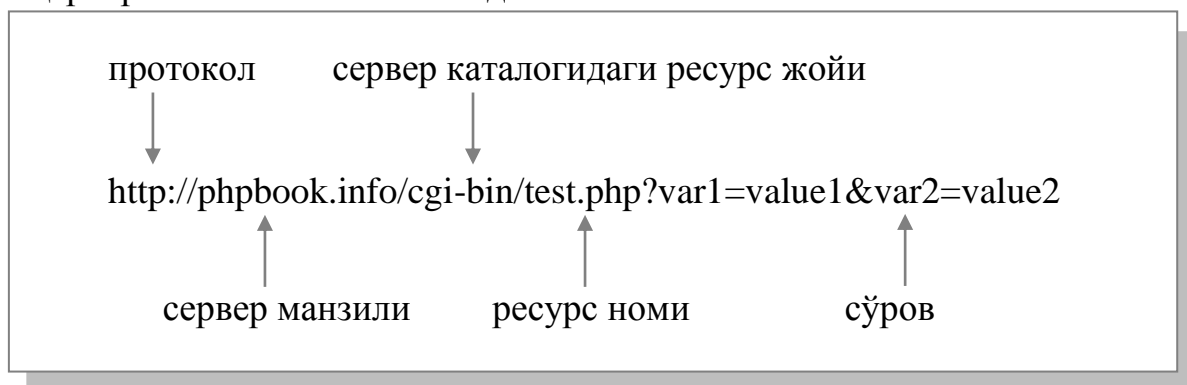
Бу ҳолда POST методи ва HTTP протоколининг 1.0 версияси қўлланиляпти.

Сўровнинг икки шаклида ҳам сўралаётган ресурснинг URI си муҳим ўрин эгаллайди. Кўпинча URL-ресурс манзили сифатида URI ишлатилади. Серверга мурожаат қилаётган вақтда URLни тўлиқ шаклини ҳам, қисқартирилган шаклини ҳам қўллаш мумкин.

Тўлиқ шакли протоколга рухсат тоифасини, ресурс сервери манзилини ва серверда манзил ресурсини ўз ичига олади(6.4 - расм).

Қисқартирилган шаклда эса протокол ва сервер манзили сервер асосидан манбанинг фақатгина қаерда жойлашганлигини кўрсатади. Тўлиқ шаклда агар мумкин бўлса, сўровни бошқа серверга жўнатиб юбориш ҳам

қўлланилади. Агар битта сервер билан ишланаётган бўлса кўпинча қисқартирилган шакл ишлатилади.



6.4 - расм. URLнинг тўлиқ шакли.

Энди сўровлар жўнатишнинг кенг тарқалган методларини кўриб чиқамиз.

6.4. Методлар

Юқорида айтиб ўтганимиздек мижознинг серверга исталган сўрови методни кўрсатиш билан бошланиши керак. Метод мижоз сўровининг мақсади ҳақида хабар беради. HTTP протоколи жуда ҳам кўп методларни қўллайди, лекин уларни ичидан фақат учтаси кўп қўлланилади: POST, GET ва HEAD. GET методи ресурс сўровида URI ёрдамида идентификация қилинган исталган маълумотни олиши мумкин. Агар URI дастурга кўрсатаётган бўлса, унда дастур ишининг натижаси сифатида дастурнинг матни эмас ўзи қайтади (агар матн дастур ишининг натижаси бўлмаса). Қўшимча маълумот сўровни қайта ишлаш учун сўровни ўзига қўйилади. GET методини ресурс қисми майдонида қўллаётган вақтда сўралган маълумот мос равишда қайтади (масалан, HTML матни хужжати).

GET методининг ҳар-хил шакллари мавжуд - шартли GET. Бу метод агар шарт бажарилган бўлса, сўров сарлавҳаси майдонида if-Modified-Since нинг мавжудлиги ҳақида сўровга жавоб юбориш кераклигини хабар беради. Аниқроқ гапирадиган бўлсак, ресурс if-Modified-Since да кўрсатилган вақтдан кейин ўзгартирилган бўлса, ресурс майдони сўровга жавоб каби берилади.

HEAD методи ҳам GET методига ўхшаш, фақат у ресурс танасини қайтаради ва шартли аналогга эга эмас. HEAD ресурси ресурсдан маълумот олиш учун қўлланилади. Бу гиперматнли жўнатмаларни тестлаш масалаларни ечиш учун керак бўлади.

POST методи серверга ресурслар аннотацияси, янгиликлари ва почтали хабарлар, керакли ва катта ҳажмдаги маълумотни жўнатиш учун маълумотлар базасида қўшимчалар қўшиш учун яратилган. POST методида GET ва HEAD методларидан фарқли равишда формалар майдонидан ёки бошқа манбадан олинган маълумотни, яъни ресурс танасини узатади.

Шу вақтгача биз фақат асосий тушунчалар ва назариялар билан танишдик. Энди уларни амалиётда қўллашни ўрганамиз. Қуйида биз серверга қандай қилиб сўров бериш ва уни жавобини қайта ишлашни ўрганамиз.

6.5. HTML формасини серверга маълумотларни жўнатиш учун қўлланилиши.

Маълумотларни серверга жўнатиш учун HTML тилида махсус конструкция – формалар мавжуд. Формалар фойдаланувчидан маълумот олиш учун яратилган. Масалан, фойдаланувчининг логини ва пароли уни сайт саҳифасидан фойдаланишига рухсат бериш учун керак. Ёки фойдаланувчининг у билан алоқа қилиш учун шахсий маълумотлари керак. Формалар айнан шундай маълумотларни олиш учун ишлатилади. Формаларга матн киритиш ёки рўйхат ичидан кераклигини танлаб киритиш мумкин. Формаларга ёзилган маълумотлар сервернинг махсус дастурига қайта ишлаш учун жўнатилади (масалан, PHP скриптига). Фойдаланувчи киритган маълумотларга қараб бу дастур уларни ҳар-хил web-саҳифа кўринишида форматлаши, маълумотлар базасига сўровни жўнатиши, ҳар-хил изохлар киритиши ва ҳоказоларни бажаради.

HTML-формасининг синтаксиси билан танишамиз. Кўпчилик бу билан таниш бўлиши мумкин, лекин биз унинг асосий жойларини қайтариб ўтамиз.

Формага маълумотларни чиқазиш учун турли бошқарувчи элементлар ишлатилади. Бу элементлардан бирида фойдаланувчи маълумотларни клавиатура ёрдамида киритади, бошқасида эса у сичқонча ёрдамида керакли вариантни танлайди. Формада фойдаланувчи томонидан ўзгартириб бўлмайдиган яширин майдонлар ҳам қатнашади. Битта саҳифа таркибида бир нечта форма бўлиши мумкин, шунинг учун бир формани бошқасидан ажратиб олиш имконини берувчи восита зарур. Шунга кўра, фойдаланувчи форма билан ишлаётганда, формадаги маълумотлар қайси саҳифага ўтиши кераклиги кўрсатилиши керак(яъни, жўнатиш тугмаси босилганда форма маълумотларни қаерга жўнатсин, ўша манзил кўрсатилиши керак). Барча масалалар қуйидагича HTML теглари билан берилган форма ичига олиб ечилади:

`<form action = амал method = «метод»>` – форма элементлари-`</form>`

Берилган фрагментдан кўриниб турибдики, форма тегида иккита зарур элемент: амал ва метод кўрсатилади. Амал, форма маълумотлари жўнатиладиган ва қайта ишланадиган саҳифа номини, метод эса маълумотларни ушбу саҳифага қандай узатиш методини аниқлайди. Иккита метод мавжуд:

- **get** методи форманинг барча маълумотларини URL тугашида узатади. Бу метод махсус тиллар билан боғлиқ турли чегаралашларни ва маълумот узунлигини камдан-кам ҳолларда қабул қилади;
- **post** методи формадаги барча маълумотларни сўров танасида узатади. Бу метод **get** га нисбатан кўп қўлланилади.

Клавиатура билан киритишга йўналтирилган форма элементлари

Формани яратиш клавиатура билан киритишга йўналтирилган элементлар билан бошланади. Бундай элементлар иккита – матнли майдон (input) ва матнли соҳа (text area).

Матнли майдон форма элементи

Матнли майдонда одатда қисқа матнли маълумотлар киритилади, масалан, исми, манзили ёки электрон почта манзили.

Синтаксиси:

```
<input type=«text» name=«ўзгарувчи_номи» size=«N» maxlength=«N» value=«»>
```

Матнли майдонни аниқлаш бешта атрибутни ўз ичига олади:

- type – элемент типи (матнли майдон учун – text);
- name – киритилган маълумот сақланадиган ўзгарувчи_номи;
- size – матнли майдоннинг браузердаги умумий узунлиги;
- maxlength – матнли майдонга киритиладиган белгиларнинг энг катта сони;

• value – одатда матнли майдонда чиқадиган қиймат.

Матнли майдонларни алоҳида тури бу паролларни киритиш майдони. У худди оддий матнли майдон каби ишлайди, битта фарқи киритилаётган белгилар юлдузчага алмашади. Паролларни киритишга форма майдони яратиш учун type=«text» ўрнига type=«password» ни киритиш керак, яъни:

```
<input type=«password» name=«ўзгарувчи_номи» size=«N» maxlength=«N» value=«»>
```

Матнли соҳа форма элементи

Матнли соҳа (textarea) бир нечта катта ҳажмдаги оддий исмлар, электрон почта манзиллари, чегараланмаган матнларни киритиш учун ишлатилади.

Синтаксиси:

```
<textarea name=«ўзгарувчи_номи» rows=«N» cols=«N» value=«»></textarea>
```

Матнли соҳани аниқлаш учта атрибутни ўз ичига олади:

- name – киритилган маълумот сақланаётган ўзгарувчи_номи;
- rows – матнли соҳани сатрлар сони;
- cols – матнли соҳани устунлар сони.

Форманинг бошқа элементларида фойдаланувчи олдиндан берилган вариантлардан бирини сичқонча ёрдамида танлайди. Булар байроқчалар(checkbox), ёкиб-ўчиргичлар(radio button) ва очиладиган рўйхатлар(select).

Checkbox форма элементи

Байроқчалар (checkbox) фойдаланувчи тайёр тўпламдан бир ёки бир нечта вариантни танлаш ҳолатларида ишлатилади.

Синтаксиси:

```
<input type=«checkbox» name=«ўзгарувчи_номи» value=«қиймат»>
```

Байроқчани аниқлаш ўз ичига учта атрибутни олади:

- type – элемент типи (байроқча учун – checkbox);
- name – киритилган маълумот сақланадиган ўзгарувчи_номи (ҳозирги ҳолатда – элемент ҳолати);

• value – одатда кўра ўзгарувчига тенг бўлган қиймат. Агар байроқча ўрнатилган бўлса, айнан ушбу қиймат кўрсатилган номдаги ўзгарувчига тенг бўлади. Агар байроқча ўрнатилмаган бўлса, value атрибути қиймати

ишлатилмайди.

radio button форма элементи

Ёқиб-ўчиргичлар(*radio button*) байроқчани бир тури ҳисобланади. У амалда қуйидагича ишлатилади: бир вақтнинг ўзида фақатгина битта ёқиб-ўчиргич ўрнатилади, қолганлари эса истисно қилинади.

Синтаксиси:

```
<input type=«radio» name=«ўзгарувчи_номи» value=«қиймат»>
```

Ёқиб-ўчиргичларни аниқлашда учта атрибут иштирок этади:

- *type* – элемент типи (ёқиб-ўчиргичлар учун – *radio*);
- *name* – киритилган маълумотлар сақланадиган *ўзгарувчи_номи* (айни пайтда – элемент ҳолати);

• *value* – одатга кўра ўзгарувчи тенгладиган *қиймат*. Агар ёқиб-ўчиргич ўрнатилган бўлса, айнан ушбу *қиймат* кўрсатилган номдаги ўзгарувчига тенгладиган. Агар ёқиб-ўчиргич ўрнатилмаган бўлса, *value* атрибутининг қиймати ишлатилмайди.

select форма элементи

Агар мумкин бўлган вариантлар рўйхати кўп бўлса ва фойдаланувчи улардан битта вариантни танлаши керак бўлган ҳолатларда очиладиган рўйхатлар(*select*)ни қўллаш алоҳида қулай ҳисобланади.

Синтаксиси:

```
<select name=«ўзгарувчи_номи»>
```

```
<option value=«қиймат1»>
```

```
<option value=«қиймат2»>
```

```
<option value=«қиймат3»>
```

```
... ...
```

```
<option value=«қийматN»>
```

```
</select>
```

Очиладиган рўйхатларни аниқлашда учта атрибут иштирок этади:

- *name* – киритилган маълумот сақланадиган *ўзгарувчи_номи* (айни пайтда – рўйхатдан танланган сатр);
- *value* – одатга кўра рўйхатда чиқувчи *қиймат*.

Яширин майдон форма элементи

Яширин майдон браузерда кўринмайди ва одатда саҳифалар ўртасида маълумотларни узатиш учун ишлатилади. Яширин майдонларни аниқлаш синтаксиси матнли майдонларни аниқлаш синтаксисига жуда ўхшаш бўлиб, фақатгина тип атрибути билан фарқ қилади.

Синтаксиси:

```
<input type=«hidden» name=«ўзгарувчи_номи» value=«қиймат»>
```

Яширилган майдонни аниқлашда учта атрибут иштирок этади:

- *type* – элемент типи (яширилган майдон учун – *hidden*);
- *name* – яширилган маълумот сақланадиган *ўзгарувчи_номи*;
- *value* – одатга кўра яширилган майдонда сақланаётган *қиймат*.

Яширин майдон браузерда кўрсатилмаса ҳам, фойдаланувчи View Source буйруғини ишлатиб формада қандай яширин *қиймат* сақланаётганлигини кўриши мумкин.

Маълумотни жўнатиш тугмаси

Маълумотларни жўнатиш тугмаси `<form>` тегидаги `action` атрибутида берилган амални бажаради. Ушбу атрибутда кўрсатилган саҳифага формадаги маълумотларни жўнатади.

Синтаксиси:

```
<input type=«submit» value=«Тугма_учун_матн»>
```

Тугмани аниқлашда иккита атрибут иштирок этади:

- `type` – элемент тип (маълумотларни жўнатиш тугмаси учун– `submit`);
- `value` – *Тугма_учун_матн*.

Формадаги маълумотларни тозалаш (сброс) тугмаси

Тозалаш тугмаси форма элементидаги барча киритилган маълумотларни ўчиради. Web формаларда бу тугмани тез-тез учратиш мумкин.

Синтаксиси:

```
<input type=«reset» value=«Тугма_учун_матн»>
```

Тугмани аниқлашда иккита атрибут иштирок этади:

- `type` – элемент тип (тозалаш тугмаси учун– `reset`);
- `value` – *Тугма_учун_матн*.

Тозалаш тугмаси кўриниши худди жўнатиш тугмаси каби, шунинг учун бу тугмаларга мос номлар берилади.

Форма элементлари билан танишиб чиқдик. Энди шу форма элементлари ёрдамида мисол кўрамыз.

6.2 - мисол. Webtest дастурининг тестни бошлаш формасини яратамиз.

6.1 - листинг. index.html

```
<h2>Ma'lumotlarni kiriting</h2>
<form action="start.php" method=POST> <!--formani yaratamiz-->
<!--berilgan formani start.php fayli qayta ishlaydi,
forma ma'lumotlarini jo'natishda POST usuli qo'llaniladi-->
Ismingiz<br><input type=text name="first_name"
value="Ismingizni kiriting"><br>
Familiyangiz<br><input type=text name="last_name"><br>
E-mail <br><input type=text name="email"><br>
<P>Fanni tanlang:<br>
<select name="fan">
<option value="tarix"> Tarix </option>
<option value="falsafa"> Falsafa </option>
<option value="matematika"> Matematika </option>
<option value="informatika"> Informatika </option>
</select>
<P><input name="confirm" type=checkbox checked>To'ldirganligingizni tasdiqlan
g <br>
<input type=submit value="Jo'natish">
<input type=reset value="Tozalash">
</form>
```

Бу файлнинг браузердаги кўриниши:

6.5 – расм. HTML формасига мисол.

HTML формаси шу каби кўринишларда яратилади. Кўриниб турганидек, формада маълумотларни жўнатиш методи кўрсатилган бўлиши керак. Энди GET ва POST методларини фарқини кўрамыз.

GET методи.

GET методи ёрдамида форма маълумотини жўнатиш учун форма таркибидаги URL га сўроқ белгисидан сўнг & амперсанд ёрдамида бирлаштирилган *номи=қиймати* жуфтликлари(сўров қисми) қўшилади:

action?name1=value1&name2=value2&name3=value3

бу ерда *action* – дастурнинг формани қайта ишлаши керак бўлган URL манзили (бу form теги билан action атрибутига берилган дастур ёки агар атрибут тушириб қолдирилган бўлса, жорий дастурнинг ўзи). *name1*, *name2*, *name3* номлар форма элементлари номи, *value1*, *value2*, *value3* лар эса ушбу элементларнинг қийматидир. Ҳамма махсус белгилар = ва & ҳам параметрлар номига қараб тушириб қолдирилади. Шунинг учун бу белгилар ва идентификатордаги кириллча белгилар номлашда қўлланилмайди.

Агар майдонга қандайдир хизматчи белги киритилган бўлса, у ҳолда у ўн олтиталик санок системасида берилади. Масалан, \$ белгиси %24 га алмашади. Ҳарфлар ҳам худди шу каби берилади.

Матн ва парол киритиш майдони(бу type=text ва type=password атрибутли INPUT элементи)да *қиймат* фойдаланувчи томонидан киритилади. Агар фойдаланувчи ушбу майдонга ҳеч нарса киритмаса унда бу қаторда *name=* ҳосил бўлади.

checkbox ва radio button форма элементларида *value* қиймати, белгиланган *value* атрибутига тенг бўлади. Белгиланмаган форма элементлари сўров тузилаётган вақтда инкор этилади. Агар зарур бўлса, checkbox форма элементи битта атрибут name(ва ҳар хил value) га эга бўлиши мумкин. radio button элементи бўлса, ҳамма таклиф қилинган вариантлардан биттасига тегишли ва шунинг учун бир хил name атрибутига ва ҳар-хил value ларга эга бўлиши керак.

Умуман олганда, GET методида маълумотларни жўнатиш учун HTML формасини яратиш шарт эмас. Адрес қатордаги URLда керакли ўзгарувчилар ва уларни қийматларини кўшиб қўйиш етарли.

6.3 - мисол URLда маълумотларни узатилиши.

<http://phpbook.info/test.php?id=10&user=pit>

Шу билан бирга маълумотларни GET методида жўнатишни бир камчилиги бор - параметрлар қийматини бирор ким ўзгартириб қўйиши мумкин. Шунинг учун дастур ёки сервер иши хавфсизлигига дахлдор парол билан ҳимояланган саҳифаларда маълумотни жўнатиш учун бу методдан фойдаланишни маслаҳат берилмайди. Бундан ташқари фойдаланувчи томонидан ўзгартириш таъқиқланган маълумотларни жўнатиш учун GET методидан фойдаланиш ярамайди.

Шунча камчиликларга қарамасдан, скриптларни созлашда (бу ҳолда жўнатилаётган ўзгарувчиларни номини ва қийматини кўриш мумкин) ва хавфсизликка дахлсиз бўлган параметрларни жўнатишда GET методидан фойдаланиш етарли даражада қулай.

POST методи.

POST методида формалар худди GET методидагидек кодланади, лекин сўров қисми URLга кўшилмайди, балки POSTнинг амаллари, яъни берилганлар блоки қисмидек юборилади. Агар action атрибути мавжуд бўлса, унда URL ушбу берилганлар блокини қаерга жўнатишни аниқлайди. Бу метод юқорида айтиб ўтилгандек катта ҳажмдаги маълумотлар блокини жўнатиш учун қўлланилади.

Жўнатилаётган файлнинг узунлиги `content_length` доирасидаги ўзгарувчиларга жўнатилади, маълумотлар типи эса `content_type` ўзгарувчисига жўнатилади.

Маълумотларни post методи билан жўнатишни фақат HTML-формаси ёрдамида амалга ошириш мумкин, чунки маълумотлар GETдагидек сўров сарлавҳасига эмас, балки сўров қисмига жўнатилади. Шунга кўра фақатгина формага киритилган қийматни ўзгартириш орқали параметр қийматини ўзгартириш мумкин. Фойдаланувчи POSTдан фойдаланаётганда серверга берилаётган маълумотларни кўрмайди.

POST сўровларининг асосий хусусияти унинг хавфсизлигини катталиги ва функционалигидадир. Шунинг учун POST методидан катта ҳажмдаги ва керакли маълумотларни жўнатишда фойдаланилади. Лекин бу механизмга ҳам бутунлай ишониб бўлмайди, чунки унда ҳам POST сўровларини ўзгартириш мумкин. Масалан, html-файлини ўз машинамизда яратиб уни

керакли маълумотлар билан тўлдирилса кифоя. Бундан ташқари POST методидан ҳамма мижозлар фойдалана олмайдилар, чунки унинг ишлатиш вариантлари чекланган.

Маълумотларни серверга исталган йўл билан жўнатишда фақат маълумотлар эмас, балки фойдаланувчи томонидан киритилган бир қатор доиравий ўзгарувчилар номли ўзгарувчилар қатори, мижозни характерлайдиган параметрлар, унинг иши ҳақидаги маълумотлар, файлларга йўл ва ҳ.к.лар ҳам берилади. Қуйида доиравий ўзгарувчилардан айримларини келтирамыз:

- **REMOTE_ADDR** - хост(компьютер)нинг сўров жўнатилаётган IP-манзили;
- **REMOTE_HOST** - сўров жўнатилган хостнинг номи;
- **HTTP_REFERER** – жорий скриптга мурожаат(ссылка) этиладиган қатор манзили;
- **REQUEST METHOD** – сўровни жўнатишда қўлланилган метод;
- **QUERY_STRING** – URLда сўроқ белгисидан кейин жойлашган сўров қисми;
- **SCRIPT_NAME** – бажарилиши керак бўлган виртуал дастурга йўл;
- **HTTP_USER_AGENT** – мижоз фойдаланаётган браузер ҳақида маълумот.

6.6. Сўровларни PHP ёрдамида қайта ишлаш.

Мижоз сўровининг серверда махсус дастурлар ёрдамида қайта ишланишини кўриб чиққансиз. Ушбу махсус дастурни PHP тилида ёзиш ва унда киритилган маълумотлар билан исталган амални бажариш мумкин. Бу дастурни ёзишда PHP нинг инструментлар ва қоидалари билан танишиб чиқиш керак.

PHP скрипти ичида мижозга HTTP протоколи асосида берилган бир канча маълумотларга рухсат олиш йўллари мавжуд. PHP 4.1.0.версиясигача бундай маълумотларга рухсат жўнатилган ўзгарувчилар номи(маълумотлар «ўзгарувчи номи, « =» белгиси, ўзгарувчи қиймати» жуфтлиги кўринишида берилган) асосида амалга оширилган. Шу тариқа агар first_name=Lola берилган бўлса, скрипт ичида Lola қийматли \$first_name ўзгарувчи ҳосил бўлади. Агар маълумотларни қандай йўл билан берилганлигини ажратиш талаб қилинган бўлса, унда берилган ўзгарувчилар калити бўлган \$HTTP_POST_VARS ва \$HTTP_GET_VARS ассоциатив массивлар кўлланилган. Худди шундай агар first_name=Lola GET методи билан жўнатилган бўлса, унда

\$HTTP_GET_VARS[«first_name»]=«Lola» бўлади.

Дастурда ўзгарувчилар номини тўғридан-тўғри юбориш хатарли эмас. Шунинг учун PHP 4.1.0 версиясидан бошлаб HTTP сўровлари ёрдамида берилган ўзгарувчиларга махсус \$_REQUEST массиви ёрдамида мурожаат этиш мумкин. Бу массивлар POST ва GET методи ҳамда HTTP cookies ёрдамида берилган маълумотларни ўз ичига олади. Бу суперглобал ассоциатив массив, шунинг учун мос ўзгарувчи(форма элементи) номини калит сифатида ишлатиб, дастурнинг исталган қисмидан унинг номини олиш мумкин.

6.2 – мисолдаги 6.1 - листингда тест топшириладиган фанлар руйхатини формаси яратилди. Энди шу форма ёрдамида жўнатилган маълумотларни қайта ишловчи test.php файлининг скриптини ёзамиз:

```
<?php
$str = "Salom,
      ".$_REQUEST["first_name"]. "
      ".$_REQUEST["last_name"]."! <br>";
$str .= "Siz test topshirish uchun
      ".$_REQUEST["fan"]." fanini tanladingiz";
echo $str;
?>
```

6.2 – листинг. Бу index.html формасини қайта ишловчи start.php файли

У ҳолда, агар биз формага “Vohid” исмини ва “Fozilov” фамилиясини киритганимизда ва фанлар ичидан Informatika фанини танлаганимизда браузер экранда куйидаги хабарни оламиз.

Натижа:

Salom, Vohid Fozilov!

Siz test topshirish uchun Informatika fanini tanladingiz

\$_REQUEST массивини киритиш учун \$HTTP_POST_VARS ва \$HTTP_GET_VARS бир хиллик массивлари мос равишда \$_POST ва \$_GET

га қайта номланади. \$_POST ва \$_GET массивлари бошқалардан фарқли равишда методлар ва функциялар ичида тўғридан – тўғри рухсатга эга суперглобал массивларга айланади.

Бу массивларни қўлланишига мисоллар келтирамиз. first_name, last_name, fan номли элементларни ўз ичига олган формани қайта ишлаш керак. POST ва бошқа методлар билан берилган маълумотларни қайта ишлашни қуйидагича амалга ошириш мумкин:

```
<?php
$str = "Salom,
    ".$ POST ["first name"]."
    ".$_POST ["last_name"] ."! <br>";
$str .= " Siz test topshirish uchun ".$ _POST["fan"] ." fanini tanladingiz";
echo $str;
?>
```

Агар биз юқоридаги мисолдагидек “Vohid “ исмини ва “Fozilov” фамилиясини киритсак ва фанлар ичидан Informatika фанини танласак, у ҳолда браузер экранида қуйидаги ҳосил бўлади.

Натижа:

Salom, Vohid Fozilov!

Siz test topshirish uchun Informatika fanini tanladingiz

Қайта ишланган скриптларни хотирага сақлаш учун PHP 4.1.0 версиясидан олдингиларида registr_globals ўзгарувчиларга бевосита номлари бўйича рухсатни таъқиқлайдиган ёки таъқиқламайдиган директиваси киритилган эди. Агар PHP нинг конфигурация файли(httpd.conf)да registr_globals=on параметри мавжуд бўлса, у ҳолда сервердаги GET ва POST методи билан берилган ўзгарувчиларга уларни номлари бўйича мурожаат этиш мумкин (яъни, \$first_name каби ёзиш мумкин). Агар registr_globals=off бўлса, у ҳолда \$_REQUEST[“first_name”] ёки \$_POST[“first_name”], \$_GET[“first_name”], \$_HTTP_POST_VARS[“first_name”], \$_HTTP_GET_VARS[“first_name”] деб ёзиш керак. Хавфсизлик нуқтаи назаридан бу директивани ўчириб қўйган маъқул(registr_globals=off). registr_globals директиваси ёқилган вақтда ҳам мижоз томонидан берилган маълумотлар юқорида қайд этилган массивларни ўз ичига олади.

Айрим ҳолларда сўров жўнатган компьютернинг IP - манзилини ёки қандайдир доиравий ўзгарувчи қийматини билиш керак бўлади. Бундай маълумотни **getenv()** функцияси ёрдамида бажариш мумкин. У параметр сифатида берилган доиравий ўзгарувчи қийматини қайтаради.

6.4 - мисол. getenv() функциясининг ишлатилиши

```
<?
getenv("REQUEST_METHOD");
// qo'llanilgan usulni qaytarish
echo getenv ("REMOTE_ADDR");
// so'rov jo'natgan foydalanuvchining
// IP- adresini chiqaradi
?>
Natija:127.0.0.1
```

Масалан, test.php файлида ушбу скриптни ёзсак ва html форма маълумотларини жўнатсак. У ҳолда натижада маълумотлар жўнатилаётган компьютер IP манзили чиқазилади.

Айтиб ўтганимиздек GET методи қўлланилганда, маълумотлар сўров қаторига “*ўзгарувчи_номи=ресурснинг URL манзил қиймати*” жуфтлиги қўшилиб жўнатилади. URL да сўроқ белгисидан кейин ёзилганларнинг барчасини

```
getenv(«QUERY_STRING»);
```

буйруғи ёрдамида олиш мумкин.

Шу сабабдан GET методидида маълумотларни бошқа хил кўринишда бериш мумкин. Масалан, қўшиш белгиси орқали бир нечта параметрларни киритиб, скрипт ичида бўлса сўров қаторини ажратиш ёки фақат битта параметр қийматини бериш мумкин. Бу ҳолда \$_GET массивида ушбу қийматга тенг бўлган бўш қатор ҳосил бўлади, сўров қаторида учраган “+” белгиси эса “_”га алмаштирилади.

POST методидида маълумотлар фақат формалар ёрдамида берилади ва фойдаланувчи серверга қандай маълумотлар жўнатилаётганини кўра олмайди. Уни кўриш учун хаккер бу формани ўзининг формасига алмаштириши керак. Бу ҳолда сервер қайта ишланган нотўғри формани жўнатилиши керак бўлган жойга жўнатмайди. Бундай ҳолга дуч келмаслик учун маълумотлар жўнатиш саҳифа билан манзил саҳифасини текшириб кўриш керак. Буни ҳам getenv() функцияси ёрдамида бажариш мумкин:

```
getenv(«HTTP_REFERER»);
```

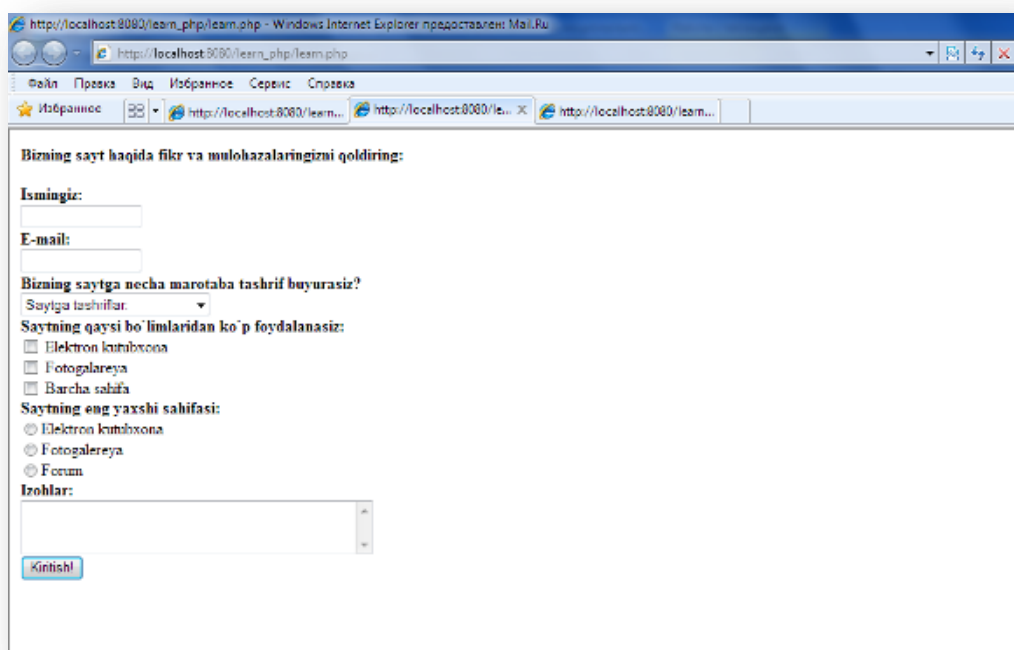
6.7.Формалар билан ишлашга мисоллар

Қуйидаги амалий мисоллар РНР да формаларни қайта ишлашни турли аспектларини ўрганишга ёрдам беради. Бу мисолларда турли йўллар билан сайтнинг интерактив имкониятларини ошириш кўрсатилади.

6.1 - Масала. Сайтнинг **Бизга ёзинг...саҳифасида** фойдаланувчи билан мулоқот ўрнатувчи форма яратамиз. Бу форма орқали фойдаланувчи ўзининг сайт ҳақидаги фикр - мулоҳазаларини жўнатиши мумкин:

6.5 – листинг.

```
<form action = "process.php" method = "post">
<b>Bizning sayt haqida fikr va mulohazalaringizni qoldiring:</b>
<p><b>Ismingiz:</b><br>
<input type="text" name="name" size="15" max-length="25" value=""><br>
<b>E-mail:</b><br>
<input type="text" name="email" size="15" max-length="45" value=""><br>
<b>Bizning saytga necha marotaba tashrif buyurasiz?</b><br>
<select name="tashrif">
<option value="">Saytga tashriflar:
<option value="0">Bu mening birinchi kirishim
<option value="1">Bir oyda 1 marta
<option value="2"> Hafta davomida
<option value="3"> Har kuni
</select><br>
<b>Saytning qaysi bo`limlaridan ko`p foydalanasiz:</b><br>
<input type="checkbox" name="kutubxona" value="kutubxona "> Elektron kutubxon
a<br>
<input type="checkbox" name="foto" value="foto"> Fotogalareya<br>
<input type="checkbox" name="barchasi" value="barchasi"> Barcha sahifa<br>
<b>Saytning eng yaxshi sahifasi:</b><br>
<input type="radio" name="asset" value="kutubxona">Elektron kutubxona<br>
<input type="radio" name="asset" value="foto">Fotogalereya<br>
<input type="radio" name="asset" value="forum">Forum<br>
<b>Izohlar:</b><br>
<textarea name="izoh" rows="3" cols="40"></textarea><br>
<input type="submit" value="Kiritish!"></form>
```



6.6 – расм. Форманинг браузердаги кўриниши.

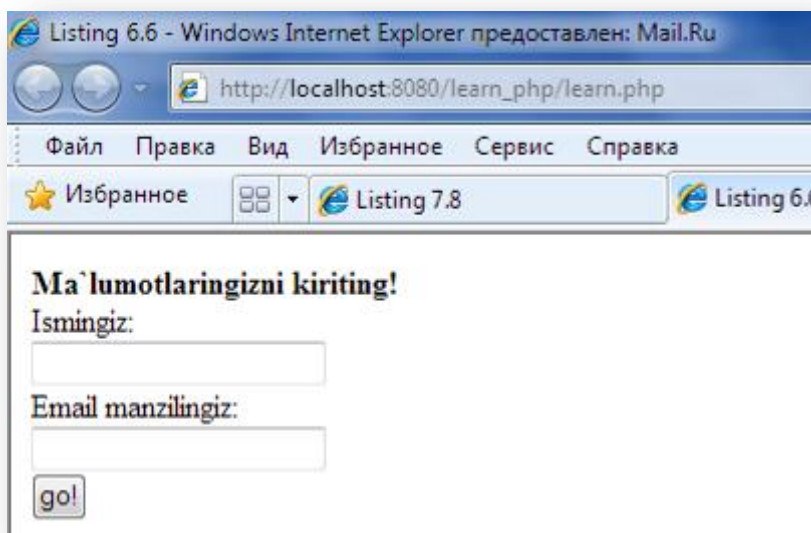
6.2 – Масала. Бир саҳифадан иккинчи саҳифага маълумотларни узатиш.

Бу мисолдаги формада фойдаланувчининг номи ва электрон почта манзили кўрсатилади. Фойдаланувчи маълумотларни жўнатиш (Go!) тугмасини босганда, форма **6.7-листинг**да берилган саҳифага мурожаат этади, яъни фойдаланувчи маълумотлари биринчи саҳифадан иккинчи саҳифага узатилади. Ўз навбатида, берилган сўровга кўра \$name ва \$mail ўзгарувчи қийматларини чиқазади.

6.6 – листинг.

```
<html>
<head>
<title>Listing 6.6</title>
</head>
<body bgcolor="#ffffff" text="#000000"link="#cbda74" vlink="#808040" alink="#808040">
<form action="listing6_7.php" method="post">
<b>Ma'lumotlaringizni kiriting!</b><br>
Ismingiz:<br>
<input type="text" name="name" size="20" maxlength="20" value=""><br>
Email manzilingiz:<br>
<input type="text" name="email" size="20" maxlength="40" value=""><br>
<input type="submit" value="go!">
</form>
</body>
</html>
```

Натижа



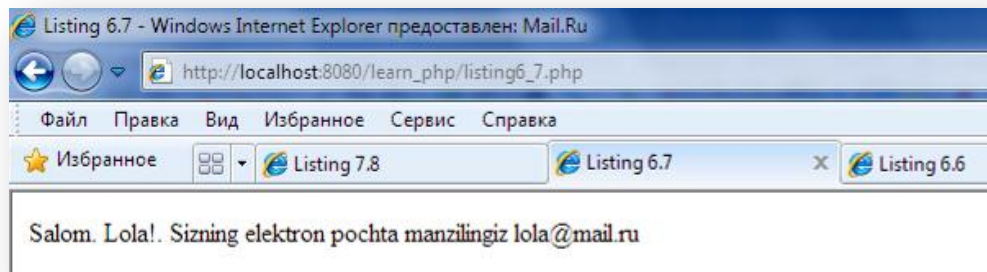
6.7-расм. Маълумотларни бир саҳифадан бошқасига жўнатиш

6.7-листинг. 6.7-расмдаги формада киритилган маълумотларни чиқазиш. 6.7-расмдаги формада киритилган маълумотлар бу саҳифада акс этиши учун php.ini файлидаги register_globals=Off ни register_globals=On га алмаштириш керак. Ёки маълумотларни массив сифатида жўнатиш керак.

```
<html>
<head>
<title>Listing 6.7</title>
</head>
<body bgcolor="#ffffff" text="#000000"link="#cbda74" vlink="#808040" alink="#808040">
<?>
```

```
// Ismi va elektron pochta manzilinini chiqazish
print "Salom. $name!. Sizning elektron pochta manzilingiz $email";
?>
</body>
</html>
```

Натижа



6.8-расм. Маълумотларни чиқазииш

Бу мисолда ҳаммаси оддий бажарилади: фойдаланувчи форма майдонларини тўлдиради ва маълумотларни жўнатиш тугмасини босади. Бошқарув 6.7-листинг саҳифасига узатилади. Бу саҳифада маълумотлар қайта ишланади ва охирида маълумотларни чиқазади.

Формадаги маълумотларни қайта ишлашни бошқа усуллари ҳам мавжуд. Яъни маълумотларни киритиш, қайта ишлаш ва чиқазииш ҳаммаси битта саҳифада бажарилади. Бу усулнинг камчилиги шундаки, ҳаммаси битта саҳифада бўлганлиги учун саҳифа катталашади ва уни таҳрирлаш ва бошқариш бир мунча қийинчилик туғдиради.

6.3 – Масала. Формани қайта ишлашнинг альтернатив йўли (битта саҳифада).

Форма маълумотларини битта саҳифада қайта ишлаш деярли оддий амалга оширилади: Форма ўзгарувчисига қийматни тенглаштириш бажарилганлиги текширилади. Агар қиймат тенглаштирилган бўлса, саҳифа уларни қайта ишлайди(бизнинг мисолда маълумотларни чиқазади), акс ҳолда эса формани ўзини чиқазади. Формани битта саҳифада амалга ошириш учун дастур коди 6.8-листингда берилган. if шартли ўтиш оператори \$seenform номли яширин майдон ўзгарувчиси ҳолатини текширади. Агар \$seenform қиймати берилмаган бўлса, форма браузерда чиқазилади, агар берилган бўлса, у ҳолда форма фойдаланувчи томонидан тўлдирилган ва киритилган маълумотлар саҳифада қайта ишланади(бизнинг мисолда – браузерга чиқазилади).

6.8 - листинг. Формага маълумотларни битта саҳифада киритиши

```
<html>
<head>
<title>Listing 6.8</title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74" vlink="#808040" alink="#f
808040">
<?
// $form ichidagi qo`shtirnoqda joylashganlarning
//hammasi ekranga chiqishi kerak
// aks holda xatolik yuzaga keladi.
```

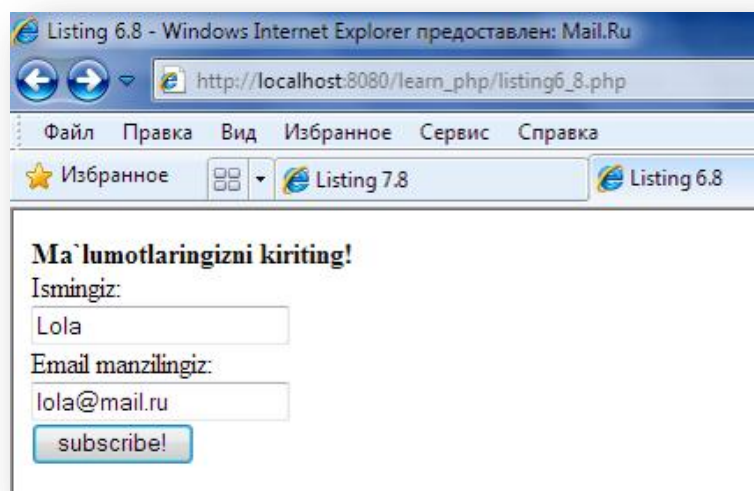
```

$form = "
<form action=\"listing6.8.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<b>Ma`lumotlaringizni kiriting!</b><br>
Ismingiz:<br>
<input type=\"text\" name=\"name\" size=\"20\" maxlength=\"20\" value=\"\"><br>
Email manzilingiz:<br>
<input type=\"text\" name=\"email\" size=\"20\" maxlength=\"40\" value=\"\"><br>
<input type=\"submit\" value=\"subscribe!\">
</form>";
// agar forma avval chiqazilmagan bo`lsa, uni chiqazish,
// tekshirish uchun $seenform yashirin o`zgaruvchisining qiymati ishlatiladi
if (!isset($seenform)):
print "$form";
else :
print "Salom. $name! Sizning elektron pochta manzilingiz $email";
endif;
?>
</body>
</html>

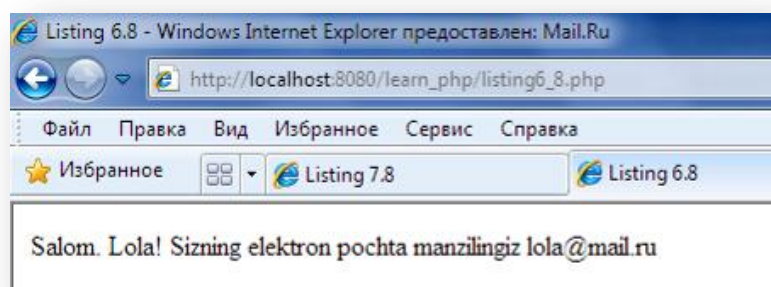
```

Шу скрипти ишга тушириш билан натижа сифатида куйидаги форма чиқазилади

Натижа



6.9 –расм. Маълумотларни формага киритишни битта саҳифада амалга ошириш



6.10 –рasm. Маълумотларни формага киритишни битта саҳифада амалга ошириш

Бу вариант муайян ноқулайликларни келтиради, саҳифа қайта юкланиши сабабли фойдаланувчи форма майдонлари тўғри тўлдирганми йўқми ҳеч нарса билолмайди.

6.4 - Масала.

Машғулотга қатнашувчи ўқувчиларни руйхатдан ўтказиш формаси тузилиб, руйхатдан ўтгандан сўнг уларга хабар жўнатилади.

6.9 – листинг. form_yakuniy.html

```
<h2>Qatnashuvchilar ro'yxati</h2>
<form action="listing6_10.php" method=POST> <!--formani yaratamiz-->
<!--berilgan formani test.php fayli qayta ishlaydi,
forma ma'lumotlarini jo'natishda POST usuli qo'llaniladi-->
Ismingiz<br><input type=text name="first_name"
value="Ismingizni kiriting"><br>
Familiyangiz<br><input type=text name="last_name"><br>
E-mail <br><input type=text name="email"><br>
<P> Siz qatnashmoqchi bo'lgan fanni tanlang:<br>
<input type=checkbox name='fan[]' value='Matematika'> Matematika <br>
<input type=checkbox name='fan[]' value='Informatika'> Informatika <br>
<input type=checkbox name='fan[]' value='Tarix'> Tarix <br>
<input type=checkbox name='fan[]' value='Falsafa'>Falsafa<br>

<P> O'zingiz haqingizda qisqacha ... <BR>
<textarea name="comment" cols=32 rows=5></textarea>
<P><input name="confirm" type=checkbox checked>To'ldirganligingizni tasdiqlan
g <br>
<input type=submit value="Jo'natish">
<input type=reset value="Tozalash">
</form>
```

Бу ерда фақатгина checkbox элементи қийматини белгилаб қўйиш керак. checkbox элементи танланганда унинг қийматлари мос равишда биринчи элементи fan массивининг биринчи элементига, иккинчи элементи массивнинг иккинчи элементига ва ҳ.к. ёзилади. checkbox элементларига ҳархил номлар ҳам бериш мумкин, лекин бу, агар фанлар кўп бўлса, маълумотларни қайта ишлашни қийинлаштиради.

Буларни ҳаммасини муҳокама этувчи ва қайта ишловчи скрипт test.php деб аталади (бу форманинг action атрибутида кўрсатилган). Маълумотларни жўнатиш учун POST методи кўрсатилади. Рўйхатдан ўтган шахсдан олинган маълумотларга қараб скрипт мос хабарни ишлаб чиқади. Агар рўйхатга олинган шахс қандайдир фанларни танлаган бўлса, унда унга ушбу фанларни қачон бўлиши ва уларни ўқийдиган маърузачилар ҳақида маълумот беради. Агар ўқувчи ҳеч қандай фанни танламаган бўлса, у ҳолда машғулоти фанларининг кейинги мажлиси ҳақида хабар берилади.

6.10 – листинг. form_yakuniy.html формасини қайта ишловчи 6.10 - listing.php файли

```
<?
// mos holda mashg'ulot o'tish vaqtlari va ma'ruzachilar haqidagi
// massivni yaratamiz
$times= array("Matematika"=>"14.30", "Informatika"=>"12.00", "Tarix"=>"15.00"
, "Falsafa"=>"14.00");
$selectors=array("Matematika"=>"Sayfulla Dottoyev", "Informatika"=>"Azim Karimo
```



```

v", "Tarix"=>"Farhod Sobirov", "Falsafa"=>"Baxtiyor Sodiqov");
define("SIGN", "Hurmat bilan ma'muriyat");
// Xat imzosini konstanta sifatida aniqlaymiz
define("MEETING_TIME", "18.00");
// Talabalar yi'gilishi vaqtini beramiz
$date = "20 may"; // ma'ruzalarni o'tish vaqtini beramiz
//Xabar matnini tuzishni boshlaymiz
$str = "Assalomu alaykum hurmatli " . $_POST["first_name"]
. " " . $_POST["last_name"]."!<br>";
$str .= "<br>Sizga shuni ma'lum qilamizki ";
$fans=$_POST["fan"]; // bu o'zgaruvchiga tanlangan
// fanlar ruyxatini saqlaymiz
if (!isset($fans)) { // agar faqat bitta fan tanlanmagan bo'lsa
    $event = " talabalarining keyingi yig'ilishi";
    $str .= "$event $date ". MEETING_TIME . " bo'lib o'tadi <br>";
} else { // hech bo'lmaganda bitta fan tanlanganda
    $event = "Siz tanlagan ma'ruzalar $date sanasida bo'lib o'tadi. <ul>";
    //count funksiyasi massivdagi elementlar sonini aniqlaydi
    for ($i=0;$i<count($fans);$i++){
        // har bir tanlangan fan uchun
        $k = $fans[$i]; // fan nomini eslab qolamiz
        $lect = $lect . "<li> $k fani bo'yicha ma'ruza $times[$k] da";
        // xabar tuzamiz
        $lect .= " (Sizning ma'ruzachingiz, $lectors[$k])";
    }
    $event = $event . $lect . "</ul>";
    $str .= "$event";
}
$str .= "<br>". SIGN; // Imzo qushamiz
echo $str; // ekranga xabar chiqaramiz
?>

```

Агар фойдаланувчи Matematika va Informatika фанларини танласа натижа:

```

Assalomu alaykum hurmatli Vohid Fozilov!

Sizga shuni ma'lum qilamizki Siz tanlagan ma'ruzalar 20 may sanasida bo'lib o'tadi.

    • Matematika fani bo'yicha ma'ruza 14.30 da (Sizning ma'ruzachingiz, Sayfulla Dottoyev)
    • Informatika fani bo'yicha ma'ruza 12.00 da (Sizning ma'ruzachingiz, Azim Karimov)

Hurmat bilan ma'muriyat

```

6.11 – расм. 6.10 -listing.php файлининг браузердаги кўриниши
Агар фойдаланувчи бирорта ҳам курсни танламаса натижа:

Assalomu alaykum hurmatli Vohid Fozilov!

Sizga shuni ma'lum qilamizki talabalarning keyingi yig'ilishi 20 may 18.00 bo'lib o'tadi

Hurmat bilan ma'muriyat

6.12– расм. 6.10 -*listing.php* файлининг браузердаги кўриниши

VII БОБ. БОШҚАРУВЧИ КОНСТРУКЦИЯЛАР

Бу бобда танлаш операторлари (*if, else* – шартли ўтиш операторлари, *switch-case* қайта уловчи операторлар), такрорлаш операторлари (*while, for, foreach*), ўтиш операторлари (*continue, break*) ва қамраб олиш операторлари (*include, require*) кабилар ҳақида сўз боради.

Бошқарувчи конструкциялар бошқача қилиб айтганда, операторлар куйидаги турларга бўлинади:

- Танлаш операторлари – шартли ўтиш операторлари (*if-else*) ва қайта уловчи (*switch-case*) операторлар
- Такрорлаш(цикл) операторлари - шартли олдиндан берилган оператор (*while*), параметрли такрорлаш оператори (*for*) ва шартли кейин келувчи операторлар (*do...while*).
- Ўтиш оператори – бошқаришнинг шартсиз узатишини таъминлайди. Уларга *continue* (такрорлашнинг ҳозирги итерациясини якунлаш), *break* (такрорлашдан чиқиш) ва *return* (функциялардан чиқиш) лар киради.
- Қамраб олиш операторлари (*include, require*).

7.1. Танлаш операторлари

if, else – шартли ўтиш операторлари

Бошқа тиллардаги каби *if* шартли ўтиш оператори PHP нинг асосий операторларидан бири. *IF* оператори шартнинг берилишига қараб, код блокини бажаради. *If* шартли ўтиш операторининг шартли (*()*) қавс ичига олиб ёзилади.

Синтаксиси:

if (шарт) бажариш_блоки;

бу ерда шарт бирор бир мазмунга эга бўлиб, скриптни қайта ишлаш жараёнида мантикий қийматга ўзгартирилади. Агар шарт бажарилса, *true*(рост) қийматни қабул қилади, яъни шарт мантикий *true*(рост) қийматга ўзгаради ва кейинги кадам (*бажариш_блоки*) бажарилади. Акс ҳолда *бажариш_блоки* бекор қилинади.

Шартларнинг мантикий қийматларга айланиш ҳолатлари

1. *False* қийматга ўзгарадиган ҳолатлар:
 - Мантикий *false*
 - Бутун нол (*0*)
 - Бўш қатор ва «*0*» қатори
 - Элементларсиз массив
 - Ўзгарувчиларсиз объект
 - Махсус *NULL*
2. Қолган барча ҳолатлар *True* қийматга ўзгаради.

7.1 –мисол. *If* шартли ўтиш оператори

Маълумот киритиш учун форма берилган. Исми киритилганми ёки йўқлиги текширилади ва мос жавоб қайтарилади. If шартли ўтиш оператори шарт ёрдамида фойдаланувчи логини киритилган ёки киритилмаганлигини текширади ва мос хабарни чиқазади.

```
<?
if (!$name)
echo "Ismingizni kiritmadingiz!";
?>
```

Бу мисолда !\$name шарти ишлатилмоқда.

Агар \$name ўзгарувчисининг қиймати бўш бўлса, !\$name шарти true(рост) қийматни қабул қилади ва шарт бажарилганлиги учун кейинги кадамга ўтилади.

If шартли ўтиш операторидан сўнг фақатгина биттадан ортиқ бўлмаган *бажариш_блоки* бажарилади. Агар *бажариш_блоки* бир нечта буйруқ сатридан иборат бўлса, ҳар бир буйруқ сатри олдида if шартли ўтиш оператори қўлланилади ёки буйруқ сатрлари {} кавс ичига олиб ёзилади, яъни бир блокка бирлаштирилади.

```
<?
if (!$name)
{
echo "<font color=red>";
echo "Ismingizni kiritmadingiz";
echo "</font>";
}
?>
```

Else operatori

Одатда фақатгина шарт бажарилган ҳолатлар қаралмайди, балки у бажарилмай қолган ҳолатларда ҳам бирор бир хулоса чиқарилади. Шарт бажарилмаган ҳолда, қандайдир ҳаракатни бажариш учун if оператори Else оператори ёрдамида кенгайтирилади.

If нинг Else оператори ёрдамида кенгайтирилган синтаксиси:

if(шарт) бажариш_блоки
else бажариш_блоки1

Агар шарт бажарилган бўлса, яъни *шарт* true(рост) қиймат қабул қилса, *бажариш_блоки* бажарилади, акс ҳолда *бажариш_блоки1* бажарилади.

```
<?php
```

if (шарт) — Агар шарт рост булса, у холда

```
{
    амал
```

Агар шарт ёлгон булса, у холда

```
}
else
{
    амал
```

```
}
?>
```

if операторини якка ҳолда ёки else оператори билан бирга ишлатиш мумкин. Лекин else операторини якка ҳолда ишлатиш мумкин эмас, фақатгина if оператори билан биргаликда ишлатилади.

7.2- мисол. Берилган сонни жуфт-тоқлигини текшириш.

```
<?
If ($a%2=0)
{
echo "$a - juft son<br>";
}
else
{
echo "$a - toq son<br>";
}
?>
```

7.3-мисол. Талаба имтиҳон топширган ёки топширмаганлиги текширилади. Агар топширган бўлса, ўтган ёки ўтмаганлиги текширилади.

```
<?
if ($natija=="")
{
echo "Talaba imtihon topshirmadi";
}
else
{
if ($natija<55) echo "Talaba imtihondan o'tmadi";
if ($natija>55) echo "Talaba imtihondan o'tdi";
}
?>
```

Демак бу мисолда биринчи if операторидаги шарт бажарилмаган ҳолда иккинчи шарт учун яна if оператори қўлланилган.

Elseif оператори

If операторининг яна бир кенгайтмасидан бири бу elseif операторидир. Elseif – бу else ва if нинг комбинациясидир.

Синтаксиси:

If(шарт) бажариш_блоки

Elseif (шарт1) бажариш_блоки1

...

Else бажариш_блокиN

Битта if блокда бир неча марта elseif операторлари қатнашиши мумкин. Агар if шарти false қиймат қабул қилса, кейинги elseif шартлар текширилади. Elseif шартларининг бирортаси true қиймат қабул қилса, мос *бажариш_блоки* бажарилади, акс ҳолда кейинги кадамга ўтилади.

7.3 - мисолни бошқачароқ шаклда elseif оператори ёрдамида ёзишимиз ҳам мумкин.

```
<?
if ($natija=="")
echo "Talaba imtihon topshirmadi";
elseif ($natija<55)
echo "Talaba imtihondan o'tmadi";
elseif ($natija>55)
echo "Talaba imtihondan o'tdi";
?>
```

Шартли ўтиш операторларида *шарт* ўрнида бир нечта шартни ҳам текшириш мумкин. Бу шартлар ВА (AND ёки &&) ва ЁКИ (OR ёки ||)

мантикий амаллари ёрдамида боғланади. Қуйидаги мисолда ВА мантикий амалини ишлатамиз.

7.4- мисол. Берилган ўзгарувчилар қийматларини солиштириш.

```
<?
$a=1;
$b=3;
$c=4;
$d=9;
if ($a!=$b and $c!=$d)
{
echo "O'zgaruvchilar qiymatlari teng emas!";
}
else
{
echo "O'zgaruvchilar qiymatlari teng!";
}
?>
Natija:O'zgaruvchilar qiymatlari teng emas!
```

Бу мисолда иккала шартда ҳам ўзгарувчи қийматлари бир-бирига тенг бўлмаганлиги, яъни шарт бажарилгани учун кейинги қадам бажарилди.

7.5 - мисол. Берилган ўзгарувчилар қийматларини солиштириш.

```
<?
$a=1;
$b=3;
$c=9;
$d=9;
if ($a!=$b and $c!=$d)
{
echo "O'zgaruvchilar qiymatlari teng emas! ";
}
else
{
echo "O'zgaruvchilar qiymatlari teng!";
}
?>
Natija:O'zgaruvchilar qiymatlari teng!
```

Бу мисолда and амали қатнашганлиги учун иккала шарт ҳам бажарилгандагина кейинги қадам бажарилади, акс ҳолда else операторидан кейинги қадам бажарилиши керак. Мисолни текшираамиз, биринчи шарт бажарилди, лекин иккинчи шарт бажарилмади, шунинг учун else операторидан кейинги қадам бажарилди.

Агар шартлар орасидаги and мантикий амал ўрнига or мантикий амални ишлатсак натижа ўзгаради.

7.6- мисол. Берилган ўзгарувчилар қийматларини солиштириш.

```
<?
$a=1;
$b=3;
$c=9;
$d=9;
if ($a!=$b or $c!=$d)
{
echo "O'zgaruvchilar qiymatlari teng emas!";
}
else
{
echo "O'zgaruvchilar qiymatlari teng!";
}
?>
```

```
?>
```

```
Natija:O'zgaruvchilar qiymatlari teng emas!
```

Бу мисолда эса ЁКИ (or ёки ||) амали қатнашганлиги учун шартларни бирортаси рост қиймат қабул қилса ҳам кейинги қадам бажарилади.

Юқоридаги мисолларда ўзгарувчи қийматларини таққослаш учун or ва and мантикий амаллари билан биргаликда == ва != амалларидан ташқари бошқа таққослаш амалларини: <, >, <=, >= ва ҳ.к ларни ҳам ишлатиш мумкин.

7.7 - мисол. Талабаларни баҳолаш мезонлари схемасини тузиб чиқамиз.

- 55 дан паст баллтўплаган талабалар – ўзлаштира олмаганлар
- 55 дан юқори 71 дан паст балл тўплаган талабалар – 3 баҳога ўзлаштирувчилар
- 71 дан юқори 86 дан паст балл тўплаган талабалар – 4 баҳога ўзлаштирувчилар
- 86 дан юқори балл тўплаган талабалар – 5 баҳога ўзлаштирувчилар

Бу мисолни PHP кодини if ва elseif операторлари ва &&(ёки AND) амали ёрдамида ёзамиз.

```
<?
```

```
if ($baho<55)
echo "O'zlashtira olmagan talabalar!";
elseif ($baho>=55 && $baho<71)
echo "3 bahoga o'zlashtirgan talabalar!";
elseif ($baho>=71 && $baho<86)
echo "4 bahoga o'zlashtirgan talabalar!";
elseif ($baho>=86)
echo "5 bahoga o'zlashtirgan talabalar!";
?>
```

Эътибор қаратинг! elseif операторини else if шаклида ҳам ёзиш мумкин, иккала ҳолат ҳам тўғри ҳисобланади.

Ичма-ич жойлашган IF-ELSE оператори

Ичма-ич жойлашган шартлар – бу if шартли ўтиш оператори ичида яна бир if шартли ўтиш операторининг ишлатилишидир. Бу ҳолда шарт текширилгандан сўнг, бажарилиши керак бўлган блок ичида бирор бир амал эмас, балки яна бир шарт текширилади.

7.8 – мисол. Ичма-ич жойлашган шартларнинг ишлатилиши

```
<?
```

```
$a=1;
$b=3;
$c=9;
$d=9;
$talaba1="bakalavr";
$talaba2="magistr";
if ($a==$b or $c==$d)
{
echo "O'zgaruvchilar qiymatlari teng!";
if($talaba1!=$talaba2)
{
echo "talaba1 va talaba2 ikki xil yo'nalish talabasi!";
}
}
else
{
```

```
echo "O'zgaruvchilar qiymatlari teng emas!";
}
?>
```

Natija: O'zgaruvchilar qiymatlari teng! talaba1 va talaba2 ikki xil yo'nalish talabasi!

Бу мисолни таҳлил қиламиз. Аввал ўзгарувчи қийматлари берилади, сўнг If оператори ёрдамида шарт текширилади, шарт бажарилса, кейинги қадам бажарилади ва шу билан бирга иккинчи шарт ҳам текширилади, иккинчи шарт ҳам бажарилса, кейинги қадамга ўтилади.

switch-case қайта уловчи оператори

Шартларни текшириш ва текшириш натижаларига кўра турли амалларни бажарувчи операторлардан бири switch операторидир. Ўзгарувчи қандай шартга эга эканлигига қараб, мос турли *бажариш_блоклари* бажарилади.

switch оператори if ва if...elseif операторига ўхшаш, фақатгина ундан фарқли томони бу операторда шарт иккитадан ортиқ қиймат қабул қилади. If операторида *шарт* true ва false қийматларни қабул қилади. switch операторида эса *шарт* оддий тип (integer, string ёки double) бўлган ихтиёрий турли қийматларни қабул қилади.

Синтаксиси:

```
switch (шарт ёки ўзгарувчи) {
  case қиймат1: бажариш_блоки1
  break;
  case қиймат2: бажариш_блоки2
  break;
  case қиймат3: бажариш_блоки3
  break;
  ...
  case қиймат4: бажариш_блоки4
  break;
  default: бирорта ҳам ҳолат юзага келмагандаги бажариш_блоки
  break;
}
```

Бу ерда *шарт* case(*қиймат1*, *қиймат2*, *қиймат3*, ...) даги қийматлар билан солиштирилади. Агар *шарт* ёки *ўзгарувчи* қиймат1, қиймат2, қиймат3, ... ларнинг қайсидир бири билан мос тушса, қийматдан кейинги икки нуқтадан биринчи break операторигача (агар бўлса) ёки switch охиригача мос *бажариш_блоки* бажарилади. Агар *шарт* ёки *ўзгарувчи* қиймат1, қиймат2, қиймат3, ... ларнинг бирортаси билан мос тушмаса default калит сўзидан кейинги *бажариш_блоки* бажарилади.

7.9 – мисол. Талабанинг исми, фамилияси, гуруҳи, тўлов-контрактни амалга ошириш йўлини танлаш имкони бўлган майдонлардан иборат форма яратамиз. Талаба ушбу формани тўлдириб, маълумотларни формани қайта ишловчи саҳифага жўнатади. Формани қайта ишловчи саҳифада танлаш операторларини ишлатишни кўриб ўтамиз.



7.1-расм. Тўлов-контрактни амалга ошириш йўлини танлаш формаси

7.1 –расмда берилган формани HTML кодининг қисми қуйида берилган.

```
<tr>
    <td>To'lov kontraktini qaysi yo'l bilan amalga oshirmoqchisiz?</td>
    <td><select name="usul">
        <option value = "plastik_kartochka">Plastik kartochka yordamida...
        <option value = "pul_utkazish">Pul o'tkazish yo'li bilan...
        <option value = "naqd">Naqd to'lov orqali...
    </select>
    </td>
</tr>
```

юқорида берилган кодни қайта ишлаш учун **if-elseif** дан фойдаланамиз.

```
<?
$usul="naqd";
if($usul=="plastik_kartochka")
echo "<p>Plastik kartochka yordamida...";
elseif($usul=="pul_utkazish")
echo "<p>Pul o'tkazish yo'li bilan...";
elseif ($usul=="naqd")
echo "<p>Naqd to'lov orqali...";
else
echo "<p>Talaba birorta ham yo'lni tanlamadi ";
?>
```

Natija:

Naqd to'lov orqali...

switch-case оператори ёрдамида:

```
<?
$usul="naqd";
switch ($usul) {
case "plastik_kartochka":
echo "<p>Plastik kartochka yordamida...";
break;
case "pul_utkazish":
echo "<p>Pul o'tkazish yo'li bilan...";
break;
case "naqd":
echo "<p>Naqd to'lov orqali...";
break;
default:
echo "<p>Talaba birorta ham yo'lni tanlamadi ";
break;
}
```

```
?>
```

```
Natija: Naqd to'lov orqali...
```

Агар `$usul` ўзгарувчиси бирорта ҳам қийматга тенг бўлмаса, у ҳолда натижа ўзгаради, яъни «Талаба бирорта ҳам йўлни танламади» ёзуви экранга чиқазилади.

Мисолнинг натижаси иккала ҳолат (if-else ва switch-case)да ҳам бир хил бўлади. Фақатгина if-elseif операторлари ёрдамида ёзилган вақтда, ҳар бир қатордаги *шарт* текширилади. switch оператори эса бошқачароқ тарзда ишлайди. switch-case операторлари ишга тушганда, *шарт* { } блок охиригача текширилади. Агар break оператори қўлланилса, унга дуч келганда, текшириш тўхтатилади.

7.2. Такрорлаш(цикл) операторлари

Php нинг фақатгина бирор бир шарт асосида ишловчи операторлари эмас, балки такрорланиш амалларини бажарувчи цикллар ҳам мавжуд. Улар while, do-while, foreach ва for.

Такрорлаш(цикл) оператори икки қисмдан иборат: тана ва бош қисмдан иборат. Тана қисми такрорлашда бажариладиган операторларни ўз ичига олади. Бош қисм эса танадаги операторларнинг такрорий бажарилишини таъминлайди.

While цикли

Синтаксиси:

while (*шарт*) {*бажариш_блоки*}

ёки

while (*шарт*) *бажариш_блоки* endwhile;

While – оддий цикл бўлиб, бу ерда *бажариш_блоки шарт* false қиймат қабул қилгунга қадар бажарилади. While циклини if дан фарқли тарафи шундаки, if операторида *шарт* текширилганда true қийматни қабул қилса, у ҳолда кейинги қадам фақатгина бир маротаба бажарилар эди. While циклида эса ҳар бир *шарт* бажарилган ҳолат учун *бажариш_блоки* бажарилади, яъни бажарилиш цикли такрорланади. *Шарт* ҳар сафар цикл бошида текширилади, шунинг учун агар *бажариш_блоки* амалга оширилаётган вақтда *шарт*га ўзгартириш киритилса ҳам цикл итерация охиригача, яъни *бажариш_блоки*нинг барча буйруқлари бажарилгунча тўхтатилмайди.

7.10 - мисол. While оператори.

1 дан 5 гача бўлган сонларни чиқазинг.

```
<?
$n=1;
while ($n<=5)
{ echo $n. "<br>";
  $n++;
}
?>
Natija:
1
2
3
4
5
```

7.11 – мисол. 1 дан 50 гача бўлган сонларнинг йиғиндисини ҳисобланг.

```
<?
$sum=0;
$i=1;
while($i<=50)
{
  $sum=$sum+$i;
  $i++;
}
echo "1 dan 50 gacha bo'lgan sonlar yig'indisi - $sum";
?>
Natija:1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275
```

do...while цикли

do...while цикли while циклига жуда ўхшаш, фақатгина битта фарқли тарафи шундаки, унда *шарт*нинг ростлиги цикл бошида эмас, балки охирида текширилади. Шунинг учун do...while циклининг *бажариш_блоки* ҳеч бўлмаганда бир маротаба бажарилади.

Синтаксиси:

do (*бажариш_блоки*) while (*шарт*);

7.12 – мисол. do...while оператори

1 дан 10 гача бўлган жуфт сонларни чиқазинг.

```
<?
$n=1;
do {
if ($n%2==0) echo $n. "<br>";
$n++;
}
while ($n<=10)

?>
Natija:
2
4
6
8
10
```

FOR цикли

Синтаксиси:

for (*амал1*; *шарт*; *амал2*)

{*бажариш_блоки*;}

амал1 – фақатгина бир маротаба цикл бошида бажарилади. Одатда бунда ҳисоблагичнинг бошланғич қиймати берилади.

шарт – шарт ҳар бир итерацияда текширилади. Агар шарт false қийматни қабул қилса, у ҳолда итерация тўхтатилади. Одатда шартда ҳисоблагич қиймати билан берилган қиймат солиштирилади.

амал2 – ҳисоблагич қийматини ўзгартириш учун, итерация охирида бажарилади.

бажариш_блоки – ҳар бир итерацияда бир маротаба бажарилади. Одатда бу ерда ўзида кодлар блокини жамлаган цикл танаси жойлашади.

амал1, *шарт*, *амал2* ларнинг ҳар бири бўш бўлиши мумкин. Агар *шарт* бўш бўлса, бу циклининг аниқланмаган вақтда бажарилиши кераклигини билдиради. (бу ҳолда `php` шартни ҳамма вақт рост деб ҳисоблайди). Бу цикл фойдасиздек кўриниши мумкин, аслида эса бу ҳолларда циклни `break` операторини қўллаш билан ҳам тўхтатиш мумкин.

7.13 - мисол: 1 дан 10 гача бўлган тоқ сонларни чиқазинг.

```
<?
for ($n=1; $n<=10; $n++)
{
if ($n%2!=0) echo $n. "<br>";
}
?>
Natija:
```

```
1
3
5
7
9
```

Шу мисолни $n \leq 10$ шартни ишлатмасдан, break операторини қўллаб ёзиш мумкин.

```
<?
for ($n=1;;$n++)
{
    if ($n%2!=0) echo $n. "<br>";
    if ($n>=10) break;
}
?>
```

Natija:

```
1
3
5
7
9
```

7.14 - мисол: 1 дан 50 гача бўлган сонларнинг йиғиндисини ҳисобланг.

```
<?
for($i=1;$i<=50;$i++)
{
    $sum=$sum+$i;
}
echo "1 dan 50 gacha bo'lgan sonlar yig'indisi - $sum";
?>
```

Natija: 1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275

Foreach

Бу оператор массивлар билан ишлаш учун мўлжалланган бўлиб, PHP нинг 4 версиясида киритилган. Турли типдаги массивлар учун foreach операторининг икки хил кўриниши мавжуд.

Синтаксиси:

foreach (*\$array as \$value*) {*бажариш_блоки*}

ёки

foreach (*\$array as \$key=> \$value*) {*бажариш_блоки*}

Биринчи навбатда \$array ўзгарувчисига берилган массивнинг ҳамма элементлари бўйича цикл шаклланади. Циклнинг ҳар бир қадамида массивнинг жорий элементи қиймати *\$value* ўзгарувчисига ёзилади ва массивнинг ички ҳисоблагичи бир бирликка ўзгаради(шунинг учун кейинги қадамда массивнинг кейинги элементи кўринади). *бажариш_блоки* ичида массивнинг жорий элементи қиймати *\$value* ўзгарувчиси ёрдамида олинади ва \$array ўзгарувчисига нечта элемент бўлса, *бажариш_блоки* шунча марта амалга оширилади.

foreach массивининг иккинчи тур синтаксисида эса, юқоридагига қўшимча равишда, циклнинг ҳар бир қадамида массивнинг жорий элементи калити *\$key* ўзгарувчисига ёзилади ва уни ҳам *бажариш_блоки* ичида қўллаш мумкин.

foreach ўз ишини бошлаганда, массивнинг ички кўрсаткичи автоматик тарзда биринчи элементга ўрнатилади.

7.15 - мисол. Foreach оператори

```
<?php
$names = array("Lola", "Karim", "Kamol");
foreach ($names as $val) {
    echo "$val <br>";
}
?>
```

Natija:Lola
Karim
Kamol

Иккинчи тури ассоциатив массивлар учун ишлатилади.

7.16 - мисол. Foreach оператори

```
<?php
$names = array(
    "Lola"=>7,
    "Karim"=>9,
    "Kamol"=>15);
foreach ($names as $i=>$startib_nomer)
{
    echo "$i ro'yhatda $startib_nomer nomerida.<br>";
}
?>
```

Natija:
Lola ro'yhatda 7 nomerida.
Karim ro'yhatda 9 nomerida.
Kamol ro'yhatda 15 nomerida.

7.3. Ўтиш операторлари

Айрим ҳолларда цикл ишини тезда тўхтатиш ёки бирор фрагментдан чиқиб кетиш талаб этилади. Ёки циклнинг охириги инструкциясидан кейин жойлашган дастур биринчи инструкцияга бошқарувни узатиши керак. Бунинг учун break ва continue операторларидан фойдаланилади.

Break(циклдан чиқиш)

Break оператори жорий циклнинг (for, foreach, while, do...while, switch) ишини тўхтатади. break оператори сонли параметр билан бирга қатнашган ҳолларда, сонли параметрга қараб бошқарувчи керакли структура ишини тўхтатади (ёки циклдан чиқади).

7.17-мисол. break оператори

```
<?php
$i=1;
while ($i) {
    $n = rand(1, 10);
    // 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
    echo "$i:$n";
    // iteratsiya tartibi va tanlangan sonlarni chiqaradi
    if ($n==5) break;
    /* Agar 5 soni tanlangan bo'lsa sikl ishini tugallaydi. Ushbu qatordan keyin
    bajarilishi kerak bo'lgan kodlar ishlamaydi */
    echo "Sikl ishlayapti <br>";
    $i++;
}
echo "<br>sikl iteratsiyalar soni $i";
?>
```

Natija:

```
1:10Sikl ishlayapti
2:7Sikl ishlayapti
3:3Sikl ishlayapti
4:4Sikl ishlayapti
5:6Sikl ishlayapti
6:6Sikl ishlayapti
7:8Sikl ishlayapti
8:10Sikl ishlayapti
9:4Sikl ishlayapti
10:4Sikl ishlayapti
11:7Sikl ishlayapti
12:2Sikl ishlayapti
13:2Sikl ishlayapti
14:2Sikl ishlayapti
15:5
sikl iteratsiyalar soni 15
```

rand функцияси ишлаётганлиги сабабли натижа ҳар гал турлича бўлиши мумкин.

Агар break операторидан кейин сон кўрсатилса, у ҳолда шу сонга тенг миқдордаги ушбу оператор қатнашган цикллар иши тўхтатади. 6.17 - мисол бу ҳолатга мисол бўла олмайди, чунки унда ичма ич жойлашган цикллар мавжуд эмас. Скриптга ўзгартириш киритамиз:

```
<?php
$i=1;
```



```

while ($i) {
    $n = rand(1, 10);
    // 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
    switch ($n){
        case 5:
            echo "<font color=blue>
            switch (n=$n)</font> dan chiqish";
            break 1;
            // switch ishini to'xtatamiz ( break qatnashgan birinchi siklni)
        case 10:
            echo "<font color=red>
            switch dan chiqish va
            while (n=$n)</font>";
            break 2;
            // switch va while ishini to'xtatamiz (break qatnashgan ikkala siklni)
        default:
            echo "switch ishlayapti (n=$n), ";
            }
            echo " while ishlayapti -$i qadam <br>";
            $i++;
            }
            echo "<br> Sikl iteratsiyalar soni $i ";
        ?>

```

Natija:

switch ishlayapti (n=4), while ishlayapti -1 qadam

switch (n=5) dan chiqish while ishlayapti -2 qadam

switch ishlayapti (n=9), while ishlayapti -3 qadam

switch dan chiqish va while (n=10)

Sikl iteratsiyalar soni 4

rand функцияси ишлаётганлиги сабабли натижа ҳар гал турлича бўлиши мумкин.

Continue(циклини давом эттириш)

Айрим ҳолларда цикл ишини бутунлай тўхтатмасдан, циклни янги итерациядан бошлашга тўғри келади, бу ҳолда continue операторини қўллаб янги итерацияни бошлаш мумкин. continue оператори *бажариш_блоки* нинг исталган циклидан кейинги инструкцияларни ишлашини тўхтатиб, янги доирада бажаришни давом эттиришни таъминлайди. Continue операторини сонли параметр билан қўллаш мумкин ва ушбу параметрда нечта ичма-ич жойлашган бошқарувчи конструкциянинг ишини тугатиш кераклиги кўрсатилади. Масалан, иккита ичма-ич жойлашган цикллар ичида Continue 2 буйруғи берилса, бошқарув иккила цикл ишлашини тўхтатади.

7.17 - мисолдаги break операторини continueга алмаштирамиз ва цикл қадамлари сонини 4 та билан тўхтатамиз.

```

<?php
$i=1;
while ($i<4) {
    $n = rand(1, 10);
    // 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
    echo "$i:$n";
    // iteratsiya generizatsiyalagan sonni chiqaramiz
    if ($n==5) {
        echo "<br> Yangi iteratsiya <br>";
        continue;
    }
    /* agar 5 soni generizatsiya qilingan bo'lsa, siklning yangi iteratsiyasini b

```

```

oshlaymiz $i o'zgarmaydi */
}
echo "sikl ishlayapti <br>";
$i++;
}
echo "<br>".$i ."siklning iteratsiyalar soni";
?>

```

Natija:

1:4sikl ishlayapti

2:4sikl ishlayapti

3:5

Yangi iteratsiya

3:8sikl ishlayapti

4siklning iteratsiyalar soni

rand функцияси ишлаётганлиги сабабли натижа ҳар гал турлича бўлиши мумкин.

Continue оператори бажарилгандан сўнг цикл иши тўхтамайди. Бу ерда цикл ҳисоблагичи 5 сонини олишда ўзгармайди, чунки у continue операторидан кейин жойлашган. Аслида биз continue ёрдамида 5 сони пайдо бўлган ҳолдан қочишга ҳаракат қиламиз. Шунинг учун continue операторини қуйидаги алмаштириш билан ёзиб кўрамиз:

```

<?php
$i=1;
while ($i<4) {
    $n = rand(1, 10);
    // 1 dan 10 gacha bo'lgan sonlarni tanlash
    if ($n!=5) {
        echo "$i:$n <br>";
        // iteratsiya va generizatsiya qilingan sonni chiqaramiz
        $i++;
    }
}
?>

```

Natija:

1:2

2:1

3:7

rand функцияси ишлаётганлиги сабабли натижа ҳар гал турлича бўлиши мумкин.

PHP да continue операторининг қўлланилишининг битта асосий тарафи мавжуд - у switch конструкцияларида худди break дагидек ишлайди. Агар switch цикл ичида жойлашган бўлса ва янги итерацияни бошлаш керак бўлса, continue 2 ни қўллаш керак.

7.4. Қамраб олиш операторлари

Оддий, лекин жуда кўп ишлатиладиган операторлар: `require()`, `include()` ларни кўриб ўтамиз. `require()`, `include()` операторлари ёрдамида PHP саҳифага файлни юклаб олинади. Файл ўзида матнларни, HTML дескрипторларни, PHP классларни, PHP функцияларни, PHP операторларни жамлаган бўлади.

WEB сайт яратишда файлни юклаб олиш нима учун керак? Сайт яратилаётган вақтда сайтни блоklarга ажратиб дастур кодларини ёзиш мақсадга мувофиқ. Масалан, сайтни юқори қисми блоки, меню қисми блоки, асосий қисми блоки, ўнг томони блоки ва шунга ўхшаш. Блокларга ажратиб ёзиш, сайтни бирор бир қисмига ўзгартириш киритишда қулай бўлиб (масалан, янги меню қўшиш ёки мавжуд менюлардан бирортасини олиб ташлаш), у ҳолда фақатгина ўша блокка ўзгартириш киритилади холос. Ана шу блоклар файлларда сақланади ва ушбу файлларни `require()`, `include()` операторлари ёрдамида юклаб олиш мумкин. `require()`, `include()` операторлари бир-бирига ўхшаш ҳолатда ишлайди.

include() оператори

Include оператори кўрсатилган файлдаги барча кодларни очиш учун ишлатилади ва дастурда бу оператор неча марта иштирок этса, у шунча марта бажарилади.

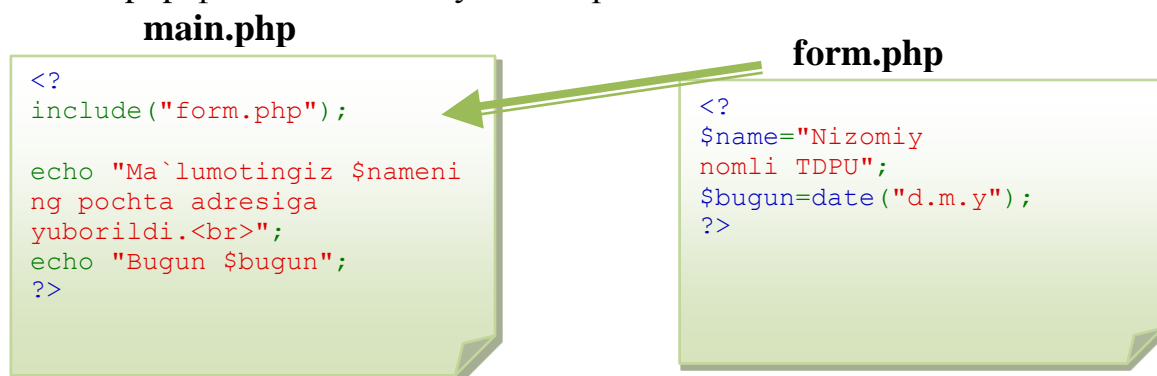
Синтаксиси:

***include** 'fayl_nomi';ёки*

***include** \$fayl_nomi;ёки*

***include** ("fayl_nomi");*

Масалан, `form.php` файлида функция ёки ўзгарувчи қийматлари жойлашган бўлсин. Асосий дастур кодини бирор қисмида ушбу функция ёки ўзгарувчи қийматларидан фойдаланиш керак бўлса `include("form.php")` ни кўямиз. Агар бир неча марта `form.php` дан фойдаланиш керак бўлса, у ҳолда асосий дастурнинг керакли қисмида бир неча марта қўллаш мумкин. Саҳифа бажарилаётган вақтда `include()` оператори `include("form.php");` буйруғи ўрнига `form.php` файлидаги мавжуд кодларни алмашади.



7.18 – мисол. `include()` қамраб олиш операторининг ишлатилиши `form.php` файли

```
<?
$name="Nizomiy nomli TDPU";
$bugun=date("d.m.y");
?>
```

main.php файли

```
<?
include("form.php");
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.<br>";
echo "Bugun $bugun";
?>
```

Натижа шуни кўрсатадики main.php ишга туширилганда саҳифа куйидаги кўринишга эга бўлади.

```
<?
$name="Nizomiy nomli TDPU";
$bugun=date("d.m.y");
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.<br>";
echo "Bugun $bugun";
?>
```

Natija:

Ma'lumotingiz Nizomiy nomli TDPU ning pochta adresiga yuborildi.

Bugun 26.01.11

Файл кенгайтмаси ва include() оператори

PHP нинг **include()** оператори мурожаат этилаётган файлнинг кенгайтмасига аҳамият қаратмайди. Файлнинг номи қулай тарзда номланиши мумкин. include() оператори ёрдамида юкланаётган файл PHP файлининг бир қисми сифатида бўлади ва қандай бўлса, шундай бажарилади. Қамраб олинаётган(юкланаётган) файл учун ихтиёрий кенгайтмани қўллаш мумкин, шулардан бири .inc. Агар файл .inc кенгайтмаси ёки бирорта бошқа стандарт бўлмаган кенгайтма билан WEB хужжатлар каталогида жойлашган бўлса ва фойдаланувчи бу файлни браузер ёрдамида ишга туширса, у ҳолда файлдаги кодларни оддий матн шаклида кўриш мумкин, яъни файл қайта ишланмайди. Шунинг учун мурожаат этиладиган файлларни бирор бир каталогда сақлаш тавсия этилади ёки стандарт кенгайтмадан фойдаланиш керак бўлади.

PHP операторлари масалан page.html каби номланган файлларда иштирок этса бажарилмайди. Аксинча, агар include() оператори ёрдамида page.html файли юклаб олинса, файл ичида сақланаётган PHP операторлари қайта ишланади. Қоидага кўра, PHP фақатгина .php каби кенгайтмага эга файллардагина операторларига мурожаат этиб, бошқа файлларни таҳлил қилади.

Демак form.php даги дастур коди main.php файлининг бир қисмига эквивалент. form.php файлини PHP код эканлигини англатувчи тегларсиз оддий матн сифатида ёзиш мумкинми? Йўқ, чунки файлни қўйиш вақтида PHP ни қайта ишлаш режимидан HTML режимига ўтади. Шунинг учун юкланаётган файл ичидаги PHP скрипти кўринишида қайта ишланадиган код мос теглар билан ёзилиши керак.

Юкланаётган файлларни қидириш қуйидаги тартибда амалга оширилади:

1. Файлни қидириш аввал жорий иш каталогига тегишли include_path да олиб борилади.
2. Агар файл топилмаса, у ҳолда қидирув жорий скрипт каталогига

тегишли `include_path` да амалга оширилади.

3. PHP созловлар файлида аниқланган `include_path` параметрида юкланадиган файлларни қидириш мумкин бўлган каталоглар номи берилади.

Масалан, бизнинг мисолда жорий иш каталоги бу - `/www/` каталогидир. Асосий файл `main.php` файлига `include/a.php` файли ва навбати билан `b.php` файли юкланаётган бўлса, парсер биринчи ўринда `b.php` файлини `/www/` каталогидан қидиради ва агар бундай файл топилмаса, у ҳолда кейин `/www/include/` каталогидан қидиради.

`include()` ёрдамида юкланаётган файл таркибидаги ўзгарувчилар `include()` жойлашган соҳага боғлиқ равишда ишлатилади. Юкланаётган файлдаги ихтиёрий ўзгарувчи юкланаётган файлни ўзида ишлатилиши мумкин. Агар `include()` функция ичида юкланса, у ҳолда юкланган файл таркибидаги код функция ичида аниқланганлар каби бўлади.

`include()` оператори ёрдамида локал файллардан ташқари `url` – манзиллари кўрсатилган ташқи файлларни ҳам юклаш мумкин. Ушбу имконият PHP созлов файлидаги `url_fopen_wrappers` директивасида назоратга олинади ва одатда бу ёқилган бўлади. Лекин windows учун PHP нинг 4.3.0 версиясигача бу имконият мавжуд бўлмаган.

`include()` бу махсус конструкция, шунинг учун ички шартли блоклар қўлланилишида уни фигуралик қавс ичига олиб ёзилади.

```
<?
If ($value=true)
{
include("file1.inc");
}
Else
{
include("file2.inc");
}
?>
```

`include()` операторини қўллашда икки хил турдаги хатолик юзага келиши мумкин – **юклаш хатолиги** (масалан, юклаш буйруғини ўзи нотўғри ёзилган бўлса, кўрсатилган файлни топиш мумкин бўлмаса ва ҳ.) ёки **бажарилиш хатолиги** (агар юкланаётган файлда хато мавжуд бўлса).

require() оператори

`require()` ва `include()` операторлари жуда ўхшаш бўлиб, ишлаш принципага кўра бир-биридан фарқ қилади. `require()` оператори тахминан C++ даги `# include` каби ишлайди. `include()` учун айтилганларнинг барчаси `require` учун ҳам ўринлидир. `require` шунингдек дастурга боғланишга ва бирор-бир файлни амалга оширишга имкон яратади. `require` ва `include` нинг асосий фарқи унинг хатоларга қандай реакция беришидир. `include` огоҳлантиради ва `script` ишини давом эттираверади. `require` да эса хато скрипт ичида фотал хатони чиқаради ва уни бажарилишини тўхтатади. Савол туғилиши мумкин: у ҳолда `require()` оператори нима учун қўлланилади, унинг `include()` операторидан қандай фарқи бор? Масала шундаки, `require()` оператори `include()` операторидан бир мунча тезроқ ишлайди.

`require()` оператори шартли оператор ичида қўлланилса, яхши самара бермайди. Агар ҳеч бўлмаганда ушбу оператор қўлланилган катор

бажарилмаса, унда юкланган файлдан битта қатор ҳам бажарилмайди. Шунингдек цикллар ҳам require() га таъсир қилмайди. Ҳатто юкланган файл цикл объекти бўлса ҳам, юклаш ўз-ўзидан бир маротабагина амалга оширилади.

PHP нинг 4.0.2 версияларигача require() нинг қўлланилиши, интерпретатор албатта юкланаётган файлни ўқишга ҳаракат қилишини англатган.

require() ни ҳам include() каби шартли блоклар ичида қўллаганда фигурали қавс ичига олиб ёзиш керак.

```
<?
If ($value=true)
{
require("file1.inc");
}
Else
{
require("file2.inc");
}
?>
```

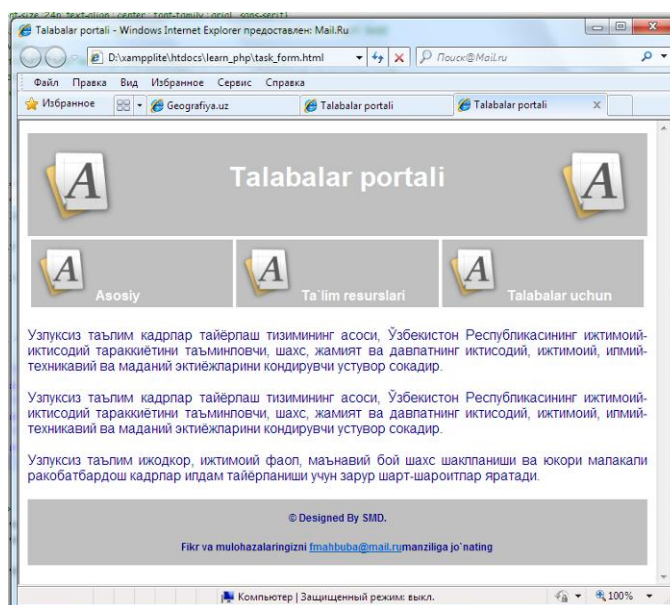
require() оператори шартли блоклар ичида қўлланилганда, ҳар сафар саҳифа бажарилганда, барча файллар юклаб олинади, лекин улардан фақатгина биттаси \$value ўзгарувчиси қийматига қараб бажарилади.

Агар require() операторининг ўрнида include() оператори қўлланилса, у ҳолда \$value ўзгарувчиси қийматига қараб фақат битта файл юклаб олинади ва бажарилади.

require() операторининг Web –сайт шаблонлари учун қўлланилиши

Web –сайт саҳифалари бир хил кўринишдан иборат бўлса, у ҳолда PHP дан фойдаланиб require() оператори ёрдамида саҳифага шаблон ва стандарт элементларни қўшиш мумкин.

Масалан, Web –сайт 7.2 - расмдаги кўринишдаги каби бўлсин. Саҳифаларда қуйидагича ҳолатлар юзага келиши мумкин: Web – сайт анча вақтдан бери ишлаб келмоқда ва у бир неча ўн, юз ва минглаб саҳифадан иборат ва уларнинг ҳаммаси бир хил кўринишда бўлиши мумкин. Унинг стандарт кўринишига кичкина ўзгартириш киритиш лозим, масалан, унинг қуйи қисмидаги электрон почта манзилини ўзгартириш ёки саҳифанинг менюлар сатрига янги меню қўшиш лозим бўлсин. У ҳолда ўн, юз, минглаб



7.2 – расм. Талабалар порталининг Web саҳифаси

саҳифани ўзгартиришга кимга ёқади? Шунинг учун саҳифани HTML кодини мақбул йўл билан қисмларга ажратилади. 7.2- расмдаги саҳифанинг коди 7.1 - листингда берилган.

7.1 – листинг. *task_form.html – Талабалар порталининг асосий саҳифасининг HTML коди.*

```
<html>
<head>
  <title>Talabalar portali</title>
  <style>
    <!--
      h1 {color: white; font-size :24p; text-align : center ; font-
family : arial , sans-serif}
      .menu {color: white; font-size: 12pt; text-align : center ; font-
family : arial , sans-serif; font-weight : bold}
      td {background: silver}
      p {color :darkblue; font-size: 12pt; text-align: justify;font-
family: arial , sans-serif}
      p.foot {color: darkblue; font-size :9pt; text-align: center; font-
family : arial , sans-serif ; font-weight : bold}
      a: link , a : visited, a : active {color : white }
    -->
  </style>
</head>
<body>
  <!-- page header -->
  <table width="100%" cellpadding = 12 cellspacing = 0 border = 0>
    <tr bgcolor = silver>
      <td align = left><img src = "logo.png "></td>
      <td>
        <h1>Talabalar portali</h1>
      </td>
      <td align = right><img src = "logo.png "></td>
    </tr>
  </table>
  <!-- menu -->
  <table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
    <tr >
      <td width = "25%">
        <img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
      <td width = "25%">
        <img src = "s-
logo.png "> <span class=menu>Ta`lim resurslari</span></td>
      <td width = "25%">
        <img src = "s-
logo.png "> <span class=menu>Talabalar uchun</span></td>
    </tr>
  </table>
  <!-- page content -->
  <p>Ўзлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Респ
убликасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-
техникавий ва маданий эктиёжларини қондирувчи устувор соҳадир. </p>
  <p>Ўзлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Респ
убликасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-техникавий ва маданий эктиёжларини қондирувчи устувор соҳадир.
</p>
  <p>Ўзлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шакллани
ши ва юқори малакали рақобатбардош кадрлар илдам тайёрланиши учун зарур шарт-
шароитлар яратади.</p>
  <!-- page footer-->
```



```

<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr>
<td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@mail.ru">fmahbuba@mail.ru</a>manziliga jo`nating </p>
</td>
</tr>
</table>
</body>
</html>

```

7.1 – листингдан кўриниб турибдики, бу файлда кодлар алоҳида сатрларга ажратилган.

“page header” (“саҳифанинг юқори колонтигули”) билан эълон қилинган қисмда порталнинг номи ва логотипи, “menu” (“меню”) қисмида менюлар панели, “page content” (“саҳифанинг таркиби”) қисмида берилган саҳифа учун матн ва сўнгида эса саҳифанинг қуйи колонтигули жойлашади. Бу файлни ажратиш ва унинг қисмларини мос равишда header.inc, home.php ва footer.inc каби номлаш керак. header.inc ва footer.inc файлларида бошқа саҳифаларда такрорланувчи кодлар ишлатилади.

home.php файли эса task_form.html файли ўрнига алмаштирилади ва у саҳифанинг уникал таркиби ва иккита require() операторидан ташкил топади (7.2 – листингдан кўрсатилгани каби).

7.2.1 – листинг. home.php – талабалар порталининг асосий саҳифасини яратувчи PHP код.

```

<?
require("header.inc");
?>

<!-- page content -->
<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Респ
убликасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-
техникавий ва маданий эктиёжларини қондирувчи устувор соҳадир. </p>
<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Респ
убликасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-техникавий ва маданий эктиёжларини қондирувчи устувор соҳадир.
</p>
<p>Узлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шакллани
ши ва юқори малакали рақобатбардош кадрлар илдам тайёрланиши учун зарур шарт-
шароитлар яратади.</p>
<?
require("footer.inc");
?>

```

home.php файлида require() оператори header.inc ва footer.inc файлларини юклайди.

header.inc файли ушбу саҳифада ишлатилувчи CSS – аниқловчи (Cascading Style Sheet, CSS), портал номи ва менюлар сатри тасвирланадиган жадвалдан ташкил топган.

7.2.1 – листинг. header.inc – талабалар порталининг барча Web саҳифалари учун такрорланиб ишлатилувчи юқори колонтигули.

```

<html>
<head>
  <title>Talabalar portali</title>
  <style>
    <!--
      h1 {color: white; font-size :24p; text-align : center ; font-
family : arial , sans-serif}
      .menu {color: white; font-size: 12pt; text-align : center ; font-
family : arial , sans-serif; font-weight : bold}
      td {background: silver}
      p {color :darkblue; font-size: 12pt; text-align: justify;font-
family: arial , sans-serif}
      p.foot {color: darkblue; font-size :9pt; text-align: center; font-
family : arial , sans-serif ; font-weight : bold}
      a: link , a : visited, a : active {color : white }
    -->
  </style>
</head>
<body>
  <!-- page header -->
  <table width="100%" cellpadding = 12 cellspacing =0 border = 0>
    <tr bgcolor = silver>
      <td align = left><img src = "logo.png "></td>
      <td>
        <h1>Talabalar portali</h1>
      </td>
      <td align = right><img src = "logo.png "></td>
    </tr>
  </table>
  <!-- menu -->
  <table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
    <tr >
      <td width = "25%">
        <img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
      <td width = "25%">
        <img src = "s-
logo.png "> <span class=menu>Ta`lim resurslari</span></td>
      <td width = "25%">
        <img src = "s-
logo.png "> <span class=menu>Talabalar uchun</span></td>
    </tr>
  </table>

```

footer.inc файли ҳар бир саҳифанинг қуйи қисмида ишлатилувчи қуйи колонтигул тасвирланадиган жадвалдан ташкил топган.

7.2.1 – листинг. header.inc – талабалар порталининг барча Web саҳифалари учун такрорланиб ишлатилувчи қуйи колонтигули.

```

<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
  <tr>
    <td>
      <p class=foot>&copy; Designed By SMD.</p>
      <p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@mail.ru">fmahbuba@mail.ru</a>manziliga jo`nating </p>
    </td>
  </tr>
</table>
</body>
</html>

```

VIII БОБ. ФУНКЦИЯЛАР

Бу бобда функция тушунчаси, функциялар, фойдаланувчи томонидан аниқланган функциялар, функция параметрлари, параметрларни қиймати ва жўнатмаси бўйича узатиш, қайтариладиган қийматлар ва жўнатмалар (return()) функциялар) билан танишилади.

Функция кўпгина дастурлаш тилларида ишлатилади. Функция дастурда батафсил аниқланган топшириқни алоҳида бажариш учун ёзилган буйруқлар тўплами(кодлар)ни белгилаш учун ишлатилади. Функция ҳар гал топшириқни бажариш учун буйруқлар тўпламини ўқиш ва қайта ишлатиш имкониятини яратади. Функциялар икки хил бўлади: **РНР** да **мавжуд функциялар(ички функциялари)** ва **фойдаланувчи томонидан аниқланган функциялар**.

РНР нинг **ички функциялари** файллар билан ишлаш, маълумотлар базасини ишлатиш, график тасвирлар яратиш ва бошқа серверларга боғланиш учун имкон беради.

Баъзи ҳолларда дастур тузиш вақтида ички функциялар имкониятларидан ташқари бирор-бир нарсани бажаришга тўғри келади. Бу ҳолларда дастурчиларнинг имкониятлари чегараланмаган, яъни ички функциялардан ташқари улар ўзлари функция яратишлари мумкин ва бу функциялар **фойдаланувчи томонидан аниқланган функциялар** дейилади. Фойдаланувчи томонидан аниқланган функция ўзида нимани мужассамлаштирганини кўриб ўтсак. Функция дастурлашда ҳам математикага ўхшаб бир нечта параметрларни уларнинг қийматига акс этишини ифодалайди, яъни функция ишининг натижаси бўлган ҳар бир берилган параметрга қандайдир қиймат қайтаради. Топшириқни бажариш учун яратилган код бирор-бир топшириқни бажарувчи махсус, мантикий ва мавжуд функциялар ва амаллар комбинациясидан иборат. Бу код функция эълон қилинганда берилиб, саҳифани бир нечта қисмида ёки бир нечта саҳифаларида функцияга мурожаат этилганда бажарилади.

8.1. Фойдаланувчи томонидан аниқланган функциялар

Фойдаланувчи томонидан аниқланган функцияни эълон қилиш янги функцияни яратади. Функция кейинчалик фойдаланувчи томонидан аниқланган функция эканлиги ҳақида фойдаланувчига хабар бериш ва дастурда РНР синтаксисини таҳлил қилиш учун билан **function** калит сўзи ёрдамида эълон қилинади. Функция **function** калит сўзи билан бошланиб, функция номини, талаб қилинаётган параметрлар ва функцияга ҳар гал мурожаат этилганда бажариладиган кодларни ўз ичига олади.

Умумий ҳолда фойдаланувчи томонидан аниқланган функцияларни эълон қилиш қуйидаги кўринишда бўлади.

Синтаксиси:

```
function funksiya_nomi(parametr1, parametr2, ..., parametrN)
{ harakatlar_maydoni
  return qaytariladigan_funksiya_qiymati;
}
```

Бу ерда

funksiya_nomi - функция номи;

harakatlar_maydoni - функция танаси, яъни бирор-бир топширикни бажарувчи кодлар;

parametr1, ..., parametrN - лар эса алмашувчи маълумотлар.

Функция қуйидаги талабларга жавоб бериши керак:

1. функция номи ва параметр номлари РНР нинг номлаш қоидаларига мос тушиши керак (криллча белгиларни қўлламаслик мақсадга мувофиқ). Функция номлари регистрга боғлиқ эмас.

2. функция параметрлари ўзгарувчилар ҳисобланади. Шунинг учун уларнинг ҳар бири олдида \$ белгиси ишлатилиши керак. Параметрлар рўйхатида ... (уч нукта) ишлатиш мумкин эмас.

3. *harakatlar_maydoni* сўзи ўрнида бирор-бир вазифани бажарувчи РНР код туриши керак. (*parametr* ларга боғлиқ бўлиши шарт эмас).

РНР нинг ички функциялари барча саҳифаларда ишлай олади. Фойдаланувчи томонидан аниқланган функциялар эса фақатгина улар эълон қилинган саҳифаларда ишлайди. Бундан келиб чиқадики, тез-тез ишлатиладиган фойдаланувчи томонидан аниқланган функцияларни алоҳида файлда сақлаш керак. Ва бу функцияларга қайси саҳифада мурожаат этилса, шу саҳифани бошида `require()` ёки `include()` операторлари ёрдамида функциялар жойлашган файл юклаб олинади.

Функцияни ички коди фигурали қавс ичида берилиб, талаб қилинган топширикни бажаради.

Агар функция ичида РНР дан чиқиб, HTML матнга ўтиш керак бўлса, саҳифани ихтиёрий жойида РНР дескриптори қуйидагича ёпилиб, очилади:

```
<?
function my_funksiya()
{
?>
Mening funksiyam elon qilindi
<?
}
?>
```

8.2. Регистр ва функция номи

Функция номини берилиши ўзгарувчи номини беришдан фарқ қилади. Ўзгарувчи номи регистрга сезгир шунинг учун \$Nomi ва \$nomi – турли хил ўзгарувчилар, лекин Nomi() ва nomi() - битта функция. Функция номи регистрга аҳамиятсиз, шунинг учун funksiya_nomi(), Funksiya_Nomi() ёки FUNKSIYA_NOMI() мурожаатлари битта натижани беради. Функция номларида ҳарфларни ихтиёрий (катта, кичик) ҳолда ишлатиш мумкин.

Функция номи қисқа ва тушунарли бўлиши керак. Агар функция саҳифани юқори қисми учун яратилаётган бўлса, у ҳолда pageheader() ёки page_header() каби номланиши мумкин.

Функция номига қуйидагича чегаралар қўйилади:

- функция **РНР нинг ички функциялари** номи билан номлана олмайди;

- функция номи фақатгина ҳарфлардан, рақамлардан ва чизиқча белгисидан иборат бўла олади.

- функция номи рақам билан бошлана олмайди.

Кўпгина дастурлаш тилларида функция номи қайтарилишига рухсат бор. Бу функцияни қайтарилиши(перегрузка) деб аталади. Фақатгина РНР қайтарилган функцияни қўлламайди, шунинг учун функция номи **РНР нинг ички функциялари** номлари ва аввал **фойдаланувчи томонидан аниқланган функция** номлари билан бир хил бўлмаслиги керак.

Аввал айтиб ўтганимиздек, **РНР нинг ички функциялари** барча РНР саҳифаларда ишлаши мумкин, **фойдаланувчи томонидан аниқланган функция** эса фақатгина эълон қилинган саҳифалардагина ишлай олади. Бу шуни англатадики, эълон қилинадиган функция номлари бошқа файлларда такроран ишлатилиши мумкин, лекин бу чалғишга (адашишга) олиб келади. Шунинг учун бу ҳолатни қўлламаган маъқул.

Функция номлари қуйидагича бўлиши мумкин:

Name();

Name2();

Name_tree();

_namefour;

Функция номлари қуйидагича бўлиши мумкин эмас:

5name()

Name_six;

Forpen()

(охирги функция номи **РНР нинг ички функцияси**)

8.3. Функцияга мурожаат этиш

Функцияга қандай мурожаат этилади? Аввал функция номи берилади ва қавслар ичида унинг параметрлари қийматлари рўйхати киритилади.

Синтаксиси:

funksiya_nomi();

бу код бирор бир параметрни талаб қилмайдиган *funksiya_nomi* номли функцияга мурожаат этади. Функцияга бу тарзда мурожаат этиш функция қайтарадиган барча қийматларни инкор этади. Кўпгина функциялар шу тарзда берилади.

Мисол учун, **PHP** нинг ички функцияларидан бири `phpinfo()` функцияси. Бу функция ўрнатилган PHP версияси ва у ҳақида маълумот олиш, PHP ва сервернинг турли ўзгарувчилари қийматларини кўрсатиш учун ишлатилади. Бу функция ҳеч қандай параметр қабул қилмайди ва унинг қайтарадиган қиймати ҳам инкор этилади. Шунинг учун `phpinfo()` нинг берилиши қуйидагича:

`phpinfo();`

Фойдаланувчи томонидан аниқланган функцияга мурожаат этиш PHP нинг ички функцияларига мурожаат этиш каби бўлади.

8.1 - мисол. Оддий функцияни эълон қиламиз:

```
<?
function my_function() // my_function() funksiyasi e`lon qilindi
{
Echo "Mening funksiyam elon qilindi";
}
my_function(); // my_function() funksiyasiga murojaat etildi
?>
Natija:Mening funksiyam elon qilindi
```

Функцияга қачон мурожаат этилади?

PHP нинг 4 версиясигача функцияга у аниқлангандан сўнг мурожаат этилади, яъни дастурлаш қаторида

`function funksiya_nomi() {...}`

блокидан сўнг.

Лекин PHP 4 да эса бундай талаб йўқ. Ҳаммаси интерпретатор олинган кодни қандай қайта ишлашига боғлиқ. Фақат шартли аниқланадиган функциялар (шартли оператор ва бошқа функциялар ишида) бундан ҳоли.

8.2-мисол. Шартли ўтиш оператори ичида функциянинг ишлатилиши

```
<?
$make = true;
/* bu yerda Make_event()ga murojaat etish mumkin emas;
chunki u bu yerda mavjud emas,
lekin Save_info()ga murojaat etish mumkin */
Save_info("Vohid", " Fozilov", "Men Informatika fanini tanladim");
if ($make)
{
// Make_event() funksiyasining aniqlanishi
function Make_event()
{
echo "<p> Tarix fanini o`rganishni xoxlayman";
}
}
```

```

}
// endi Make_event()ga murojaat etish mumkin
Make_event();
// Save_info funksiyasining aniqlanishi
function Save_info($first, $last, $message)
{
    echo "<br>$message<br>";
    echo "Ismi sharifi: ". $first . " ". $last . "<br>";
}
// Save_info ga bu yerda ham murojaat etish mumkin
Save_info("Laziz", "Abdullayev", "Men Falsafa fanini tanladim");
?>

```

Natija:

Men Informatika fanini tanladim

Ismi sharifi: Vohid Fozilov

Tarix fanini o`rganishni xoxlayman

Men Falsafa fanini tanladim

Ismi sharifi: Laziz Abdullayev

Агар функция дастурда бир марта аниқланган бўлса, уни иккинчи мартаба қайта аниқлаш ёки ўчириб ташлаш мумкин эмас. Функциялар регистрга боғлиқ эмаслигига қарамасдан уларга қандай ном билан аниқланган бўлса, ўша ном билан мурожаат этиш лозим.

8.4. Параметрлар (ёки Аргументлар)

Кўпгина функцияларда ўз топшириқларини бажариш учун бир ёки ундан ортиқ *параметрлар*ни узатилишини талаб этилади. *Параметрлар* маълумотларни, қийматларни функцияга етказиш имконини беради. Ҳар бир *параметр* ўзида ўзгарувчини ва константани мужассамлаштиради. Улар () қавс ичида функция номидан сўнг берилади.

funksiya_nomi("parametr1 uchun qiymat", "parametr2 uchun qiymat",...);

Бу ерда "parametr1 uchun qiymat", "parametr2 uchun qiymat",... лар ўрнида параметрларнинг қийматлари берилади. **Масалан,**

funksiya_nomi(2);

funksiya_nomi(7.993);

funksiya_nomi(\$variable);

Охирги берилган функцияда \$variable ўзгарувчиси ихтиёрий типда, хатто массив ҳам бўлиши мумкин.

Параметр қийматлари ёки қийматларни ўзида жамлаган ўзгарувчи номлари турли йўл билан узатилади. Функцияга маълумотларни функция *параметрлари* ёрдамида уч хил йўл билан узатиш мумкин:

- 1) қиймати бўйича (одатга кўра ишлатилади)
- 2) жўнатмаси (ссылка) бўйича
- 3) кўринмаслиги бўйича

Функция *параметрларини* оддий йўл билан узатиш **қиймати бўйича** узатиш деб аталади. *Параметрлар*ни узатишда узатилаётган қийматни ўзида жамлаган янги ўзгарувчи яратилади. У мавжуд (ички) ўзгарувчини нусхаси. Бу қиймат ихтиёрий тарзда ўзгариши мумкин, лекин мавжуд (ички) ўзгарувчини қиймати функция ташқарисида ўзгармай қолади. *Параметр* функцияга қиймати бўйича узатилаётганда параметрнинг функция ичида ўзгариши унинг функция ташқарисидаги қийматига таъсир қилмайди.

Қийматни кетма-кет ошириб боровчи функцияни яратамиз.

8.3 - мисол. Параметрларни қиймати бўйича узатиш

```
<?
Function increment($value, $a=1)
{
$value=$value+$a;
}
$value=15;
increment($value);
Echo $value;
?>
Natija:15
```

Натижа 15 чиқади, яъни \$value ўзгарувчини қиймати ўзгармайди.

Демак, бу код биз талаб этган вазифани бажармади. Бу эса *harakatlar_maydoni* билан боғлиқ. Код вазифасига кўра \$value ўзгарувчисини яратади, сўнг дастур increment() функциясига мурожаат этади. Функция ичида \$value ўзгарувчиси унга мурожаат этишига кўра яратилади, сўнг унинг қийматига 1 кўшилади ва функция ичида \$value қиймати 16 га тенг бўлади, лекин натижани экранга чиқаришга буйруқ берилмаганлиги ва функция ташқарисида \$value ўзгарувчиси глобал майдонда аниқланган бошқа

ўзгарувчи бўлганлиги учун ўзгармай қолади, шунинг учун натижа 15 га тенг бўлади.

Бу муаммони ҳал қилиш йўлларида бири бу - `$value` ўзгарувчисини функцияда глобал сифатида эълон қилиш керак, лекин бу шуни англатадики, бу функцияни ишлатишда унинг қиймати оширилиши керак бўлган ўзгарувчисини `$value` деб номлаш талаб этилади. Бундан ҳам мақбул йўл сифатида **жўнатма бўйича** узатишни айтишимиз мумкин. Функция *параметрларини* ўзгартиришга рухсат бериш учун уларни **жўнатма (ссылка) бўйича** узатиш керак.

Жўнатма бўйича параметрлар узатилаётганда янги қиймат яратиш билан бирга функция жўнатмадан ички ўзгарувчиларни қабул қилади. Бу жўнатма `$` доллар белгиси билан бошланувчи ўзгарувчи номи ёки бирор бир ихтиёрий ўзгарувчи. Фарқи шундаки *параметр* олаётган қийматни ички ўзгарувчига узатади.

Параметрларни жўнатма ёрдамида узатишда функция эълон қилинаётган вақтда *параметр* олдида (`&`) амперсанд белгиси қўйилади. Функцияга мурожаат этишда эса ҳеч қандай ўзгартириш киритиш шарт эмас.

8.2 - мисолни камчилигини тўғрилаймиз.

8.3 – мисол. Параметрларни жўнатма бўйича узатиш.

```
<?
Function increment (&$value, $a=1)
{
    $value=$value+$a;
}
$b=15;
Echo $b.'  
';
increment($b);
Echo $b;
?>
Natija:
15
16
```

Функция оддий йўл билан узатилган бўлса, унда *параметрлар* қийматини аниқлаш мумкин. Оддий йўл бўйича узатилган қиймат ўзгарувчи ва класс вакили ёки бошқа функциядан чақирилган бўлмасдан константа бўлиши керак.

Параметрларни кўринмаслиги бўйича узатиш.

Бизда берилган *параметрга* боғлиқ ҳолда ўзгарадиган информацион хабар яратувчи функция мавжуд. Агар функция *параметри* қиймати берилмаган бўлса, у ҳолда қиймат ўрнида “Ma`muriyat” ёзуви қўлланилади.

8.4 - мисол. Параметрларни кўринмаслиги бўйича узатиш

```
<?php
function xabar($sign="Dekanat"){
    // bu yerda sign «Dekanat» mazmuniga ega
    echo "Keyingi majlis ertaga bo`ladi.";
    echo "$sign .<br>";
}
xabar();
// funksiyani parametrsiz chaqiramiz. Bu holda yozuv -Dekanat
xabar("Hurmat bilan Afzalov");
// natija «Hurmat bilan Afzalov» bo`ladi
```

```
?>
```

Natija:

Keyingi majlis ertaga bo`ladi.Dekanat .

Keyingi majlis ertaga bo`ladi.Hurmat bilan Afzalov .

Бир нечта *параметрлар* билан берилган функцияни оддий йўл билан эълон қилинаётганда, қиймати билан берилган параметрлар ҳамма параметрлардан кейин ёзилиши керак. Акс ҳолда, функцияга мурожаат этилаётганда бу параметрлар ташлаб юборилади ва натижа хато чиқади.

8.5 - мисол. Талаба томонидан бирор-бир мақола каталогга қўшилиши керак. Мақолани каталогга қўшиш учун саҳифада мақоланинг номи, муаллифи ва қисқа изоҳи кабилар тавсифлари киритилади. Агар талаба мақоланинг муаллифини киритмаган бўлса, у ҳолда мақола муаллифини Komilov Islom деб дастурнинг ўзи киритсин.

```
<?php
function Add_maqola($nomi, $sizohi,
$muallifi="Komilov Islom"){
echo "Katalogga $nomi maqolasini qo`shamiz. ";
echo "<br>Muallif: $muallifi";
echo "<br>Qisqa izoh: ";
echo "$sizohi<br>";
}
```

```
Add_maqola("Informatika asoslari",
"Bu maqola informatika bo`yicha ...",
"Yuldashev Umid");
Add_maqola("Ma`lumotlar bazasi",
"Bu maqola ma`lumotlar bazasi haqida ...");
```

```
?>
```

Natija:Katalogga Informatika asoslari maqolasini qo`shamiz.

Muallif: Yuldashev Umid

Qisqa izoh: Bu maqola informatika bo`yicha ...

Katalogga Ma`lumotlar bazasi maqolasini qo`shamiz.

Muallif: Komilov Islom

Qisqa izoh: Bu maqola ma`lumotlar bazasi haqida ...

Агар биз қуйидагича ёзсак:

```
<?php
function Add_maqola($muallifi="Komilov Islom", $nomi, $sizohi)
{
echo "Katalogga $nomi maqolasini qo`shamiz. ";
echo "<br>Muallif: $muallifi";
echo "<br>Qisqa izoh: ";
echo "$sizohi<br>";
}
Add_maqola("Ma`lumotlar bazasi", "Bu maqola ma`lumotlar bazasi haqida ...");
?>
```

Natija:

Warning: Missing argument 3 for Add_maqola(), called in
D:\xampplite\htdocs\learn_php\learn.php on line 9 and defined in

D:\xampplite\htdocs\learn_php\learn.php on line 2

Katalogga Bu maqola ma`lumotlar bazasi haqida ... maqolasini qo`shamiz.

Muallif: Ma`lumotlar bazasi

Qisqa izoh:

Ўзгарувчи параметрлар руйхати

PHP4 да функцияга нечта *параметр* бўйича мурожаат этишни олдиндан билмасдан туриб функцияни яратиш мумкин. Бундай функцияни ёзиш учун ҳеч қандай махсус синтаксислар талаб этилмайди. Ҳаммаси **PHP** нинг **ички функциялари** ёрдамида бажарилади `func_num_args()`, `func_get_arg()`, `func_get_args()`.

`func_num_args()` функцияси **фойдаланувчи томонидан аниқланган функцияда** бериладиган параметрлар сонини қайтаради. Бу функция фақат **фойдаланувчи томонидан аниқланган функция** ичида қўлланилиши мумкин. Агар `func_num_args()` функцияси **фойдаланувчи томонидан аниқланган функциянинг** ташқарисида қўлланилса унда интерпретатор бу ҳақида хабар беради.

8.6 - мисол. `func_num_args()` функциясининг ишлатилиши

```
<?php
function DataCheck(){
    $n = func_num_args();
    echo "Funksiya parametrlari soni $n ta<br>";
}
DataCheck();
// «Funksiya parametrlari soni 0 ta » qatorini chiqaradi
DataCheck(1,2,3,6);
// «Funksiya parametrlari soni 3 ta » qatorini chiqaradi
?>
```

Natija:
Funksiya parametrlari soni 0 ta
Funksiya parametrlari soni 4 ta

`func_get_arg()` (бутун *параметр_номери*) функцияси **фойдаланувчи томонидан аниқланган функция** параметрлари руйхатидан рақамлар кетма - кетлиги бўйича *параметр_номери* га мос берилган параметрни қайтаради. **Фойдаланувчи томонидан аниқланган функция** параметри нолдан бошлаб саналади. `func_num_args()` дагидек бу функция ҳам фақат **фойдаланувчи томонидан аниқланган функция** ичида қўлланилиши мумкин.

параметр_номери **фойдаланувчи томонидан аниқланган функцияда** берилган параметрлар сонидан катта бўлиши мумкин эмас. Акс ҳолда `func_get_arg()` функцияси false ни қайтаради ва мос хабарни ишлаб чиқади.

8.7 - мисол. Параметрларда берилган маълумотлар типини текшириш учун функция яратайлик.

```
<?
Function DataCheck(){
    $check =true;
    $n = func_num_args();
    // funksiyaga berilgan parametrlar soni
    /* birinchi berilgan parametr butun son ekanligini tekshiramiz */
    if ($n>=1)
    if (!is_int(func_get_arg(0))) $check = false;
    /* ikkinchi berilgan parametr satr ekanligini tekshiramiz*/
    if ($n>=2)
    if (!is_string(func_get_arg(1))) $check = false;
    return $check;
}
if (DataCheck(a123,"text"))
    echo "Berilganlar shartlarni qoniqtiradi<br>";
```

```

else echo "Berilganlar shartlarni qoniqtirmaydi<br>";
if (DataCheck(324))
    echo " Berilganlar shartlarni qoniqtiradi<br>";
else echo "Berilganlar shartlarni qoniqtirmaydi";
?>

```

Natija:

Berilganlar shartlarni qoniqtirmaydi

Berilganlar shartlarni qoniqtiradi

func_get_args() функцияси **фойдаланувчи томонидан аниқланган функцияда** берилган параметрлар руйхатидан тузилган массивни қайтаради. Массивнинг ҳар бир элементи функцияда берилган параметрга мос тушади. Агар функция **фойдаланувчи томонидан аниқланган функцияда** қўлланилмаётган бўлса, у ҳолда хабар ишлаб чиқилади.

8.7 - мисолга ушбу функцияни қўллаб ўзгартириш киритамиз. Функцияга берилаётган ҳар бир тоқ параметр бутун сон эканлигини текширамыз.

```

<?
Function DataCheck(){
    $check =true;
    $n = func_num_args();
    // funksiyaga berilgan parametrlar soni
    $args = func_get_args();
    // funksiya parametri massivi
    for ($i=1;$i<=$n;$i++){
        $v = $args[$i];
        if ($i % 2 == 0){
            if (!is_int($v)) $check = false;
            // juft parametr butun sonligini tekshiramiz
        }
    }
    return $check;
}
if (DataCheck("text", 324))
    echo "Berilgan juft parametr juft son<br>";
else echo "Berilganlar shartlarni qoniqtirmaydi";
?>

```

Natija:Berilganlar shartlarni qoniqtirmaydi

Кўриб турганимиздек **func_num_args()**, **func_get_arg()**, **func_get_args()** функциялар комбинацияси **фойдаланувчи томонидан аниқланган функциялар** ўзгарувчи параметрлар рўйхатига эга бўлиши учун қўлланилади. Бу функциялар фақат PHP 4 да берилган. PHP 3 да шунга ўхшаш натижага эришиш учун массив функцияси қўлланилади.

8.8 - мисол. Функциядаги ҳар бир жуфт параметр бутун сон эканлигини текшириш учун скриптни қўйидагича ёзиш мумкин:

```

<?
Function DataCheck($params){
    $check =true;
    $n = count($params);
    // funksiyaga berilgan parametrlar soni
    for ($i=0;$i<$n;$i++){
        $v = $params[$i];
        if ($i % 2 != 0){
            // juft parametr butun son ekanligini tekshiramiz
            if (!is_int($v)) $check = false;
        }
    }
}

```

```
}  
return $check;  
}  
if (DataCheck("text", 324))  
    echo "Berilgan har bir juft parametr butun son <br>";  
else echo "berilganlar shartlarni qoniqtirmaydi";  
?>
```

Natija: Berilgan har bir juft parametr butun son

8.5. Ҳаракатлар майдони

Олдинги мавзуларда ўзгарувчиларни бирор бир файл ичида эълон қилиб, саҳифага require() ёки include() операторлари ёрдамида юклаб олиш ҳақида гапириб ўтган эдик, функцияда эса ўзгарувчиларни функцияни ўзида эълон қилинади.

Ўзгарувчиларнинг ҳаракат диапазони ўзгарувчиларни кўринарли ва белгиланганлигига қараб бошқаради. Турли дастурлаш тилларида ўзгарувчиларни ҳаракат диапазони ўрнатилган турлича қоидаларга бўйсунди.

RНР да эса қуйидагича қоида мавжуд:

- функция ичида аниқланган ўзгарувчилар улар эълон қилинган оператордан бошлаб функцияни якунловчи қавсгача бўлган майдонда ҳаракатланади. Бу майдон функция майдони, ўзгарувчилар эса – **локал ўзгарувчилар** деб аталади.

- функция ташқарисида аниқланган ўзгарувчилар улар эълон қилинган оператордан бошлаб, функция ичида эмас, балки файл охиригача бўлган майдонда ҳаракатланади. Бу майдон глобал майдон, ўзгарувчилар эса – **глобал ўзгарувчилар** дейилади.

- require() ва include() операторлари ёрдамида юклаб олинган файлларда эълон қилинган ўзгарувчилар функция ҳаракат майдонида қабул қилинмайди. Агар операторлар функция ичида ишлатилса, функция майдонида қабул қилинади. Агар операторлар функция ташқарисида ишлатилса, глобал майдонда қабул қилинади.

- функция ичида аниқланган ёки ишлатилаётган ўзгарувчиларни глобал майдонда ҳаракатлана олиши учун global калит сўзи ёрдамида белгилаш(кўрсатиш) мумкин, яъни ўзгарувчиларни функция ичида қўллаш учун бу ўзгарувчилар глобал ўзгарувчи сифатида эълон қилиниши керак. Бунинг учун функция танасида global калит сўзидан кейин уларнинг номлари келтирилиши керак.

global \$var1, \$var2, \$var3, ...,

- ўзгарувчилар unset(\$variable_name) функцияси ёрдамида ўчирилади. Агар ўзгарувчи ўчирилган бўлса, ҳаракат майдонида бошқа ишлатилмайди.

8.6. Функция ичида ўзгарувчиларни ишлатилиши

fn() функцияси ичида \$var ўзгарувчиси эълон қилинди. Бу ўзгарувчи функция ичида эълон қилингани учун эълон қилинган жойдан бошлаб функция якунигача функция ҳаракатланиш майдонида ишлай олади. Функциядан ташқарида эса \$var ўзгарувчисига мурожаат этилди, яъни янги \$var ўзгарувчиси яратилди. Бу янги ўзгарувчи глобал ҳаракатланиш майдонида файл охиригача ишлай олади. Агар бу ўзгарувчини echo оператори ёрдамида экранга чиқазсак, ҳеч қандай қиймат чиқмайди. Чунки бу ўзгарувчи функция ташқарисида қийматга эга эмас.

```
<?
Function fn()
{
$var="mazmun";
}
Echo $var;
?>
```

Қуйидаги мисол бунга тескари. Ўзгарувчини функция ташқарисида эълон қиламиз, сўнгра функция ичида уни ишлатамиз.

8.9 - мисол. Ўзгарувчиларни функция ичида ва ташқарисида ишлатилиши

```
<?
Function fn()
{
Echo "funksiya ichida, \ $var=". $var. "<br>";
$var="mazmun2";
Echo "funksiya ichida, \ $var=". $var. "<br>";
}
$var="mazmun1";
Fn();
Echo "funksiya tashqarisida, \ $var=". $var. "<br>";
?>
```

Natija:
funksiya ichida, \$var=
funksiya ichida, \$var=mazmun2
funksiya tashqarisida, \$var=mazmun1

функция бу ҳолда ҳам бажарилмади, чунки ўзгарувчи функция ичига чақирилмади, биринчи бажарилиши лозим бўлган оператор \$var="mazmun1";

бу ерда глобал майдонда ҳаракатлана оладиган ва қиймати mazmun1 га тенг \$var ўзгарувчиси яратилади.

Кейинги бажарилаётган оператор – функцияга мурожаат этиш. Функция ичидаги сатрлар кетма-кет бажарилади. Функцияни биринчи сатри \$var ўзгарувчисига мурожаат этади. Бу сатр бажарилганда яратилган \$var ўзгарувчисини кўра олмайди, шунинг учун функция майдонида янги ўзгарувчи яратади ва уни экранга чиқазади. Натижада 1 сатр экранга чиқади.

Функция ичидаги кейинги сатр \$var ўзгарувчисини «mazmun2» га тенглаштиради. Ҳаракат функция ичида бўлаётганлиги учун бу сатр глобал эмас, балки локал ўзгарувчи \$var қийматини ўзгартиради. Иккинчи сатр эса бу ўзгаришни экранга чиқазади.

Шундан сўнг функция ишини тугатади ва саҳифани охириги сатри бажарилади. Echo оператори ёрдамида глобал ўзгарувчини ўзгармаганлигини кўриш мумкин.

Глобал ўзгарувчилар

Агар функция ичида яратилган ўзгарувчи глобал бўлиши керак бўлса, унинг олдида **global** калит сўзи ишлатилади.

8.10 - мисол. Глобал ўзгарувчиларни ишлатилиши

```
<?
Function fn()
{
global $var;
$var="contents";
Echo "funksiya ichida, \$var=".$var."<br>";
}
Fn();
Echo "funksiya tashqarisida, \$var=".$var."<br>";
?>
```

Natija:
funksiya ichida, \$var=contents
funksiya tashqarisida, \$var=contents

Бу мисолда, \$var глобал ўзгарувчи сифатида эълон қилинган бўлиб, ўзгарувчи майдонда global \$var сатри бажарилган жойдан бошлаб аниқланган. Шунинг учун ўзгарувчи функция ташқарисида ҳам ишлатилиши мумкин.

Функцияни унга мурожаат этилаётган жойдан юқорида ёки пастда ҳам эълон қилиш мумкин. (Эътибор беринг, функция ҳаракатланиш майдони ўзгарувчи ҳаракатланиш майдонидан тубдан фарқ қилади!)

Статик ўзгарувчилар

Ўзгарувчиларни функция ичида, функциядан чикқандан кейин ҳам улар қийматини сақлаб қолинадиган қилиб қўллаш учун уларни статик ўзгарувчилар сифатида эълон қилиш керак. Агар дастур бажарилиши функция қиймати чегараларидан чикса, у ҳолда статик ўзгарувчилар фақат функция ичида кўринади. Бундай ўзгарувчиларни эълон қилиш учун static калит сўзи қўлланилади:

```
static $var1, $var2;
```

Статик ўзгарувчига исталган қиймат бириктирилиши мумкин, лекин жўнатмани унга бириктириб бўлмайди.

8.11 - мисол. Статик ўзгарувчиларни ишлатилиши

```
<?
function Test_s(){
static $a = 1;
// Ifoda va jo`natmani biriktirish mumkin emas
$a = $a*2;
echo $a;
}
Test_s(); // 2 ni chiqaradi
echo $a; // $a faqatgina funksiya ichida qiymatga ega bo`lganligi uchun
// hech narsani chiqarmaydi
Test_s(); // funksiya ichida $a=2, shuning uchun
// funksiya ishi natijasi 4 soni bo`ladi
?>
```

Natija:24

8.7. Қайтарилувчи мазмунлар

Юқорида мисол тариқасида келтирилган функцияларнинг ҳаммаси қандайдир ҳаракатларни бажаради ва натижани экранга чиқазади. Баъзи ҳолларда исталган функция ишининг натижаси сифатида қандайдир қийматни экранга чиқазиш эмас, балки фақатгина натижани қайтариш керак бўлади. Бу **return** буйруғи ёрдамида амалга оширилади. Қайтариладиган қиймат ҳар-хил тоифада бўлиши мумкин, шу қаторда рўйхат ҳам, объект ҳам. Интерпретатор функция танасида **return** буйруғини кўргач, уни бажарилишини дарров тўхтади ва қайси қаторда функцияга мурожаат этилган бўлса ўша қаторга ўтади.

8.12 – мисол. Қайтарилувчи мазмунлар

```
<?
function kvadrat($a)
{
    $nat=$a*$a;
    return $nat;
}
$y=kvadrat($a);
$c=$y/10+25;
echo $c;
?>
Natija:25
```

8.8. Қайтариловчи жўнатмалар

Ўз ишининг натижаси сифатида функция қандайдир ўзгарувчи(глобал, статик ёки параметрлардан бирига)га жўнатмани ҳам қайтариши мумкин. Қайтариловчи жўнатмалар функцияни қайси ўзгарувчига жўнатма бириктирилганини аниқлаш керак бўлган ҳолларда жуда ҳам қўл келади. Функциядан жўнатмани олиш учун уни эълон қилинишида ва ҳар сафар функцияга мурожаат этилаётганда номи олдида амперсанд белгиси (&) ни қўйиш керак. Одатда функция жўнатма билан берилган бўлса, жўнатмани қандайдир глобал ўзгарувчига (ёки унинг қисми – жўнатмани глобал массив элементига), статик ўзгарувчига (ёки унинг қисмига) ёки параметрлардан бирига қайтаради.

8.13 - мисол. Қайтариловчи жўнатмалар

```
<?
$a = 'juft son';
$b = 'toq son';
function &ref($par){
global $a, $b;
if ($par % 2 == 0) return $a;
    else return $b;
}
$c=4;
$var =&ref($c);
echo $c. " - ". $var. "<br>";
// 4 - juft son chiqaradi
$c = 11;
$var =&ref($c);
echo $c. " - ". $var. "<br>";
// 11 - toq son chiqaradi
?>
```

Natija:
4 - juft son
11 - toq son

8.9. Ўзгарувчи функциялар

PHP ўзгарувчи функциялар концепциясини қўллайдди. Бу, агар ўзгарувчи кавслар билан тугаган бўлса, у ҳолда PHP худди шундай номли функцияни қидиришини ва уни бажаришга ҳаракат қилишини билдиради.

8.14 – мисол. Ўзгарувчи функцияларнинг ишлатилиши

```
<?
/* ikki oddiy funksiyani yaratamiz:
Add_sign - qatorga yozuv qo'shadi va
Show_matn - matn qatorini chiqaradi */
function Add_sign($string, $sign="Hurmat bilan Komil")
{
    echo $string ." ".$sign;
}
function Show_matn()
{
    echo "Xabarni pochta orqali jo`natish<br>";
}
$func = "Show_matn";
// Show_matn funksiyasi nomiga teng qiymatli o`zgaruvchini yaratamiz
$func();
// Bu Show_matn funksiyasiga murojaat etadi
$func = "Add_sign";
// Add_sign funksiyasi nomiga teng qiymatli o`zgaruvchini yaratamiz
$func("Hammaga salom <br>");
// Bu "Hammaga salom" parametrli
// Add_sign funksiyasiga murojaat etadi
?>
```

Natija:
Xabarni pochta orqali jo`natish
Hammaga salom
Hurmat bilan Komil

Бу мисолда Show_matn функцияси фақат сатрни чиқариш учун хизмат қилади. Махсус echo() функцияси мавжудлиги туфайли нима учун алоҳида функция яратдик деган савол туғилади. Гап шундаки echo(), print(), unset(), include() каби функцияларни ўзгарувчи сифатида қўллаб бўлмайди. Шунинг учун биз

echo "ТЕХТ" ; ни қуйидагидек ёзсак, у ҳолда интерпретатор қуйидаги хатони чиқазади

```
<?
$func = "echo";
$func("ТЕХТ");
?>
```

Natija:
Fatal error: Call to undefined function echo() in D:\xampplite\htdocs\learn_php\learn.php on line 3

Шунинг учун юқоридаги санаб ўтилган функцияларни ўзгарувчи сифатида қўллаш учун юқоридаги мисолдагидек янги функция яратиб олиш керак.

8.10. Ички функциялар

RНР – Web саҳифаларни яратишда ишлатиладиган кенгайтирилган функциялар кутубхонасига эга технология. RНР нинг 100 дан ортиқ ички (стандарт) функциялари мавжуд. Бу функцияларни кўпчилиги қандайдир маълумотларни қабул қилиб ва натижа сифатида бирор-бир маълумот чиқазади.

Функцияни эълон қилишни оддий формаси сифатида RНР нинг ички функцияларидан бири ***date()*** *функцияси*ни кўрамиз.

Синтаксиси:

date(параметр);

Эътибор қаратинг, функцияга берилаётган маълумот қавс ичига олиб ёзилмоқда. Бу қиймат функция *аргументи* ёки *параметри* деб аталади. *Параметрлар* кирувчи қийматлар ҳисобланиб, функция экранга *параметрларга* боғлиқ аниқ бир натижани чиқазади.

Бу ерда *параметр* ўрнида қуйидагилар кўрсатилиши мумкин:

- a — "am" ёки "pm"
- A — "AM" ёки "PM"
- d — ой куни, рақамли, 2 та рақамдан (биринчи ўринда ноль туради)
- D — ҳафта куни, матнли, 3 ҳарфдан иборат; "Fri" ва ҳ.
- F — ой, матнли, тўлиқ; "January" ва ҳ.
- h — соат, рақамли, 12 соатли форматда
- H — соат, рақамли, 24 соатли форматда
- i — минут, рақамли
- j — ой куни, рақамли, бошида келувчи нолсиз
- l ('L' катта ҳарфли) — ҳафта куни, матнли, тўлиқ; "Friday" ва ҳ.
- m — ой, рақамли
- M — ой, матнли, 3 ҳарфдан иборат; "Jan" ва ҳ.
- s — секунд, рақамли
- S — инглизча тартибдаги суффикс, матнли, 2 белгидан иборат; "th", "nd" ва ҳ.
- U — секундда, аср бошидан бошлаб
- Y — йил, рақамли, 4 та рақамдан иборат
- w — ҳафта куни, рақамли, 0 якшанбани билдиради
- y — йил, рақамли, 2 та рақамдан иборат
- z — йил куни, рақамли; "299" ва ҳ.

8.15 - мисол. **date()** функциясининг ишлатилиши

```
<?
$d=date(d);
$m=date(m);
$y=date(y);
Echo "Bugun: $d.$m.$y";
?>
Natija:Bugun: 27.01.11
```

Isset() *функцияси* ўзгарувчи мавжуд ёки мавжуд эмаслигини текширади, агар ўзгарувчи мавжуд бўлса, функция true(1) қийматини, акс ҳолда false(0) ни қайтаради.

Синтаксиси:*isset*(ўзгарувчи);**8.16 - мисол. Isset()** функциясининг ишлатилиши

```
<?
$c=34;
If(isset($c))
{
Echo "O`zgaruvchi mavjud va uning qiymati $c ga teng";
}
Else
{ Echo "O`zgaruvchi mavjud emas";
}
?>
```

Natija: O`zgaruvchi mavjud va uning qiymati 34 ga teng

Unset() функцияси ўзгарувчиларни ўчириш учун ишлатилади.**Синтаксиси:***unset*(ўзгарувчи);**8.17 - мисол. Unset()** функциясининг ишлатилиши

```
<?
$k=34;
Echo "funksiyaga murojaat etishdan avval k=". $k."<br>";
unset($k);
Echo "funksiyaga murojaat etgandan so`ng k=". $k."<br>";
?>
```

Natija:
funksiyaga murojaat etishdan avval k=34
funksiyaga murojaat etgandan so`ng k=

mail() функцияси хабарни электрон почта орқали берилган манзилга жўнатади.**Синтаксиси:***boolean mail* (*string* олувчи, *string* мавзу, *string* хабар [, *string* қўшимча_сарлавҳа])

«мавзу» параметрида, хабар мавзуси берилади. «хабар» параметри эса хабар матнидан таркиб топади, шарт бўлмаган параметр - «қўшимча_сарлавҳа» да хабар билан жўнатиладиган қўшимча маълумот белгиланади (масалан, HTML форматлаш атрибутлари). **Мисол учун,**

```
<?
$email = "yourname@yourserver.com";
$mavzu = "Bu mavzu";
$xabar = "Bu xabar";
$sarlavha = "From: somebody@somesite.com";
mail ($email, $mavzu, $xabar, $sarlavha);
?>
```

Биз PHP нинг ички функцияларини баъзилари билан, яъни echo(), print(), include(), require(), date(), isset(), unset(), mail() кабилар билан танишдик. Аслида юқорида санаб ўтилган функцияларнинг ичидан date() дан ташқари ҳаммаси тил конструкциялари ҳисобланади. Улар PHPнинг ядроси ҳисобланади ва ҳеч қандай солаш ва модулларни талаб этмайди. Лекин шундай функциялар борки улар билан ишлаш учун ҳар-хил кутубхоналарни ўрнатиш ва мос модулни улаш керак. Масалан, маълумотлар базаси билан

ишловчи MySQL функцияси билан ишлаш учун PHP ни шу кенгайтмада комплекслаш керак. Охирги вақтларда тарқалган кенгайтмалар ва мос равишда уларнинг функциялари улар билан ишлашда интерпретаторнинг ҳеч қандай қўшимча созлашларисиз ишлашда PHPнинг бошланғич таркибига киритилган.

Шунга ўхшаш PHP нинг массивлар билан, файллар билан, маълумотлар базаси билан ишловчи ички функциялари мавжуд. Биз бу функциялар билан кейинги бўлимларда танишамиз.

8.1 - Масала

html- формасини яратиш имконини берувчи интерфейс ёзамиз. Бу интерфейс фойдаланувчи кўрсатган элементлар ва уларни нечта сонда яратишни, уларга ном беришни танлаганда керакли формани ўзи қайта ишлаши керак.

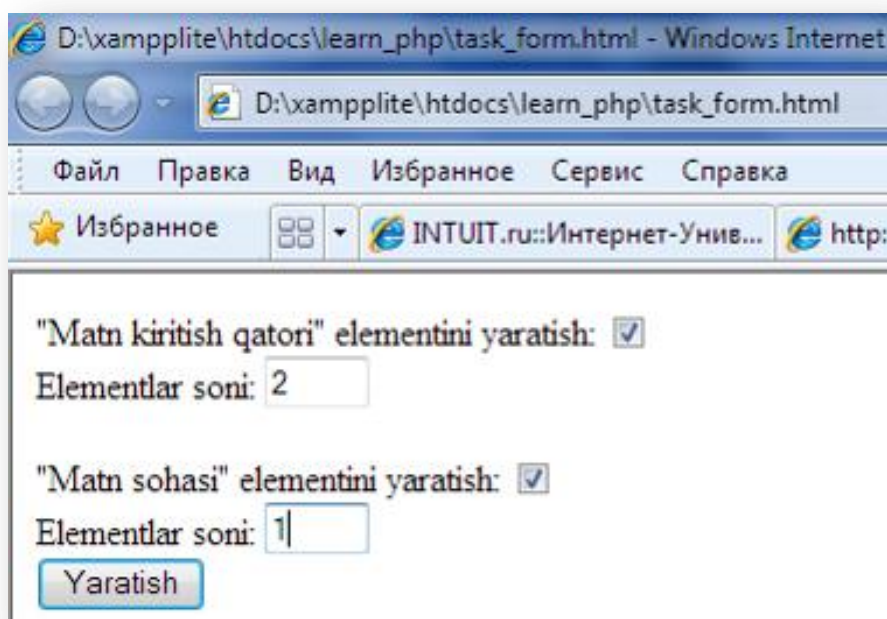
Масалани бир нечта қисмга бўламиз: 1. Яратиладиган элементларни танлаш ва уларнинг сонини киритиш; 2. Яратилган элементлар номини киритиш ва олинган маълумотларни қайта ишлаш, ҳамда 3. Формани узлуксиз қайта ишлаш. Биринчи масала оддий: яъни Яратиладиган элементларни танлаш ва уларнинг сонини киритиш учун формани ёзамиз.

8.18 – листинг. task_form.html

```
<form action="ask names.php">
"Matn kiritish qatori" elementini yaratish: <input
type=checkbox name=types[]
value=string><br>
Elementlar soni: <input type=text
name=numbers[string]
size=3><br>
<br>
"Matn sohasi" elementini yaratish: <input
type=checkbox
name=types[] value=text><br>
Elementlar soni: <input type=text
name=numbers[text]
size=3><br>
<input type=submit value="Yaratish">
</form>
```

Биз элемент формаси номига масалан types[] деб ёзадиган бўлсак, бу унинг номи – массивнинг кейинги элементи types эканлигини билдиради. Яъни бизда биринчи элемент номи types[0] (Matn kiritish satri), иккинчиси types[1](Matn sohasi) bo`ladi.

task.form.html браузерда қуйидагича кўринишга эга бўлади(8.1 - расм):



8.1 - расм. Яратиладиган элементлар ва уларнинг сонини кiritиш учун форма

Бундай форма маълумоти жўнатилгандан кейин, қандай элементлар ва уларнинг ҳар биридан нечта яратиш кераклиги ҳақида маълумотга эга бўлиш мумкин. Кейинги скриптда бу элементлар учун ном берилади:

8.19 – листинг. ask_names.php

```
<?
$file = "task.php";
/* shu script bilan formani qayta ishlovchi fayl*/

function Ask_names(){
global $file;
//bu o'zgaruvchini berilgan funksiya tashqarisida
//qo'llashimiz uchun global sifatida e'lon qilamiz
if (isset($_GET["types"])){
    $st = '<form action="'. $file. '>';
    foreach ($_GET["types"] as $k => $type){
/* yaratilishi kerak bo'lgan hamma
elementlar tipini olamiz */
        $num = $_GET["numbers"][$type];
        // har bir tipdan nechta element kerak
        for ($i=1;$i<=$num;$i++){
            $st.= "$type tipidagi $i-element nomini kiriting: ";
            $st.= "<input type=text name=names[$type][]><br>";
        }
        $st.= "<input type=hidden name=types[] value=$type>";
        $st.= "<input type=hidden name=numbers[] value=$num><br>";
    }
    $st .= "<input type=submit name=send value=Jo'natish ></form>";
return $st;
// $st o'zgaruvchisida so'rov nomi
// uchun forma kodi saqlanadi
} else echo "Select type";
}
echo Ask_names();
// funksiyanı chaqiramiz
// va ishning natijasini chiqaramiz
```

>

8.1 -расмдаги формада фойдаланувчи матн киритиш қатори типига 2 та элемент ва матн соҳаси типига 1 та элемент яратиш кераклигини кўрсатган бўлсин. У ҳолда ask_names.php скрипти уни қайта ишлагач, биз қуйидагига эга бўламиз(8.2 - расм):

8.2 - расм. Яратилган элементлар номини киритиш учун форма

Бу формада қаторларга “Nomi”, “Muallifi” ва “Qisqacha mazmuni” каби номларни киритайлик. Бу маълумотларни task.php скрипти қайта ишлайди.

8.20 – листинг. task.php

```
<?
$show_file = "task_show.php";
/* Ushbu fayl formasi yaratgan ma`lumotlarni
   qayta ishlovchi fayl */
function Create_element($type,$name){
    // funksiya kiritilga tipi va nomiga ko`ra
    // element yaratadi
    switch($type){
        case"string":
            $str .= "$name: <input type=text name=string[]><br>";
            break;
        case "text":
            $str .= "$name: <textarea name=text[]></textarea><br>";
            break;
    }
    return $str;
}
function Create_form(){
    // funksiya kerakli elementlar
    // bilan forma yaratadi
    global $show_file;
    $str = '<form action="'. $show_file. '>';
    foreach ($_GET["types"] as $k => $type){
        // elementlar tipini qayta tanlaymiz
        $num = $_GET["numbers"][$k];
        // bu elementlar tipining soni
        for ($i=1;$i<=$num;$i++){
            $arr = $_GET["names"][$type][$i-1];
            // yaratilayotgan elementning nomi
            $str .= Create_element($type,$arr);
            // elementni yaratish uchun funksiyani chaqiramiz
        }
    }
}
```

```

}
$str .= "<input type=submit value=send></form>";
echo $str;
}
$crt = "Create_form";
$crt(); // Create form formasini yaratish
// funksiyasini chaqiramiz
?>

```

Юқоридаги скрипт натижасида куйидаги кўринишдаги форма ҳосил бўлади(8.3 - расм):

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/learn_php/task.php?names%5Bstring%5D%5B%5D=Nomi&names%5Bstring%5D%5B%5D=Nomi`. The browser's menu bar includes 'Файл', 'Правка', 'Вид', 'Избранное', 'Сервис', and 'Справка'. The address bar shows the current page is 'Избранное' (Favorites). The main content area displays a form with three input fields: 'Nomi:', 'Muallifi:', and 'Qisqacha mazmuni:'. Below the 'Qisqacha mazmuni:' field is a 'send' button.

8.3 - расм. Дастурни қайта ишловчи форма

IX БОБ. ФАЙЛЛАР БИЛАН ИШЛАШ

Бу бобда файллар яратиш, файлда маълумотларни ўқиш, файлларни ўчириш, сервердаги файлларни мавжудлигини текшириш каби саволларга жавоб берилади. (fopen, fwrite, fclose, file, fget, unlink, file_exists функциялари)

9.1. Маълумотларни сақлаш

Маълумотларни сақлашнинг иккита асосий усули мавжуд: икки ўлчамли файлда ва маълумотлар базасида.

Икки ўлчамли файл турли форматларда бўлиши мумкин, бироқ икки ўлчамли (flat) файл деганда оддий матнли файлни тушунамиз. Қуйида биз талабалар ҳақидаги маълумотларни матнли файлга ёзишни кўриб ўтамиз. Бунда ҳар бир талабанинг маълумотлари битта сатрга ёзилади.

Бу усул оддий ва чегараланган. Агар катта ҳажмдаги маълумотлар билан ишлашга тўғри келганда, маълумотлар базаси билан ишлаган маъқул. Бироқ, икки ўлчамли файл кенг кўламда қўлланилади, шунинг учун файллар билан ишлаш технологияларини ўрганиш зарур.

Маълумотларни файлга ёзиш уч босқичда амалга оширилади:

1. Файлни очиш. Агар файл мавжуд бўлмаса, уни яратиш талаб этилади.
2. Маълумотларни файлга ёзиш.
3. Файлни ёпиш.

Худди шунга ўхшаш, маълумотларни файлда ўқиш ҳам шу уч босқичга боғлиқ ҳолда бажарилади:

1. Файлни очиш. Агар файлни очиш мумкин бўлмаса (масалан, файл мавжуд бўлмаса), бу ҳолат текширилади ва бу ҳолатдан тўғри чиқиб кетиш йўли қаралади.
2. Файлдан маълумотларни ўқиш.
3. Файлни ёпиш.

Зарур бўлганда, файлда маълумотларни бир ўқишда файлнинг қанча қисмини ўқиш кераклигини танлаш мумкин. Кейинроқ барча мавжуд имкониятларини кўриб ўтамиз.

9.2. Файлни очиш ва файлга боғланиш

PHP да файлни очиш учун `fopen()` функцияси ишлатилади. Файлни очишда файл устида қандай амал олиб бориш мўлжалланаётганлиги кўрсатилади. Бу рухсат типи дейилади.

Сервер операцион тизими очилган файл билан қандай амал бажариш кераклигини билиши керак. Ундан қуйидагиларни билиши талаб этилади: балки файл очик бўлиши ва айна вақтда агар саҳифа эгасининг ушбу саҳифа билан ишлашга ҳуқуқи етарли бўлса файл очик ва бошқа бир саҳифа билан қайта ишланаётган бўлиши мумкин. Мантиқан олганда, рухсат типлари операцион тизимга бошқа фойдаланувчилар ёки сценарийлардан тушаётган сўровларни қайта ишлаш усулини аниқлаш механизмини, шунингдек бирор файл билан ишлаш учун рухсат ва ҳуқуқларнинг мавжудлигини текшириш имкониятини беради.

Файлни очиш қуйидаги учта қоидага асосланади:

1. Файлни фақатгина ўқиш учун, фақатгина ёзиш учун ёки ўқиш ва ёзиш учун очилади.
2. Файлга ёзиш вақтида ихтиёрий файл таркиби қайта ёзилади ёки янги маълумот файлнинг охирига ёзилади.
3. Тизимдаги файлга ёзиш вақтида иккилик ва матнли файллар бир-биридан фарқланади шунинг учун файл типи кўрсатилиши талаб этилади.

Умуман олганда, PHP нинг файллар яратиш учун белгиланган функцияси мавжуд эмас. Кўпгина функциялари сервердаги файллар системасидаги мавжуд файллар билан ишлайди. Лекин, вақтинчалик файлларни яратиш учун бир нечта функциялари мавжуд.

Энг оддий файлни яратиш учун локал ва удалённый файлларни очувчи функциядан фойдаланилади. Бу функция `fopen()` деб номланади. `fopen()` берилган файлни дастурни бошқарувчи поток билан боғлайди.

***fopen()** функцияси*

Синтаксиси:

resource **fopen** (*файл_номи, рухсат_типи*
[, *use_include_path*(*йўлни_қўйиши*)])

Бу функция натижаси сифатида очилган файлга кўрсаткични (ресурс типи) қайтаради. Функция параметрлари сифатида очилиши керак бўлган *файл_номи*, файлга *рухсат_типи*(бу ерда файл билан қандай амал бажарилмоқчилиги кўрсатилади) ва *include_path* (кўрсатилган файлни қаердан қидиришни кўрсатилади) берилади. Бу учта параметр ҳақида батафсилроқ маълумот берамиз.

файл_номи параметри файлнинг тўғри локал номи ёки файлнинг тармоқдаги URL-манзилидан таркиб топган сатрдан иборат бўлади. Агар *файл_номи* кириш протоколини кўрсатиш билан бошланса(масалан, `http://...` Ёки `ftp://...`), у ҳолда интерпретатор буни URL манзили номи деб ҳисоблайди ва қайта ишловчини кўрсатилган URL протоколидан қидиради. Агар қайта ишловчи топилган бўлса, у ҳолда PHP худди оддий файл(`allow_url_fopen` директиваси) каби URL объекти билан ишлашга рухсат берилганлигини текширади. Агар `allow_url_fopen=off` бўлса, у ҳолда `fopen()` хатоликни

чақиради ва огоҳлантиришни чиқазади. Агар *файл_номи* протокол билан бошланмаса, у ҳолда *файл_номи* локал ном сифатида кўрсатилган бўлади. Локал файлни очиш учун, PHP бу файлга мос рухсат ҳукукига эга бўлиши керак.

файл_номи бир неча хил кўринишларда берилиши мумкин:

- агар параметр локал файл номидан иборат бўлса, **fopen()** функцияси бу файлни очиб, кўрсаткични қайтаради;

- агар параметр `php://stdin`, `php://stdout` ёки `php://stderr` кўринишида берилган бўлса, мос кириш/чиқиш стандарт потоклари очилади;

- агар параметр `http://` олд қўшимчаси билан бошланса, у ҳолда функция серверга HTTP боғланишни очади ва кўрсаткични кўрсатилган файлга қайтаради;

- агар параметр `ftp://` олд қўшимчаси билан бошланса, у ҳолда функция серверга FTP боғланишни очади ва кўрсаткични (манипуляторни) кўрсатилган файлга қайтаради. Бу ҳолатда асосий эътиборни ушбу иккита ҳолга қаратиш лозим: агар сервер FTP нинг пассив режимини қўлламаса, у ҳолда **fopen()** га мурожаат этиш муваффақиятсиз якунланади. Хусусан, FTP файллари ёки ўқиш учун ёки ёзиш учун очилади.

рухсат_типи параметри қуйидаги қийматлардан бирортасини қабул қилиши мумкин. (12.1 - жадвал)

9.1 - жадвал. Файлнинг рухсат типлари

Рухсат типи	Номланиши
r	Ўқиш режими - файлни фақат ўқиш учун очади; кўрсаткич файлга унинг бошидан бошлаб ўрнатилади.
r+	Ўқиш режими - файл ўқиш ва ёзиш учун очилади; кўрсаткич файлга унинг бошидан бошлаб ўрнатилади.
w	Ёзиш режими - файл фақат ёзиш учун очилади; кўрсаткич файлни бошидан бошлаб ўрнатилади. Агар файл мавжуд бўлса, файл нол узунликкача қисқартирилади, яъни таркибидагиларнинг барчаси ўчирилади. Агар файл мавжуд бўлмаса, уни очиш вақтида файлни яратади.
w+	Ёзиш режими - файл ўқиш ва ёзиш учун очилади; кўрсаткич файлни бошидан бошлаб ўрнатилади. Агар файл мавжуд бўлса, таркибидагиларнинг барчаси ўчирилади. Агар файл мавжуд бўлмаса, уни очишга ҳаракат қилади ва натижада файлни яратади.
a	Қўшиш режими - файл фақат қўшиш (ёзиш) учун очилади; кўрсаткич файл таркибини охирига ўрнатилади. Агар файл мавжуд бўлмаса, уни очишга ҳаракат қилади ва натижада файлни яратади.
a+	Қўшиш режими - файлни ёзиш ва ўқиш учун очади; кўрсаткич файл таркибини охирига ўрнатилади. Агар файл мавжуд бўлмаса, уни очишга ҳаракат қилади ва натижада файлни яратади.

x	Фақатгина ёзиш учун файл яратади ва очади; кўрсаткични файлни бошига ўрнатади. Агар файл мавжуд бўлса, у ҳолда <code>fopen()</code> <code>false</code> ни чиқазади ва огоҳлантириш беради. Агар файл мавжуд бўлмаса, у ҳолда уни яратишга уриниб кўради. Рухсатнинг бу типи РНР 4.3.2 версиясидан бошлаб қўлланилади ва фақатгина локал файллар билан ишлайди.
x+	Ёзиш ва ўқиш учун файл яратади ва очади; кўрсаткични файлни бошига ўрнатади. Агар файл мавжуд бўлса, у ҳолда <code>fopen()</code> функцияси <code>false</code> ни қайтаради ва огоҳлантириш беради. Агар файл мавжуд бўлмаса, у ҳолда уни яратишга уриниб кўради. Рухсатнинг бу типи РНР 4.3.2 версиясидан бошлаб қўлланилади ва фақатгина локал файллар билан ишлайди.

Қуйида файлни очиш функцияси `fopen()` га мисол келтирилган. Файл очилмаган ҳолатда, хатолик ҳақида хабар чиқазиш учун `die()` чакируви `fopen()` функцияси билан биргаликда ишлатилади.

```
<?
$file = "user.txt"; // бирорта файл
$fh = fopen($file, "a+") or die("File ($file) doesnot exist!");
?>
```

Қуйидаги фрагмент РНР сайтига уланишни очади (<http://www.php.net>):

```
<?
$site = "http://www.php.net";
// Server HTTP bog'lanishi barchaga ochiq
$sh = fopen($site, "r");
?>
```

`use_include_path` параметрида 1 ёки `TRUE` қийматлари ўрнатилган бўлса, интерпретатор `fopen()` да кўрсатилган файлни `include_path`дан қидиради. `include_path` – бу РНР ни сошлаш файлида файлларни ёқиш учун жойлаштирилган, директориялар рўйхатида берилган директива. `fopen()` функциясидан ташқари бу директива `include()` ва `require()` функцияларида ҳам ишлатилади.

9.1 – мисол. `fopen()` функциясининг қўлланилиши

```
<?php
$х = fopen("my_file.html", "w");
/* my_file.html fayliga ma'lumot yozish uchun
agar u mavjud bo'lsa, uni ochadi yoki
mavjud bo'lmasa shunday nomli
bo'sh faylni yaratadi */
$х = fopen("dir/another_file.txt", "w+");
/* dir katalogida yozish va o'qish uchun
another_file.txt faylini ochadi yoki yaratadi */
$х = fopen(
    "http://www.server.ru/dir/file.php", "r");
/* ko'rsatilgan manzildagi faylni o'qish uchun ochadi */
?>
```

Файл яратишдан олдин қайси операцион тизимда ишляпсиз ва бу файлни қайси операцион тизим(ОТ)да ўқиш мумкин, шуни билиш керак. Гап шундаки, турли операцион тизимларда сатр тугалланиши турлича белгиланади. Unix ОТида сатр тугалланиши `\n`, Windows типигаги ОТ ларда

эса сатр тугалланиши `\r\n` билан белгиланади. Сатрларни Unix типига тизимлардан Windows тизимига ўтказиш учун сатр тугалланиши белгисини махсус байроқ `t` сифатида белгилаш таклиф этилади. Бунинг аксини бажарувчи `b` байроғи ҳам мавжуд бўлиб, бинар файлларда тез-тез ишлатилади. Бу байроқларни *файл* учун танланган *рухсат тип*и белгиси охирида ёзиб ишлатиш мумкин. Масалан, ўқиш учун очиётган файлда, сатр охиридаги барча белгиларни `\r\n` га қайта кодлаштириш учун, `r` ўрнига `rt` ни ишлатиш керак. Агар бинар файлларни очишда `b` байроғи ишлатилмаса, у ҳолда мавжуд файлларни ўзгартириш билан боғлиқ хатолар юзага келади. Дастурни турли платформаларга кўчириш масалаларида, **fopen()** ёрдамида файлларни очишда доимо `b` байроғини ишлатиш тавсия этилади.

Агар **fopen()** ёрдамида файлни очиш ёки яратиш амалга ошмаса, у ҳолда РНР да хатолик юзага келади, `fopen` функцияси эса ўзининг иши натижаси сифатида `false` қийматни қайтаради. Бундай кўринишдаги хатоликни `@` белгиси ёрдамида яшириш мумкин.

Масалан, қуйидаги сатр, агар файлни очиш амалга ошмаса ҳам, хатолик ҳақидаги огоҳлантиришни чиқазмайди:

```
$h = @fopen("dir/another_file.txt","w+");
```

Бу ҳолда **fopen()** функцияси фақатгина бўш файлни яратади ва унга ёзиш учун рухсатни таъминлайди. Бу файлга маълумотларни қандай ёзиш мумкин? Мавжуд файлдан маълумотларни қандай ўқиш мумкин?

Бу саволларга жавоб беришдан аввал **fopen()** ёрдамида очилган боғланишни ёпишни кўриб ўтамиз. Файл билан ишлаб бўлгандан сўнг ҳар доим **fclose()** функцияси ёрдамида боғланишни ёпиш керак.

9.3. Файл билан боғланишни ёпиш

Файлдаги маълумотни ўқиш, ёзиш ёки бошқа бирор-бир зарур амалларни бажариш учун бу файллар билан **fopen()** ёрдамида боғланиш ўрнатилади. Бу амаллар бажариб бўлингандан сўнг, албатта боғланишни ёпиш зарур. Бунинг учун **fclose()** функцияси ишлатилади.

fclose() функцияси

Синтаксиси:

fclose (файлга кўрсаткич)

Бу функция натижаси сифатида боғланишни ёпиш муваффақиятли амалга оширилган бўлса, TRUE қийматини, акс ҳолда FALSE ни қайтаради. **fclose()** функцияси **fopen()** ёки **fsockopen()** функциялари ёрдамида очилган файлларни муваффақиятли ёпади. Бу функциянинг параметри сифатида **fopen()** функцияси ёрдамида очилган файл кўрсаткичини кўрсатиш керак.

9.2 - мисол. fclose() функциясининг ишлатилиши

```
<?php
$h = fopen("my_file.html", "w");
fclose($h);
?>
```

Агар файл билан боғланиш ёпилмаса, скрипти бажаришда ҳеч қандай хатолик юзага келмайди. Лекин бу серверда жиддий оқибатларга олиб келиши мумкин. Масалан, хаккер очиқ боғланишдан фойдаланган ҳолда файлга вирус ёзиши мумкин, серверда бу ортиқча йўқотишни юзага келтиради. Шунинг учун файл билан зарур амалларни бажаргандан сўнг боғланишни ёпишни маслаҳат берамиз.

Очиқ файллар билан иккита асосий амал бажариш мумкин – ўқиш ва ёзиш. Ундан аввал эса файл мавжудлигини текшириб олиш зарур.

9.4 Файл мавжудлигини текшириш

Биз амаллар бажармоқчи бўлган файллар мавжуд бўлмаса, ёки ўқиш ва ёзиш учун рухсат берилмаган бўлса, у ҳолда амаллар бажарувчи функцияларни ҳеч қайси бири ишлаймади ва PHP хатолик ҳақида хабар чиқазади. Шунинг учун файл мавжудлигини қуйидаги функциялар ёрдамида текшириб олиш керак: `file_exists()`, `is_writable()`, `is_readable()`.

Файл мавжудлигини текшириш: `file_exists()` функцияси

Синтаксиси:

`bool file_exists (файл ёки каталог номи)`

`file_exists()` функцияси аргумент сифатида берилган номдаги файл ёки каталог мавжудлигини текширади. Агар каталог ёки файл сервер файллар системасида мавжуд бўлса `TRUE`, акс ҳолда— `FALSE` ни қайтаради. Бу функциялар ишининг натижаси кэшланади(компьютер хотирасидаги жилдда сақлаб қўяди). Ушбу кэшларни тозалаш тавсия этилади, бунинг учун дастур кодига `clearstatcache()` буйруғини қўшиб, мувозанат ҳолатидаги кэш тозаланади. Локал бўлмаган файллар учун `file_exists()` функциясини ишлатиб бўлмайди.

9.3 - мисол. `fclose()` функциясининг ишлатилиши

```
<?
$file = "user.txt";
if (file_exists($file))
{
    $fh = fopen($file, "r");
    // bu yerda fayllar bilan amallar bajariladi
    fclose($fh);
}
else
{print "File $file does not exist!";}
?>
```

9.4 - мисол. `file_exists()` функциясининг қўлланилиши

```
<?php
$filename = 'c:/users/files/my_file.html';
if (file_exists($filename)) {
    print " <b>$filename</b> fayli mavjud";
} else {
    print " <b>$filename</b> fayli mavjud emas ";
}
?>
Natija: c:/users/files/my_file.html fayli mavjud emas
```

Файлга ёзиш мумкинлигини текшириш: `is_writable()` функцияси

`bool is_writable (string файл)`

Файл мавжудлигини текширишдан ташқари, бу файлга маълумот ёзиш мумкинлигини ҳам текшириш мумкин, бунинг учун `is_writable()` функцияси ёки унинг псевдоними —`is_writable()` функциясидан фойдаланилади. Ёзиш имкониятининг мавжудлиги файл учун қандай текширилса, каталог учун ҳам шундай текширилади.

Синтаксиси:

`bool is_writable (файл ёки каталог номи)`

бу функция агар файл мавжуд бўлса ва унга ёзиш мумкин бўлса TRUE қийматини қайтаради. Файлга рухсат серверда ишлаётган фойдаланувчининг(кўпинча бу фойдаланувчи nobody ёки www) фойдаланувчи номи орқали амалга оширилади. `is_writable()` функцияси ишининг натижаси кэшланади.

Файлни ўқиш мумкинлигини текшириш: `is_readable()` функцияси

Файлдаги маълумотни ўқиш мумкинлигини текшириш учун `is_readable()` функциясидан фойдаланилади. Ўқиш имкониятининг мавжудлиги файл учун қандай текширилса, каталог учун ҳам шундай текширилади.

Синтаксиси:

`bool is_readable` (файл ёки каталог номи)

9.5 - мисол. `my_file.txt` файли мавжудлиги ва уни ўқиш учун рухсат борлигини `is_readable()` функцияси ёрдамида текшириш.

```
<?php
$filename = 'c:/users/files/my_file.txt';
if (is_readable($filename)) {
    print "<b>$filename</b> fayl mavjud va o`qish mumkin";
} else {
    print "<b>$filename</b> fayl mavjud emas yoki o`qish mumkin emas";
}
?>
```

Ёки

```
<?
$faylnomi="my_file.txt";
if(is_readable($faylnomi))
{
    // agar fayl mavjud va unga yozish mumkin bo`lsa
    $fh = fopen($faylnomi,"r");
    // faylni ochib ko`rsatkichni joriy pozitsiyasini fayl oxiriga o`rnatish
}
else { print "$faylnomi is not readable!";}
?>
```

9.5. Файлга маълумот ёзиш

fwrite() функцияси

fopen() функцияси ёрдамида рухсат очилган файлга маълумот ёзиш учун *fwrite()* функцияси ишлатилади.

Синтаксиси:

`int fwrite (файлга кўрсаткич, ўзгарувчи [, узунлиги])`

Бу функция *ўзгарувчи* таркибидаги сатрларни *файлга кўрсаткич* кўрсатган файлга ёзади. Агар қўшимча параметр - *узунлиги* берилган бўлса, у ҳолда ёзув ёзилаётган белгилар сони ушбу параметр қийматига тенг бўлганда ёки сатр охирига етганда тугатилади.

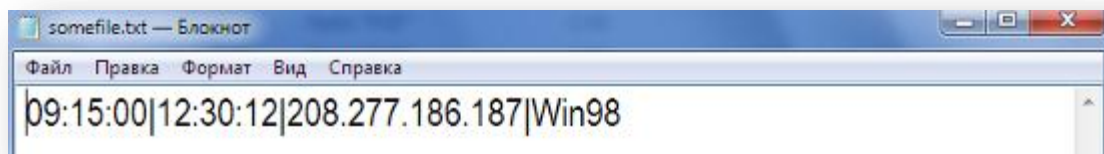
fwrite() функцияси ўзининг иши натижаси сифатида ёзув байти сонини ёки хатолик юзага келса `false` ни чиқазади.

9.6 - мисол. *fwrite()* функциясининг қўлланилиши

```
<?
// foydalanuvchi sayti trafigi haqidagi ma'lumot
$data = "09:15:00|12:30:12|208.277.186.187|Win98";
$filename = "somefile.txt";
// faylni ochib ko'rsatkichni fayl oxiriga o'rnatish
$fh = fopen($filename, "a+");
// agar fayl mavjud va unga yozish mumkinligini tekshirish
if ( is_writeable($filename) ) :
// $data tarkibini faylga yozish
$success = fwrite($fh, $data);
// faylni yopish
fclose($fh); else :
print "Could not open $filename for writing";
endif;
?>
```

Натижа

скрипт ишга тушганда сўнг `somefile.txt` файл таркибида:



9.1 – расм. Дастур коди ишга тушгандан сўнг ҳосил бўлган `somefile.txt` файли

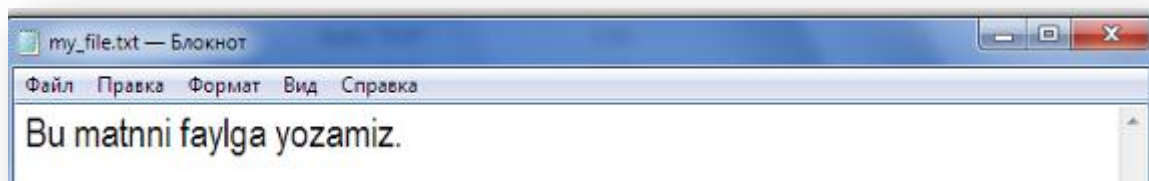
9.7 - мисол. *fwrite()* функциясининг қўлланилиши.

Бизнинг ишчи каталогимизда `my_file.txt` файли мавжуд эмас. Уни яратамиз ва унга матн сатрларини ёзамиз:

```
<?php
$faylnomi="my_file.txt";
$h = fopen($faylnomi,"w");
$text = "Bu matnni faylga yozamiz.";
if (fwrite($h,$text))
    echo "Yozuv muvaffaqiyatli yakunlandi";
else
```

```
echo "Ma'lumotlarni yozishda xatolik yuzaga keldi";
fclose($h);
?>
```

Natija:Yozuv muvaffaqiyatli yakunlandi



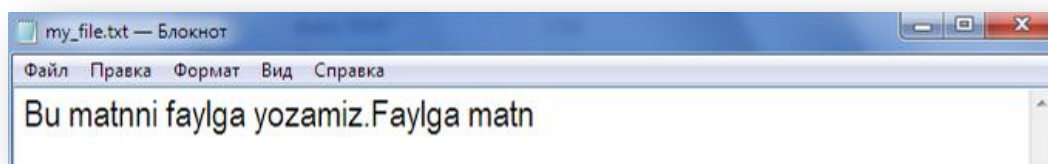
9.2 – расм. Дастур коди ишга тушгандан сўнг ҳосил бўлган *my_file.txt* файли

Бу скрипт иши натижаси сифатида агар ёзиш муваффақиятли амалга оширилса, браузерда *my_file.txt* файлини очганимизда «Bu matnni faylga yozamiz.» хабарини кўришимиз мумкин. Агар ушбу файл юқоридаги скрипти бажаргунча мавжуд бўлган бўлса, у ҳолда ундаги барча маълумотлар ўчирилади ва маълумот қайта ёзилади.

Агар биз қуйидагича скрипт ёзсак:

```
<?php
$faylnomi="my_file.txt";
$h = fopen("$faylnomi ", "a");
$add_text = "Faylga matn qo'shamiz.";
if(fwrite($h,$add_text,11))
    echo "Matn qo'shish muvaffaqiyatli amalga oshirildi <br>";
else echo "Ma'lumotlarni qo'shishda xatolik yuzaga keldi <br>";
fclose($h);
?>
```

Natija:Matn qo'shish muvaffaqiyatli amalga oshirildi



9.3 – расм. *my_file.txt* файлининг кейинги кўриниши

У ҳолда, аввалдан мавжуд бўлган *my_file.txt* файлидаги сатр давомига *\$add_text* ўзгарувчиси таркибидаги сатрнинг 11 та белгиси қўшилади, яъни «Faylga matn»

fwrite()(file write - файлга ёзиш) функциясининг у каби ишлайдиган **fputs()**(file put string – файлга сатрларни ёзиш) ўхшаши мавжуд. Бу функцияни **fwrite()** функцияси ўрнида ишлатиш мумкин ва унинг синтаксиси **fwrite()** кабидир. **fputs()** функциясининг синтаксиси:

int fputs (файлга кўрсаткич, ўзгарувчи [, узунлиги])

9.1 - масала. 9.2 - жадвалда берилган талабалар маълумотларини қайта ишловчи форма яратамиз. Бунда ҳар бир талабанинг маълумотлари битта

файлга ёзилади. Кейин ушбу ёзилган маълумотларни кўриш учун web - интерфейс яратамиз.

9.2 - жадвал. Талабалар ҳақидаги маълумотлар

stud_id	surname	name	stipend	kurs	city	birthday	univ_nomi
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

Маълумотларни файлга ёзишда жадвалнинг ҳар бир қатори файлнинг алоҳида сатрларига ёзилади. Сатрларни бир-биридан ажратиш учун янги сатр белгиси («\n») ишлатилади.

Маълумотлар майдонлари жадвалда берилган тартибда ёзилади, бу майдонларни ажратиш учун эса табуляция белгиси («\t») ишлатилади. Майдонларни ажратиш учун ўқиш осон бўлган ихтиёрий белгини ишлатиш мумкин.

Жадвалдаги маълумотларни **9.9 - листинг**да берилган форма ёрдамида файлга ёзамиз.

9.8 - листинг. *insert_form.php*. Талабалар маълумотларини киритиш учун форма.

```
<form action="insert.php" method=post>
<h2>Talaba ma'lumotlarini kiriting</h2>
  <table border=0 bgcolor=#cccccc>
    <tr>
      <td>Talaba id</td>
      <td><input type="text" name="stud_id" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Familiyasi</td>
      <td><input type="text" name="surname" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Ismi</td>
      <td><input type="text" name="name" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Stipendiyasi</td>
      <td><input type="text" name="stipend" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Kursi</td>
      <td><input type="text" name="kurs" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Manzili</td>
      <td><input type="text" name="city" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Tug'ilgan sanasi</td>
      <td><input type="text" name="birthday" size=10 maxlength=10></td>
    </tr>
    <tr>
      <td>Universiteti</td>
      <td><input type="text" name="univ" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td colspan=2 align=center><input type=submit value="Kiriting"></td>
```

```
</tr>
</table>
```

Talaba ma'lumotlarini kiriting

Talaba id

Familiyasi

Ismi

Stipendiyasi

Kursi

Manzili

Tug'ilgan sanasi

Universiteti

9.4 - расм. Талаба маълумотларини киритиш учун форма.

9.9 - листинг. Расмда тасвирланган формадаги маълумотларни файлга ёзувчи *insert.php* файли.

```
<?php
$faylnomi="student.txt";
$h = fopen($faylnomi,"a");
$text = $stud_id."\t".$surname."\t".$name."\t".$stipend."\t".$kurs."\t".$city
        ."\t".$birthday."\t".$univ."\n";
if (fwrite($h,$text))
    echo "Yozuv muvaffaqiyatli yakunlandi";
else
    echo "Ma'lumotlarni yozishda xatolik yuzaga keldi";
fclose($h);
?>
```

Natija:Yozuv muvaffaqiyatli yakunlandi

1	Islomov	Ikrom 150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil 200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid 150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir 100	2	Jizzax	8/12/1991	TDPU

9.5 - расм. Талаба маълумотларини киритилган *student.txt* файли.

9.6. Файлдан маълумотларни ўқиш

Файлдаги маълумотларни ўқиш файллар билан ишлашнинг энг асосий амалларидан биридир. Қуйида файлдан ўқишни самарадорлигини оширувчи бир қанча функциялар берилган.

Агар мавжуд файлдаги маълумотни ўқимоқчи бўлсак, маълумот ёзишдаги каби `foren()` функциясини ишлатиш етарли эмас. Бу очиқ файлга кўрсаткични қайтаради холос, лекин файлдаги бирорта сатрни ўқимайди. Шунинг учун, файлдан маълумотларни ўқиш учун, махсус функциялар(*file*, *readfile*, *file_get_contents*, *fread*, *fgets* ва ҳ.)дан бирортасидан фойдаланилади.

Файл охирини аниқлаш: *feof()* функцияси

PHP да файл позицияси кўрсаткичи файл охирига келганлигини кўришни текширувчи махсус функция мавжуд. Бу булев функция - ***feof()***. ***feof()*** функцияси битта параметр *файлга кўрсаткич*ни қабул қилади.

Синтаксиси:

string feof(файлга кўрсаткич)

Бу функция номини эслаб қолиш осон, унинг тўлиқ номи File End Of File (Файл: файл тугади).

Сатрларни берилган узунлигига кўра ўқиш: *fread()* функцияси

Бу функциядан файлдаги маълумотларни ўқишда фойдаланилади. Уни бинар файлдаги маълумотларни ўқиш учун ҳам ишлатиш мумкин.

Синтаксиси:

string fread (файлга кўрсаткич, узунлиги)

бу функция *файлга кўрсаткич* кўрсатган файлдан *узунлиги* параметрида кўрсатилган узунликка(байтга) тенг маълумотни ўқийди. *файлга кўрсаткич* параметри аниқ мавжуд бўлган, масалан `foren()` ёрдамида очилган файл билан боғланган ресурс типдаги ўзгарувчи бўлиши керак. *файлга кўрсаткич* очилган ва ўқиш учун рухсат берилган файлга юборилади. Маълумотларни ўқиш файл охиригача ёки *узунлиги* параметрида кўрсатилган байтга тенг сатрни ўқиб бўлгунча давом этади.

fread() функциясининг натижаси сифатида файлдаги сатрдан ўқилган маълумотлар чиқазилади.

Бу функцияда *узунлиги* параметри албатта кўрсатилиши керак. Шундай экан, агар биз файлдаги барча сатрларни ўқимоқчи бўлсак, у ҳолда унинг узунлигини билишимиз керак. PHP кўрсатилган файл узунлигини ҳисоблай олиш имкониятига эга. Бунинг учун ***filesize()*** функциясидан фойдаланилади.

Файл ўлчамини(узунлигини) аниқлаш: *filesize()* функцияси

Файл ўлчами ***filesize()*** функцияси ёрдамида аниқланади. Бу функция файл ўлчамини байтларда чиқазади.

Синтаксиси:

filesize(файл номи)

Бу функция **fread()** функцияси билан биргаликда файлни бир вақтда барчасини (ёки унинг аниқ бир қисмини) ўқиш учун ишлатилади.

Хатолик юзага келса, бу функция **false** ни қайтаради. Афсуски, уни фақатгина локал файлларни ҳажми(узунлиги)ни олишгагина ишлатиш мумкин.

9.10 - мисол. **fread()** функциясининг қўлланилиши.

my_file.txt файли таркибини ўқиймиз

```
<?php
$h = fopen("my_file.txt","r+") or die("Can't openfile!");
// faylni yozish va o`qish uchun ochamiz
$content = fread($h,
    filesize("my_file.txt"));
// fayl tarkibidagi satrlarni o`qiyamiz
echo filesize("my_file.txt").'<br>';
// faylni uzunligini chiqazamiz
fclose($h); // fayl bilan bog`lanishni yopamiz
echo $content;
// brauzer ekraniga fayl tarkibini chiqazish
?>
```

Natija: 36
Bu matnni faylga yozamiz.Faylga matn

Windows каби тизимларда мавжуд бинар файл(масалан расм)ни ўқиш учун файл таркибининг охирида **rb** ёки уни ўхшаши **b** байроғи ёрдамида очиш тавсия этилади.

filesize() функцияси ўз ишининг натижасини кешлаб қўяди(компьютер хотирасидаги жилдда сақлаб қўяди). Агар **my_file.txt** файли таркиби ўзгартирилиб, сўнг юқоридаги скрипт қайта ишга туширилса унинг ишини натижаси ўзгармайди. Скрипти бу файлдаги маълумотларни ўқиш учун бошқа функция(масалан, **fgetss()**)дан фойдаланиб ёзсак ҳам, натижа **fread()** даги каби бўлади.

*Самма-самма ўқиш: **fgets()**, **fgetss()**, **fgetcsv()** функциялари*

***fgets()** функцияси*

fgets() функцияси ёрдамида файлдаги матн сатри ўқилади. Бу функцияни синтаксиси **fread()** каби бўлиб, ундан фарқли тарафи ўқиладиган сатр узунлигини кўрсатиш шарт эмас:

string fgets (файлга кўрсаткич [, узунлиги])

fgets() функцияси ўз иши натижаси сифатида *файлга кўрсаткич* кўрсатган файлдан сатр узунлиги (узунлиги-1) байтга тенг сатрни чиқазади.

9.11 – мисол. **fgets()** функциясининг қўлланилиши

```
<?php
$h = fopen("my_file.txt","r+");
$content = fgets($h,2);
// my_file.txt faylining birinchi
// satridan birinchi belgini o`qiydi
fclose($h);
echo $content;
?>
```

Natija: B

fread() ва **fgets()** функцияларнинг иккаласи ҳам кўрсаткич агар файл охирига келса, файлдан маълумотларни ўқишни тўхтатади.

Масалан, my_file.txt файлидаги барча сатрларни қуйидагича ўқиш мумкин:

```
<?php
$h = fopen("my_file.txt", "r");
while (!feof($h)) {
    $content = fgets($h);
    echo $content, "<br>";
}
fclose($h);
?>
```

Natija: Bu matnni faylga yozamiz. Faylga matn

Эсга оламиз, РНР да битта белги – бу бир байт. Агар ўкиладиган сатр узунлиги кўрсатилмаса (берилаётган имконият РНР 4.2.0 версиясидан бошлаб киритилган), у ҳолда 1 Кбайт (1024 байт) матн ёки 1024 белги ўкилади.

РНР 4.3 версиясидан бошлаб, агар *узунлиги* параметри берилмаса, сатрни ҳаммаси ўкилади.

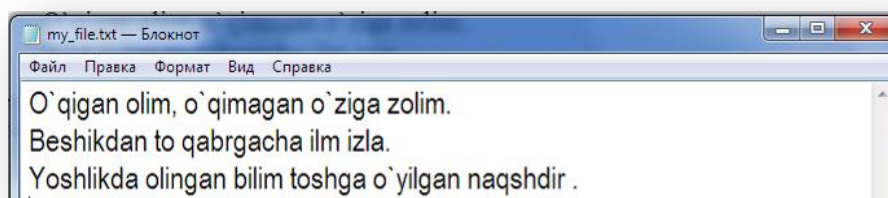
Хатолик юзага келса **fgets()** функцияси false ни қайтаради. РНР нинг 4.3 версиясидан бошлаб иккилик файллар учун бу функция хавфсиз ҳисобланади.

Қуйидаги шартлардан бирортаси бажарилса ўқиш тўхтатилади:

- файлдан (*узунлиги*-1) байт ёки белги ўқилган бўлса;
- файлдан кейинги(янги) сатрга ўтиш белгиси ўқилса;
- файлдан файл тугаганлиги белгиси(EOF) ўқилса;

Агар файлдан ҳар бир сатрни ўқиш талаб этилса, у ҳолда иккинчи параметр сифатида сатрдаги байтлар сонига тенг қийматни бериш керак. Файлдан сатрма-сатр ўқиш ва чиқазиш учун мисол келтирамиз:

9.12 - мисол. my_file.txt файли таркиби қуйидагича бўлсин.



9.6– расм. my_file.txt файлининг кўриниши

9.6 – расмдаги матнли файлни ўқиб браузерга чиказувчи дастур коди қуйидагича берилади:

```
<?
$fh = fopen("my_file.txt", "r");
while (!feof($fh)) {
    $line = fgets($fh, 4096);
    print $line. "<br>";
}
fclose($fh);
?>
```

Natija:
O`qigan olim, o`qimagan o`ziga zolim.
Beshikdan to qabrgacha ilm izla.
Yoshlikda olingan bilim toshga o`yilgan naqshdir .

fgetss() функцияси

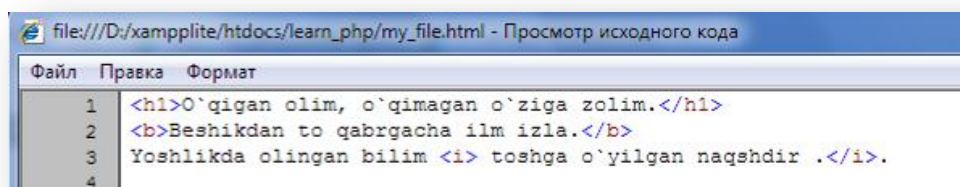
fgets() функциясининг яна бир ўхшаши бу – **fgetss()** функциясидир. Бу функция ҳам кўрсатилган файлдан сатрларни ўқиш учун имкон яратади ва шу билан бирга файлда учрайдиган барча **html** ва **PHP**– тегларни ўчиради.

Синтаксиси:

string fgetss(файлга кўрсаткич, узунлиги [, рухсат берилган теглар])

Эътибор беринг, бу ерда узунлиги параметри албатта бўлиши керак.

9.13 - мисол. **my_file.html** файли таркиби куйидагича бўлсин.

**9.7 – расм. my_file.html файлининг кўриниши**

Экранга **my_file.html** файлининг барча сатрларини **** ва **<i>** тегларидан бошқа барча тегларини ўчириб чиқазамиз:

```
<?php
$х = fopen("my_file.html", "r");
while (!feof ($х)) {
    $content = fgetss($х, 1024, '<b><i>');
    echo $content, "<br>";
}
fclose($х);
?>
Natija:
O`qigan olim, o`qimagan o`ziga zolim.
Beshikdan to qabrgacha ilm izla.
Yoshlikda olingan bilim toshga o`yilgan naqshdir ..
```

fgetcsv() функцияси

fgetcsv() — функцияси **fgets()** функциясининг яна бир варианты. У куйидагича **синтаксисга** эга:

array fgetcsv(int файлга кўрсаткич, int узунлиги, string [ажратувчи]);

бу функция файлдаги сатрларни *ажратувчи* сифатида берилган белги ёрдамида ажратиш учун ишлатилади. Ажратувчи сифатида ишлатилувчи белгилар электрон жадваллар ва б. ларда қўлланилган табуляция белгиси, вергул ва ҳ. лар бўлиши мумкин.

Бу функцияга **fgets()** функциясига мурожаат этган каби мурожаат этилади, фақатгина унга майдонларни ажратиш учун ишлатилган ажратувчи белги берилади. *Узунлиги* параметри файлдан ўқиладиган энг узун сатрдан катта узунликка эга бўлиши керак.

Агар берилган **9.1 - масала**миздаги талабанинг ҳар бир маълумотини матнли сатр сифатида эмас, балки алоҳида ўзгарувчига тиклаш талаб этилса, **fgetcsv()** функцияси ишлатилади. Ушбу функция файлдан сатрларни олиб, ҳар бир учраган табуляция белгисида сатрни ажратади. Ҳосил бўлган маълумотлар массивга жойлаштирилади (ушбу мисолда **-\$talaba** га). Массивлар билан ишлашни кейинги бўлимда кўрамиз.

9.14 – листинг. 9.1 – масалада талаба маълумотлари киритилган *student.txt* файлидаги маълумотларни массив шаклида чиқазамиз.view.php файли.

```
<?php
$h = fopen("student.txt", "r");
while (!feof($h)) {
    $talaba = fgetcsv($h, 100, "\t");
    print_r($talaba);
    echo "<br>";
}
fclose($h);
?>
```

Natija:

Array ([0] => 1 [1] => Islomov [2] => Ikrom [3] => 150 [4] => 1 [5] => Andijon [6] => 3/12/1992 [7] => TDPU)

Array ([0] => 3 [1] => Fozilov [2] => Fozil [3] => 200 [4] => 3 [5] => Qashqadaryo [6] => 1/12/1990 [7] => TDPU)

Array ([0] => 6 [1] => Salimov [2] => Vohid [3] => 150 [4] => 4 [5] => Namangan [6] => 7/06/1989 [7] => NamDU)

Array ([0] => 10 [1] => Karimov [2] => Botir [3] => 100 [4] => 2 [5] => Jizzax [6] => 8/12/1991 [7] => TDPU)

Белгини ўқиш: *fgetc()* функцияси

Агар файлдан маълумотларни сатр-сатр ўқиш мумкин бўлса, у ҳолда белгима-белги ўқиш ҳам мумкин. Бунинг учун **fgetc()** функциясидан фойдаланилади.

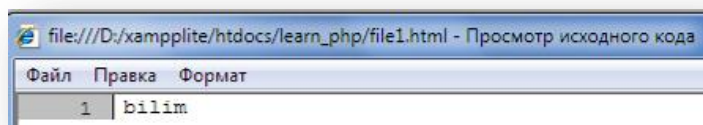
Синтаксиси:

string fgetc (файлга кўрсаткич)

Бу функция *файлга кўрсаткич* да кўрсатилган файлдаги белгиларни чиқазади ва агар сатр охири учраса FALSE қийматини қайтаради.

9.15 - мисол. Бу мисолда файл белгиларини **fgetc()** функцияси ёрдамида битталаб ўқиш кўрсатилган:

file1.html файли таркиби куйидагича бўлсин.



9.8 – расм. *file1.html* файлининг кўриниши

```
<?php
$h = fopen("file1.html", "r");
while (!feof ($h)) {
    $content = fgetc($h);
    echo $content. "<br>";
}
fclose($h);
?>
```

Natija:

b
i
l
i
m

Файл таркибини ўқиш учун `fopen()` функцияси ёрдамида файлни очиб, боғланиш ўрнатиш шарт эмас. PHP да бунини битта файл номини ишлатиб бажара оладиган функциялари мавжуд. Бу функциялар `readfile()`, `file()` ва `file_get_contents()`.

9.7. Файллар билан ишловчи функциялар

Readfile() функцияси

Синтаксиси:

int readfile (файл номи [, use_include_path])

readfile() функцияси *файл номи* параметрида берилган файлни ўқийди ва уни таркибини, белгилар байтини экранга чиқаради. Агар қўшимча параметр *use_include_path* TRUE қийматига тенг бўлса, у ҳолда берилган номдаги файлни *include_path* га кирувчи каталоглардан қидиради. Дастурда бу функция файлнинг ўқилган белгилари ва байтини чиқаради, акс ҳолда хатолик юзага келганда— FALSE ни қайтаради. Хатолик ҳақидаги хабарни бу функцияда @ амали ёрдамида яшириш мумкин.

9.16 - мисол. **readfile()** функциясининг қўлланилиши

Қуйидаги скрипт, *fayl.txt* файли мавжуд бўлса, уни таркибини ва ҳажмини чиқаради. Акс ҳолда хатолик юзага келганда «Error in readfile» сатрини чиқаради.

```
<?php
$n = @readfile ("somefile.txt");
/* fayl tarkibini ekranga chiqazadi va uning hajmini $n o'zgaruvchisiga yozib
qo'yadi*/
if (!$n) echo "Error in readfile";
/* arap readfile() funksiyasi xatolik bilan bajarilsa,
u holda $n=false va xato haqida xabarni chiqazamiz */
else echo $n;
// agar xato bo'lmasa, o'qilgan
// belgilar sonini chiqazamiz
?>
Natija:Fayllar bilan ishlashni o'rganamiz. 36
```

readfile() функцияси ёрдамида файл номи сифатида берилган, URL-манзили кўрсатилган, узоқ масофадаги файл таркибини ўқиш мумкин (агар бу опция серверни созлашда ўчирилмаган бўлса).

Файл таркибини дарҳол экранга чиқазиш ҳар доим ҳам қулай эмас. Гоҳида, кейинчалик бу файл билан бирор-бир амални бажариш имкони бўлишлиги учун файлдаги маълумотни ўзгарувчига ёзиш керак бўлади. Бунинг учун **file()** ёки **file_get_contents()** функцияларини ишлатиш мумкин.

Файлдаги маълумотларни массивга ўтказиш: file() функцияси

file() функцияси файлдаги маълумотларни массив типдаги ўзгарувчига ўтказиб ўқийди. Бу ерда массивнинг ҳар бир элементи файлнинг битта сатрига мос тушади. Бу функциянинг синтаксиси **readfile()** функцияси каби бўлиб, фарқли тарафи у натижаси сифатида массивни чиқаради.

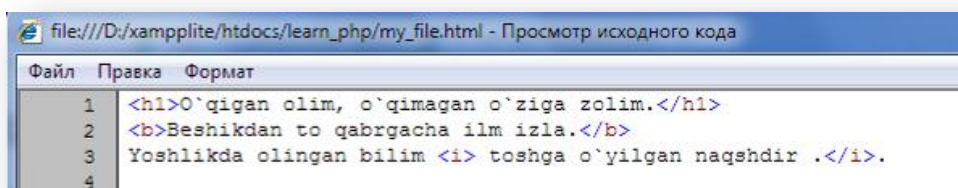
Синтаксиси:

array file (файл номи [, use_include_path])

Бу функция қандай массивни чиқаради? Берилган массивнинг ҳар бир элементи биз ўқиётган маълумотлар файли(унинг номи файл номи параметрида берилади)нинг сатри ҳисобланади. **file()** функциясида хатолик юзага келса, бошқалари каби false ни қайтаради. Қўшимча параметр *use_include_path* берилган файл *include_path* каталогида бор ёки йўқлигини аниқлаш учун ишлатилади. Агар сервер томонидан таъқиқланмаган бўлса, бу функция ёрдамида узоқ масофадаги файлларни ҳам очиш мумкин. PHP 4.3

версиясидан бошлаб бу функция орқали бинар файллар билан ишлаш хавфсиз ҳисобланади.

9.17 - мисол. *file()* функциясининг қўлланилиши *my_file.html* файли таркиби қуйидагича бўлсин.



9.9 – расм. *my_file.html* файлининг кўриниши

Унинг таркибини *file()* функцияси ёрдамида ўқиймиз:

```
<?
$file_array = file("my_file.html");
while (list( $line_num, $line) = each($file_array))
{
    print "<b>Satr $line_num:</b> ". htmlspecialchars($line). "<br>\n";
}
?>
```

Natija:

Satr 0: <h1>O`qigan olim, o`qimagan o`ziga zolim.</h1>
 Satr 1: Beshikdan to qabrgacha ilm izla.
 Satr 2: Yoshlikda olingan bilim <i> toshga o`yilgan naqshdir .</i>.
 Satr 3:

Файл кўрсаткичи ўрнини ўзгартириш: *rewind()*, *fseek()* ва *ftell()*

Файл ичида файл кўрсаткичининг ўрнини(позициясини) аниқлаш ва уни *rewind()*, *fseek()* ва *ftell()* функциялари ёрдамида ўзгартириш.

rewind() функцияси файл кўрсаткичини файлнинг бошланишига ўрнатади.

Синтаксиси:

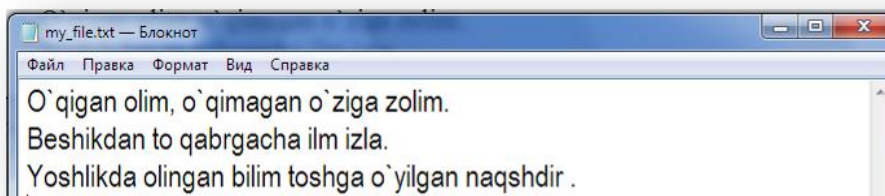
int rewind(файлга кўрсаткич);

ftell() функцияси эса кўрсаткич ўрнини файл бошидан бошлаб байтларда аниқлайди.

Синтаксиси:

int ftell(файлга кўрсаткич);

9.18 - мисол. *my_file.txt* файли таркиби қуйидагича бўлсин.



9.10 – расм. *my_file.txt* файлининг кўриниши

9.10 – расмдаги файлнинг таркибини ўқиймиз:

```
<?
$fh = fopen("my_file.txt", "r");
echo fread( $fh, filesize("my_file.txt" )) ;
echo "Fayl ko`rsatkichi o`rni ushbu nuqtada - ". (ftell ($fh));
echo "<br>";
rewind($fh);
echo "rewind dan so`ng, ko`rsatkich o`rni - ". (ftell($fh));
echo "<br>";
fclose($fh);
?>
```

Natija:

O`qigan olim, o`qimagan o`ziga zolim.

Beshikdan to qabrgacha ilm izla.

Yoshlikda olingan bilim toshga o`yilgan naqshdir.

Fayl ko`rsatkichi o`rni ushbu nuqtada - 124

rewind dan so`ng, ko`rsatkich o`rni - 0

Бу мисолда файлни ўқиб бўлгандан сўнг, кўрсаткич файлни охирини кўрсатади, яъни кўрсаткич сатр бошидан 124 байт силжийди. `rewind()` функциясига мурожаат этгандан сўнг, кўрсаткич яна 0 ўринга – файл бошига ўрнатилади.

fseek() функцияси файл кўрсаткичини файл ичидаги баъзи нуқталарга ўрнатиш учун ишлатилади.

Синтаксиси:

int fseek(int файлга кўрсаткич, int offset);

fseek() функциясига мурожаат этгандан сўнг, *файлга кўрсаткич* файл бошидан бошлаб *offset* байт силжиган нуқтага ўрнатилади.

`rewind()` функцияси `fseek()` функциясига нол миқдорда силжиш учун мурожаат этилганда (*offset=0* да) эквивалент бўлади.

Масалан, `fseek()` функциясини файлдаги ёзувларни ўртасини топишда ёки бинар қидирувларни бажаришда ишлатиш мумкин.

Файлни сатрларга ўқиб олиш: file_get_contents() функцияси

PHP нинг 4.3 версиясидан бошлаб файл таркибини сатрларга ўқиб олиш имконияти пайдо бўлди. Бу **file_get_contents()** функцияси ёрдамида бажарилади. Бу функция ҳам параметр сифатида *файл номи*ни олади ва `use_include_path` кўрсаткичига кўра уни `include_path` каталогидан қидиради.

Синтаксиси:

string file_get_contents (файл номи [, use_include_path])

бу функция **file()** функциясига жуда ўхшаш, фақатгина у файл таркибини сатр сифатида чиқазади. Бундан ташқари у бинар маълумотларга ишлов бериш учун хавфсиз ва агар сервер созлашларида таъқиқланмаган бўлса, масофада жойлашган файлдаги маълумотларни ўқий олади.

Файлни ўчириш: unlink() функцияси

Файл устида бажариладиган амаллардан бири – бу файлни ўчириш амали. PHP да файлни ўчириш учун `unlink()` функциясидан фойдаланилади.

Синтаксиси:

bool unlink (файл номи)

бу функция *файл номида* кўрсатилган номга эга бўлган файлни ўчиради ва агар функция иши муваффақиятли амалга оширилган бўлса TRUE ни, акс ҳолда хатолик юзага келса FALSE ни қайтаради. Файлни ўчириш учун ҳам, унга мос рухсат ҳуқуқига эга бўлиш керак (масалан, ўқиш учун берилган рухсат, файлни ўчиришга рухсат бермайди).

9.19 - мисол. *Unlink()* функциясининг қўлланилиши

```
<?php
$filename = 'd:/xampplite/htdocs/learn_php/my_file.html';
if (unlink($filename))
{echo $filename.' fayli o`chirildi'; }
else
{echo $filename.' fayli o`chirilmadi';}
// d:/xampplite/htdocs/learn_php/my_file.html
// nomli faylni o`chirish
?>
```

Natija:d:/xampplite/htdocs/learn_php/my_file.html fayli o`chirildi

Ташқи дастурларни юргизиш: exec() функцияси

RНР сценарийлари серверда жойлашган дастурларни ҳам бажариши мумкин. Бундай имкониятлар тизимни web- браузер орқали бошқаришда тез-тез ишлатилади, ҳамда тизим ҳақида маълумот олиш учун қулай.

exec() функцияси берилган дастурни ишга туширади ва у чиқарган маълумотларни охириги сатрини қайтаради.

Синтаксиси:

string exec (string буйруқ [, string массив [, int қайтариш]])

exec() функцияси иши натижаси сифатида ҳеч нарса чиқармаган ҳолда, фақатгина буйруқни бажаради. Буйруқ натижасида олинган барча маълумотлар шарт бўлмаган параметр - массивда сақланади. Бундан ташқари, агар берилган массив параметрида *қайтариш* ўзгарувчиси ҳам берилса, буйруқ бажарилиши сўнгида қайтариш кодлари ушбу ўзгарувчига тенгланади.

9.20 – мисол. *exec()* функциясининг ишлатилиши

Сервер билан алоқани текшириш

```
<?
exec("ping -n 5 www.php.net", $ping);
for ($i=0; $i< count($ping);$i++){
print " <br>$ping[$i] "; }?>
```

Natija:
При проверке связи не удалось обнаружить узел www.php.net.
Проверьте имя узла и повторите попытку.

9.21 – мисол. *exec()* функциясининг ишлатилиши

Сервер билан алоқани текшириш

```
<?
exec(" ping -n 5 192.168.75.253 ", $ping);
for ($i=0; $i< count($ping);$i++){
print " <br>$ping[$i] "; }?>
```

Натижа

Обмен пакетами с 192.168.75.253 по 32 байт:

Ответ от 192.168.75.253: число байт=32 время<1мс TTL=128

Ответ от 192.168.75.253: число байт=32 время<1мс TTL=128

Ответ от 192.168.75.253: число байт=32 время<1мс TTL=128

Ответ от 192.168.75.253: число байт=32 время<1мс TTL=128

Ответ от 192.168.75.253: число байт=32 время<1мс TTL=128

Статистика Ping для 192.168.75.253:

Пакетов: отправлено = 5, получено = 5, потеряно = 0 (0% потерь),

Приблизительное время приема-передачи в мс:

Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек

9.11 – расм. Сервер билан алоқани текшириш

Файллар тизими билан ишлаш: *basename()* функцияси

Сервердаги файлларни кўриш ва турли амалларни бажариш учун PHP нинг махсус функцияси мавжуд. Сервер файллари атрибутлари(турган жойи, эгаси ва ваколатлари) ҳақидаги маълумотлар тез-тез керак бўлади. **basename()** функцияси берилган тўлиқ номдан файл номини ажратиб олади.

Синтаксиси:

string **basename**(*string* тўлиқ номи)

файлнинг таянч номини тўлиқ номдан ажратиб олиш учун қуйидагича ёзилади:

```
<?
$path = "/usr/local/phppower/htdocs/index.php";
$file = basename ($path); // $file = "index.php"
?>
```

Бу функция тўлиқ номдан унга кўрсатилган йўлни ўчиради ва файл номини ўзини қолдиради.

9.8.Файлни муҳофазалаш(блокировкалаш)

Фараз қилайлик, бир вақтнинг ўзида иккита фойдаланувчи web-форма ёрдамида маълумотларни киритмоқчи. Биринчи фойдаланувчи `fopen()` функциясига мурожаат этиб, файлга ёзишни бошлади, иккинчи фойдаланувчи ҳам `fopen()` функциясига мурожаат этди ва файлга ёзиш учун уринмоқда. Натижада файл таркиби қандай бўлади? Аввал биринчи фойдаланувчи маълумоти ёзилади, кейин иккинчисиникими ёки аксинча? Ёки бу икки маълумот ихтиёрий тарзда галма-гал ёзиладими?

Бу саволларга жавоб бериш ишлатилаётган операцион тизимга боғлиқ ва уларга аниқ жавоб бериб бўлмайди. Шунга ўхшаш муаммолардан қочиш учун файлларни муҳофазалаш ишлатилади. РНР да муҳофазалаш `flock()` функцияси ёрдамида амалга оширилади. Бу функцияга файлни очгандан сўнг, лекин файлга маълумотларни ёзиш ва маълумотларни ўқишдан олдин мурожаат этилади.

Синтаксиси:

`bool flock (int файлга кўрсаткич, int блок_амали);`

бу функцияга параметр сифатида *файлга кўрсаткич* ва `сон(блок_амали)` берилади. Бу сон талаб қилинаётган муҳофазалаш амалининг тури. Агар муҳофазалаш муваффақиятли амалга ошса, функция `true` қийматини, акс ҳолда `false` ни қайтаради.

блок_амали параметрининг қийматлари(муҳофазалаш амалининг тури) 9.3 - жадвалда берилган.

9.3 - жадвал. *flock()* функциясининг *блок_амали* параметрининг қийматлари.

блок_амали параметрининг қиймати	Вазифаси
\	Ўқишни муҳофазалаш. Бу шуни англатадики, файл бошқа ўқувчи иловалар билан бирга(совместно) ишлатилиши мумкин.
2	Ёзишни муҳофазалаш. Бу якка (монополия) режими. Файл бирга ишлаш учун рухсатга эга эмас.
3	Мавжуд муҳофазадан чиқариш.
+4	<i>блок_амали</i> параметрининг жорий қийматига 4 қўшилса, жорий муҳофазалаш бажарилаётган вақтда бошқа муҳофазалашга уринишларни бартараф этади.

Агар `flock()` функциясини ишлатишга қарор қилинган бўлса, уни берилган файл ишлатилаётган тегишли барча саҳифаларга киритиш керак, акс ҳолда у ўз мазмунини йўқотади.

9.22 – мисол. Муҳофазалаш учун мисол кўрамиз, `learn.php` файлини қуйидагича ўзгартирамиз:

```
<?
$goshish="Yuqorida maqollar berilgan.";
$fh = fopen("my_file.txt", "a", 1);
flock($fh, 2); // yozish uchun faylni muhofazalash
fwrite($fh, $goshish);
flock($fh, 3); // yozish uchun muhofazalashni olib tashlash
```



```
fclose($fh);
?>
```

Ҳамда insert.php саҳифасида ҳам файлни блокировкакашни қўшамиз:

```
<?php
$fh = fopen("my_file.txt", "r");
flock($fh, 1); // fayldan o`qish uchun faylni muhofazalash
flock($fh, 3); // muhofazalashni olib tashlash
fclose($fh);
?>
```

Энди код ишончли, бироқ ҳали ҳам мукамал эмас. Агар иккита саҳифа бир вақтнинг ўзида битта файлни муҳофазалашга уринса, нима бўлади? Жараёнлар муҳофазалашни ўрнатишда бахслашиб қолиб, конфликт юзага келиши мумкин ва бунга қайсидир бири муваффақ бўлади, бироқ ўз навбатида яна янги муаммо туғилади. Шунинг учун бундай масалаларни ечишда МББТ ни ишлатилади, бу каби масалалар МББТда нисбатан муваффақиятли ечилади.

9.9. Серверга файлни юклаш

Энди бир мунча қийинроқ лекин амалда тез-тез қўлланиладиган–серверга файлни юклаш масаласини кўриб ўтамыз. Аввалроқ кўриб ўтган **9.1** – масаламизга ўзгартириш киритамиз, яъни ушбу форма ёрдамида фақатгина талаба маълумотлари эмас, балки талабанинг расми ҳам киритилсин. Бунинг учун серверга файл(расм)ни юклаб олувчи форма яратамиз, тўғрироғи insert_form.php саҳифасига ўзгартириш киритамиз. Бу форма ёрдамида файлни юклаб олиш учун form тегида enctype атрибутини multipart/form-data қиймати билан, ҳамда file типигаги input элементини ишлатамиз.

9.23 – мисол. Серверга файлни юклаш учун форма

```
<form enctype="multipart/form-data" action="insert.php" method=post>
<h2>Talaba ma'lumotlarini kiriting</h2>
  <table border=0 bgcolor=#cccccc>
    <tr>
      <td>Talaba id</td>
      <td><input type="text" name="stud_id" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Familiyasi</td>
      <td><input type="text" name="surname" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Ismi</td>
      <td><input type="text" name="name" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Stipendiyasi</td>
      <td><input type="text" name="stipend" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Kursi</td>
      <td><input type="text" name="kurs" size=5 maxlength=5></td>
    </tr>
    <tr>
      <td>Manzili</td>
      <td><input type="text" name="city" size=30 maxlength=30></td>
    </tr>
    <tr>
      <td>Tug'ilgan sanasi</td>
      <td><input type="text" name="birthday" size=10 maxlength=10></td>
    </tr>
    <tr>
      <td>Universiteti</td>
      <td><input type="text" name="univ" size=30 maxlength=30></td>
    </tr>
    <!--faylni yuklash-->
    <tr>
      <td>Rasmni yuklang: </td>
      <td>
        <input type="hidden" name="MAX_FILE_SIZE" size="50" value="30000" />
        <input type="file" name="myfile" /></td>
      <td colspan=2 align=center><input type=submit value="Kiriting"/></td>
    </tr>
  </table>
```

Бу формада яширин майдон ҳам иштирок этади. Ушбу яширин майдонда файл(расм)га рухсат этилган максимал ҳажм байтда кўрсатилади. Агар файлни юклаш жараёнида файл ҳажми бу майдонда кўрсатилган

ҳажмдан катта бўлса, у ҳолда хатолик ҳақида хабар акс эттирилади. Браузерда яратилган форма таркиби матн киритиладиган сатр, локал дискдан файлни танлаш учун тугма ва файлни жўнатиш учун тугмалардан ташкил топган. (9.12- расм).

9.12 - расм. Серверга файлни юклаш учун форма мисоли

Энди олинган файлни қайта ишлаш учун скрипт ёзамиз.

Серверга юкланган файл ҳақидаги барча маълумотлар глобал массив `$_FILES` таркибида бўлади. Бу массив PHP 4.1.0 версиясидан бошлаб киритилган. Агар `register_globals` директиваси ёқилган бўлса, у ҳолда узатилган ўзгарувчиларнинг қиймати уларнинг номи билан ишлатилиши мумкин.

Юқоридаги формада `Обзор...` тугмасини босиб бирор бир файл (масалан, 16.JPG) юклаб олиниб, `Faylni jo'nating` тугмаси босилади. Агар мижоз компьютеридан 16.JPG номли ҳажми 7759 байтга тенг файл юклаб олинган бўлса, у ҳолда `print_r($_FILES);` буйруғи ёрдамида қуйидагиларни экранга чиқазиш мумкин:

```
Array ( [myfile] => Array (
  [name] => 16.JPG
  [type] => image/jpeg
  [tmp_name] => d:\xampplite\tmp\phpEDDA.tmp
  [error] => 0
  [size] => 7759 ) )
```

Умуман олганда, `$_FILES` массиви ҳар доим қуйидаги элементларга эга:

- `$_FILES['myfile']['name']` – файлни мижоз машинасидаги номи.
- `$_FILES['myfile']['type']` – агар браузер бу маълумотни берган бўлса, жўнатилган файлнинг mime-типи. Бизнинг мисолда бу image/jpeg.
- `$_FILES['myfile']['size']` – юкланган файлнинг ҳажми(байтда).
- `$_FILES['myfile']['tmp_name']` – серверда сақланган файлнинг вақтинчалик номи.
- `$_FILES['myfile']['error']` – юклаш жараёнида юзага келган хатолар коди.

Бу ерда 'myfile' –серверга файлни юклашни амалга оширишга ёрдам берувчи форма элементи номи. Агар форма элементи номи бошқа ном билан номланса, бу ном ҳам бошқача бўлиши мумкин. Лекин бошқа калитлар (name, type ва ҳ.) ихтиёрий бошқа форма учун ўзгармай қолади.

Агар register_globals=On бўлса, у ҳолда `$_FILES['myfile']['name']` га эквивалент `$myfile_name` каби қўшимча ўзгарувчиларни ҳам ишлатиш мумкин бўлади.

PHP да файлни юклашда бешта типдаги хатолик вужудга келади ва мос равишда `$_FILES['myfile']['error']` бешта қийматга эга бўлади:

0 – хатолик юзага келмади ва файл муваффақиятли юкланди;

1 – юкланаётган файл ҳажми php.ini созлаш файлидаги upload_max_filesize директивасида ўрнатилган ҳажмдан ошиб кетган ;

2 – юкланаётган файл ҳажми html формадаги MAX_FILE_SIZE элементида кўрсатилган ҳажмдан ошиб кетган ;

3 – файл қисман юкланган;

4 – файл юкланмади;

Агар php.ini созлаш файлининг upload_tmp_dir опцияси ёрдамида бирор бир бошқа папка кўрсатилмаса, файл одатга кўра сервернинг вақтинчалик папкасига юкланади. Юкланган файлни керакли папкага ўтказиш учун move_uploaded_file() функциясидан фойдаланиш мумкин.

Синтаксиси:

bool move_uploaded_file (файлнинг_вақтинчалик_номи, белгиланган_жойи)

бу функция *файлнинг_вақтинчалик_номи* билан белгиланган файл ҳақиқатда HTTP механизмининг POST методи билан юкланганлигини текширади. Агар шундай бўлса, файл *белгиланган_жойи*(бу параметр таркибида йўл сифатида сақлаш учун янги папка, ҳамда янги файл номи жойлашган) параметрида берилган файлга ўтказилади.

Агар *файлнинг_вақтинчалик_номи* да нотўғри юкланган файл берилса, у ҳолда ҳеч қандай амал бажарилмайди ва **move_uploaded_file()** функцияси FALSE ни қайтаради. Агар файл бирор бир сабаб билан *белгиланган_жойи* га ўтказилмаса ҳам функция юкоридаги натижани қайтаради. Бу ҳолларда интерпретатор мос огоҳлантиришни чиқазади. Агар файл берилган *белгиланган_жойи* параметрида мавжуд бўлса, у ҳолда **move_uploaded_file()** функцияси уни қайта ёзади. Демак юкоридаги форма ёрдамида файл юклаб олинди, вақтинчалик юкланган жойидан керакли каталогга файлни

ўтказамиз. Буни учун қуйидаги скиптни ёзамиз ва бу саҳифани insert.php деб номланган, талаба маълумотларини файлга ёзувчи саҳифамизга ҳам ўзгартириш киритамиз.

9.24 – мисол. Файлни серверга юклаш дастури

```
<?
//ma`lumotlarni faylga yozish
$faylnomi="student.txt";
$h = fopen($faylnomi,"a");
$text = $stud_id."\t".$surname."\t".$name."\t".$stipend."\t".$kurs."\t".$city
."\t".$birthday."\t".$univ."\n";
if (fwrite($h,$text))
    echo "Yozuv muvaffaqiyatli yakunlandi";
else
    echo "Ma`lumotlarni yozishda xatolik yuzaga keldi";
fclose($h);
//rasm(fayl)ni yuklash
/* PHP ning 4.1.0 dan avvalgi versiyalarida
$_FILES massivi bilan birga
$_HTTP_POST_FILES massivini ham
ishlatish kerak*/
$uploadaddir = 'D:/xampplite/htdocs/learn_php/upload/';
// ushbu papkaga
// yuklangan rasm(fayl)ni saqlaymiz
$destination = $uploadaddir . $_FILES['myfile']['name'];
// rasm(fayl) nomini o`zgartirmaymiz
print "<pre>";
if (move_uploaded_file(
    $_FILES['myfile']['tmp_name'],
    $destination)) {
    /* rasmni(faylni) vaqtinchalik papkadan
    saqlash uchun tanlangan papkaga o`tkazamiz*/
    print "Fayl muvaffaqiyatli yuklandi <br>";
} else {
    echo "Faylni yuklashda xatolik yuzaga keldi.
    Ba`zi bir ma`lumotlar:<br>";
    print_r($_FILES);
}
print "</pre>";

?>
Natija:
Yozuv muvaffaqiyatli yakunlandi
Fayl muvaffaqiyatli yuklandi
```

Натижада файл вақтинчалик юкланган жойидан кўрсатилган папкага ўтказилди. Яъни файлни юклаб олиш муваффақиятли бажарилди.

9.10.Қайта ишлашнинг рационал усули: Маълумотлар базасини бошқариш тизими

Барча кўриб ўтилган мисолларда икки ўлчамли файллар ишлатилди. Кейинги бўлимда MySQL — реляцион маълумотлар базасини бошқариш тизимини кўриб ўтамиз. Савол туғилиши мумкин: «Бу нима учун керак?»

Иккилик файллар(двумерных файлов)да ишлаш билан боғлиқ муаммолар:

Икки ўлчамли файллар билан ишлашда қатор муаммоларга дуч келинади:

- Икки ўлчамли файл катталашган сари у билан ишлаш секинлашади.
- Аниқ бир ёзувни ёки ёзувлар гуруҳини файлдан қидириш қийин. Агар ёзувлар тартибланган бўлса, у ҳолда қидириш учун калитли майдонда ёзувнинг фиксирланган узунлигини қўллаш билан бирга бинар қидирув турининг бирор-бири ишлатилади. Агар бирор-бир шаблонга мос маълумотни қидириш керак бўлса(масалан, Андижонда яшовчи барча талабаларни), у ҳолда ҳар бир ёзувни алоҳида ўқиш ва текшириш керак бўлади.

- Рақобатли рухсат (берилган рухсат бўйича бир вақтнинг ўзида икки ёки ундан ортиқ фойдаланувчи битта амални бажариши) муаммолар келтириб чиқариши мумкин. Худди аввал кўриб ўтилган, файлни муҳофазалашда юзага келган конфликт ҳодисалар каби. Бундан ташқари, тармоқдаги «нозик нуқталарни» пайдо бўлишига сабаб бўлади. Сайтдаги кўп миқдордаги фойдаланувчилардан етарли даражада шиддат билан ишлаётган маълумотлар оқимига ўзларининг маълумотларини киритиш учун кутиш талаб этилади(чунки бу вақтда файлнинг муҳофазасини олиб ташлаш иши бажарилаётган бўлади). Агар кутиш узоқ вақт давом этса, фойдаланувчилар бошқа сайтлардан фойдаланишни маъқул кўришади.

- Қаралаётган файлларни қайта ишлаш кетма-кетликда амалга оширилиши керак — ўқиш файлнинг бошидан бошлаб, унинг охиригача бажарилади. Агар ёзув қўшиш ёки ёзувни файлнинг ўртасидан ўчириш зарур бўлса, у ҳолда аввал файлнинг барчасини хотирага ўқиб олиш, ўзгартиришни бажариш ва файлга қайтадан ёзишга тўғри келинади, бу охир-оқибат мушкул ишга айланади. Катта ҳажмдаги маълумотли файллар билан ишлашда бу жараён сезиларли даражада тизимнинг ортиқча юкланишига сабаб бўлади.

- Файллар билан ишлаётганда чегараланишдан ташқари, маълумотларга турли даражадаги фойдаланиш ҳуқуқларини таъминлашнинг ҳеч қандай усули йўқ.

Бу муаммолар РМББТда қандай ечилади?

Реляцион маълумотлар базасини бошқариш тизими (РМББТ) ушбу барча муаммоларни ҳал этади:

- РМББТ маълумотлардан фойдаланишни иккилик файлга нисбатан тезроқ амалга оширади. MySQL маълумотлар базасини бошқариш тизими эса барча РМББТ ичида энг кўп фойдаланилаётган тизимлардан биридир.

- РМББТ да аниқланган киритериялар асосида маълумотлар тўпламини олиш учун сўровлар осон жўнатилади.

- РМББТ одатда кўп сонли сўровлар билан ишлаш механизмига эга бўлади, бу эса РМББТ дан фойдаланувчи дастурчини кўп сонли сўровлар ҳақида ўйлашдан озод этади.

- РМББТ маълумотлардан ихтиёрий тарзда фойдаланишни таъминлайди.

- РМББТда фойдаланувчи ҳуқуқларини таъминлаш тизими мавжуд бўлиб, MySQL эса бу борада алоҳида имкониятларга эга.

Кўриниб турибдики, маълумотларни сақлаш тизимини ташкил этишда РМББТ дан фойдаланиш функционал имкониятларни беради.

Х БОБ. МАССИВЛАР

PHP нинг массив элементлари билан ишловчи бир қанча функциялари мавжуд ва улар: массив элементини қидириш, массив элементини саралаш; бир ёки бир нечта, биринчи ёки охирги элементларни қўйиш ва ўчириш; массив ҳажмини катталаштириш; стандарт элементларни қўйиш каби имкониятларга эга. Бу функциялар массивни саралаш билан боғлиқ тез-тез учраб турадиган масалаларни ечишда ёрдам беради. Бу бобда бундай функцияларни бир нечтасини кўрамиз ва улар ёрдамида бир қанча амалий масалаларни ечамиз. Массив элементини қидириш, массив элементини саралаш.

10.1 Массивлар

Олдинги маърузаларда массив типдаги маълумотлар ҳақида гапириб ўтганмиз. Массивни икки хил усул билан яратиш мумкин:

Array()

1. array() функцияси ёрдамида

```
$array_name= array ([value1, value2,...]);
```

Ассоциатив массивларда

```
$array_name = array («key1»=>«value1»,  
«key2»=>«value2»);
```

2. тўғридан-тўғри қиймат бериш йўли билан

```
$array_name []=value1;
```

Ассоциатив массивларда

```
$array_name[«key1»]= value1;
```

Биздан маълумотлар базасидан ўчирилиши керак бўлган фанлар рўйхатини сақлаш талаб этилган бўлсин. Уларни массив кўринишида сақлаш мумкин: бу ерда калит - фан идентификатори (уни уникал номери), қиймат эса – фан номи. Бу массивни қуйидагича яратиш мумкин:

```
<?
$del_items = array("10"=>"Fizika",
                  "12"=>"Infromatika");
$del_items["13"] = "Php da dasturlash";
// massivga element qo'shamiz
?>
```

10.1 – масала. Аввалги бўлимда кўриб ўтган талабалар ҳақидаги маълумотлар жойлашган жадвал ҳам массив типдаги маълумотларга мисол бўла олади. Массивлар билан ишлашга доир мисолларда ушбу жадвалдан фойдаланамиз.

10.1 - жадвал. Талабалар ҳақидаги маълумотлар

stud_id	surname	name	stipend	kurs	city	birthday	univ_nomi
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

Ушбу жадвалда берилган талабаларнинг исмларидан ташкил топган массивни ёзамиз:

```
<?
$stalabalar=array("Ikrom", "Fozil", "Vohid", "Botir");?>
```

Бу массив таркибини кўриш учун ўзгарувчи номи ва калит (индекс) керак бўлади. Ушбу массив таркибини чиқазамиз.

```
echo "$stalabalar[0] $stalabalar[1] $stalabalar[2] $stalabalar[3]";
```

Агар массив кетма-кет сонлар билан индексланган бўлса, уни таркибини чиқариш учун циклдан фойдаланиш мумкин:

```
<?
$stalabalar=array("Ikrom", "Fozil", "Vohid", "Botir");
for ($i=0; $i<4; $i++)
{
    echo "$stalabalar[$i] ";
}
?>
```

Natija:Ikrom Fozil Vohid Botir

Кўп ўлчовли массивлар

Энди жадвалдаги маълумотларни барчасини массив шаклида аниқлаймиз ва қуйидагича PHP кодини ёзамиз:

```
<?
$stalabalar=array(
    array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
    array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
    array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamDU"),
    array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
);
?>
```

Бундан кўриниб турибдики, талабалар массиви тўртта массивдан ташкил топган, яъни кўп ўлчовли массив. Бир ўлчовли массивга мурожаат этишда массив номи ва унинг индексидан фойдаланган эдик. Икки ўлчовли массивда эса, ҳар бир элементнинг иккита индекси бўлади – сатри ва устуни. Бу массивнинг таркибини кўриш қуйидагича амалга оширилади:

```
echo "|".$stalabalar[0][0]."|".$stalabalar[0][1]."|".$stalabalar[0][2]."|".$stalabalar[0][3]."|".$stalabalar[0][4]."|".$stalabalar[0][5]."|".$stalabalar[0][6]."|".$stalabalar[0][7]."|<br>";
echo "|".$stalabalar[1][0]."|".$stalabalar[1][1]."|".$stalabalar[1][2]."|".$stalabalar[1][3]."|".$stalabalar[1][4]."|".$stalabalar[1][5]."|".$stalabalar[1][6]."|".$stalabalar[1][7]."|<br>";
echo "|".$stalabalar[2][0]."|".$stalabalar[2][1]."|".$stalabalar[2][2]."|".$stalabalar[2][3]."|".$stalabalar[2][4]."|".$stalabalar[2][5]."|".$stalabalar[2][6]."|".$stalabalar[2][7]."|<br>";
```

Ёки ушбу натижани олиш учун циклдан фойдаланиш ҳам мумкин:

```
<?
$stalabalar=array(
    array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
    array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
    array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamDU"),
    array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
);
for($row=0;$row<4;$row++)
{
    for($column=0; $column<8; $column++)
    {
        echo "|".$stalabalar[$row][$column];
    }
    echo "|<br>";
}
?>
```

Natija:

```
|1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|
|3|Fozilov|Fozil|200|3|Qashqadaryo|1/12/1990|TDPU|
|6|Salimov|Vohid|150|4|Namangan|7/06/1989|NamDU|
|10|Karimov|Botir|100|2|Jizzax|8/12/1991|TDPU|
```

Жадвалдаги маълумотларнинг массивини устун номларини ишлатиб ҳам ҳосил қилиш мумкин. Бундай массив ассоциатив массив бўлади.

```
$talabalar=array(
    array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150, kurs=>
1, city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU"),
    array(stud_id=>"3",surname=>"Fozilov", name=>"Fozil", stipend=>200, kurs=>
3, city=>"Qashqadaryo", birthday=>"1/12/1990", univ_id=>"TDPU"),
    array(stud_id=>"6",surname=>"Salimov", name=>"Vohid", stipend=>150, kurs=>
4, city=>"Namangan", birthday=>"7/06/1989", univ_id=>"NamDU"),
    array(stud_id=>"10",surname=>"Karimov", name=>"Botir", stipend=>100, kurs=
>2, city=>"Jizzax", birthday=>"8/12/1991", univ_id=>"TDPU")
);
```

Бундай массивнинг таркибини кўриш учун ҳам циклардан фойдаланиш мумкин:

```
<?
$talabalar=array(
    array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150, kurs=>
1, city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU"),
    array(stud_id=>"3",surname=>"Fozilov", name=>"Fozil", stipend=>200, kurs=>
3, city=>"Qashqadaryo", birthday=>"1/12/1990", univ_id=>"TDPU"),
    array(stud_id=>"6",surname=>"Salimov", name=>"Vohid", stipend=>150, kurs=>
4, city=>"Namangan", birthday=>"7/06/1989", univ_id=>"NamDU"),
    array(stud_id=>"10",surname=>"Karimov", name=>"Botir", stipend=>100, kurs=
>2, city=>"Jizzax", birthday=>"8/12/1991", univ_id=>"TDPU")
);
for($row=0;$row<4;$row++)
{
    echo "|".$talabalar[$row]["stud_id"]."|".$talabalar[$row]["surname"]."|".
$talabalar[$row]["name"]."|".$talabalar[$row]["stipend"]."|".$talabalar[$row]["kurs"].
"|".$talabalar[$row]["city"]."|".$talabalar[$row]["birthday"]."|".
$talabalar[$row]["univ_id"]."|<br>";
}
?>
```

```
Natija: |1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|
|3|Fozilov|Fozil|200|3|Qashqadaryo|1/12/1990|TDPU|
|6|Salimov|Vohid|150|4|Namangan|7/06/1989|NamDU|
|10|Karimov|Botir|100|2|Jizzax|8/12/1991|TDPU|
```

10.2. Массивлар устида амаллар

Массив – бу маълумот типи, маълумотларни бу типи устида амаллар бажариш мумкин. Массивларни қўшиш ва таққослаш мумкин.

Массивларни қўшиш учун «+» операторидан фойдаланамиз. Умуман олганда бу амал массивларни бирлаштириш амали. Бизда иккита \$a ва \$b массивлари мавжуд, уларни қўшиш(бирлаштириш) натижаси \$a ва \$b массивлардан ташкил топган \$c массивга тенг бўлади. Агар мос калитлар учрашса, у ҳолда натижада биринчи массив(яъни, \$a)нинг элементлари олинади.

10.1 - мисол. Массивларни қўшиш

```
<?
$a = array("i"=>"Informatika", "m"=>"Matematika");
$b = array("i"=>"Ingliz tili", "m"=>"Manbashunoslik", "f"=>"Fizika");
$c = $a + $b;
$d = $b + $a;
print_r($c). '<br>';
/* Array([i]=>Informatika [m]=>Matematika [f]=>Fizika) */
print_r($d);
/* Array([i]=>Ingliz tili [m]=>Manbashunoslik [f]=>Fizika) */
?>
Natija:
Array ( [i] => Informatika [m] => Matematika [f] => Fizika)
Array ( [i] => Ingliz tili [m] => Manbashunoslik [f] => Fizika)
```

Массивларни тенг ёки тенг эмас, эквивалент ёки эквивалент эмаслигини текшириш ёрдамида таққослаш мумкин.

Барча массив элементларининг калит/қиймат жуфтликлари мос тушса бу массивлар тенг дейилади.

Солиштирилаётган массивларнинг калит/қиймат жуфтликлари бир хил тартибда ёзилса ва тенг бўлса, у ҳолда бу массивлар эквивалент массивлар дейилади.

PHP да тенг қиймати - «==» белгиси билан, эквивалент эса - «===» белгиси билан белгиланади.

10.2 - мисол. Массивларни таққослаш

```
<?
$a = array("i"=>"Informatika", "m"=>"Matematika");
$b = array("m"=>"Matematika", "i"=>"Informatika");
if ($a == $b) echo "Massivlar teng va";
else echo " Massivlar teng emas va";
if ($a === $b) echo " ekvivalent";
else echo " ekvivalent emas";
// " Massivlar teng va ekvivalent emas "
// natijasini olamiz
?>
Natija:Massivlar teng va ekvivalent emas
```

10.3. Массивлар билан ишловчи функциялар

each() функцияси

Бу функция массивнинг жорий элементини қайтаради(чиқазади) ва кейинги элементини жорий деб белгилайди.

Синтаксиси:

each(массив);

each() функциясига *while* цикли ичида мурожаат этилганда, у навбатма-навбат массивнинг ҳар бир элементини қайтаради ва массив охирига етгач ўз ишини тўхтатади.

```
<?
$talabalar=array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150
, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU");
while( $element = each( $talabalar ) )
{echo $element[ "key" ];
echo " - ";
echo $element[ "value" ] ;
echo "<br>";
}

?>
Natija: stud_id - 1
surname - Islomov
name - Ikrom
stipend - 150
kurs - 1
city - Andijon
birthday - 3/12/1992
univ_id - TDPU
```

Бу мисолда *\$element* ўзгарувчиси ҳам массив элементларини қабул қилиб олаётганлиги учун массив бўлади. *each()* функциясига мурожаат этганда, у саккизта қийматдан ва саккизта индексдан иборат катаклардан ташкил топган массивни беради. *key* ва 0 катаги жорий элементнинг калити, *value* ва 1 катаги – жорий элементнинг қийматларидан таркиб топган. Катаклардан қайси бирини танлаш аҳамиятсиз, ушбу мисолда рақамланган катакларни эмас, номли катакларни ишлатдик, яъни ассоциатив массивдан фойдаландик.

List() функцияси

List() функцияси *array()* га ўхшаш бўлиб, унинг асосий вазифаси – бир вақтнинг ўзида массив ичидаги бир нечта қийматни бир нечта ўзгарувчига тенглаштиради. Бу функция берилган массивни бир нечта қийматларга ажратади.

Синтаксиси:

void list(ўзгарувчи1 [,ўзгарувчи2 , ...])

list() - маълумотлар базасидан ёки файлдан маълумотларни ўзгарувчига ўқиб олиш учун жуда қулай функция.

	0	1	2	3	4
\$name=	Lola	Karim	Vasila	Akmal	Botir

list(\$stalaba1, \$stalaba2, \$stalaba3, \$stalaba4, \$stalaba5)=\$name

Бу ерда \$stalaba1= Lola, \$stalaba2=Karim, \$stalaba3= Vasila, \$stalaba4=Akmal, \$stalaba5= Botir

10.3 - мисол. **list()** функциясини қўллаш. 10.1 – жадвалдан фойдаланамиз.

```
<?
$stalabalar=array("1", "Islomov", "Ikrom", "150", "1", "Andijon", "3/12/1992",
    "TDPU");
list($stud_id, $surname, $name, $stipend, $kurs, $city, $birthday, $univ_id)=
$stalabalar;
echo $stud_id.'<br>';
echo $surname.'<br>';
echo $name.'<br>';
echo $stipend.'<br>';
echo $kurs.'<br>';
echo $city.'<br>';
echo $birthday.'<br>';
echo $univ_id;
?>
```

Natija:

```
1
Islomov
Ikrom
150
1
Andijon
3/12/1992
TDPU
```

10.4 - мисол. List() функциясини each() функцияси билан қўллаймиз.

```
<?
$stalabalar=array(stud_id=>"1", surname=>"Islomov", name=>"Ikrom", stipend=>150
, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU");
while( list($key, $value) = each( $stalabalar ) )
    {echo "$key - $value <br>";
    }
?>
```

Natija: stud_id – 1
 surname – Islomov
 name – Ikrom
 stipend – 150
 kurs – 1
 city – Andijon
 birthday - 3/12/1992

univ_id - TDPU

бу мисолда `each()` функцияси `$talabalar` массивидан жорий элементни олиб, уни массив сифатида қайтаради ва кейинги элементни жорий этади. `List()` функцияси эса `each()` функцияси қайтарган массивнинг 0 - ва 1 – элементини иккита янги ўзгарувчига(`$key` ва `$value`) тенглайди.

range() функцияси

range() функцияси қуйи ва юқори чегараси аниқ бўлган бутун сонлар интервалини осон ва тез яратиш имконини беради. **range()** функцияси кўрсатилган интервалдаги барча бутун сонлардан ташкил топган массивни чиқазади.

Синтаксиси:

`array range(int қуйи_чегара, int юқори_чегара).`

10.5 – мисол. range() функциясини қўллаш

```
<?
$lot= range(0,9);
// $lot=array(0,1,2,3,4,5,6,7,8,9) .
?>
```

range() функциясида 0 дан 9 гача бўлган интервал кўрсатилганлиги учун `$lot` массиви кўрсатилган интервалдаги бутун сонлардан иборат бўлади.

Массив элементларини ҳисоблаш: sizeof(),count(),array_count_values() sizeof() функцияси

Массивларни устида яна бир муҳим амални бажариш мумкин - массив элементлари сонини аниқлаш. **sizeof()** функцияси - массив элементлари сонини ҳисоблайди. Яна бир шунга ўхшаш функция **sizeof()** нинг кенгайтирилган формаси - бу **count()** функциясидир.

count() функцияси

count() функцияси **sizeof()** га ўхшаш бўлиб, массивдаги қийматларнинг сонини ҳисоблайди. Умуман олганда, бу функция ўзгарувчининг элементлари сонини ҳисоблайди.

Агар бу функцияни массив ўзгарувчидан бошқа бирор бир ўзгарувчи учун қўлланилса, натижа 1 чиқади. `NULL` типдаги ўзгарувчи эса бундан мустасно, яъни `count(NULL)` 0 га тенг.

Бу функция кўп ўлчовли массив элементлари сонини ҳисоблаш учун қўлланилганда, қўшимча параметр `COUNT_RECURSIVE` ишлатилади.

Синтаксиси:

`count(ўзгарувчи, [COUNT_RECURSIVE]);`

10.6 -мисол. count() функциясини қўллаш

```
<?
$del_items = array("langs" => array( "10"=>"Python", "12"=>"Lisp"),
                  "other"=>"Infomatika");
echo count($del_items) . "<br>";
// 2 chiqadi
echo count($del_items,COUNT_RECURSIVE);
// 4 chiqadi
?>
Natija:
2
```


array_count_values()* функцияси*Синтаксиси:*****array_count_values()***(ўзгарувчи);

array_count_values(\$ar) функцияси \$ar массивида уникал қиймат неча марта учраганлиги сонини чиқазади.

10.7 - мисол. array_count_values() функциясини қўллаш

```
<?
$ar=array(4,5,1,2,3,1,2,2,1,2,1,1);
$nat=array_count_values($ar);
echo '<table>';
echo '<tr><td>qiymatlar</td><td>soni</td></tr>';
for($i=1;$i<=5;$i++){
echo '<tr><td>'. $i. '</td><td>'. $nat[$i]. '</td></tr>';
}
echo '</table>';
?>
```

Natija:

qiymatlar	soni
1	5
2	4
3	1
4	1
5	1

Натижа шуни кўрсатадики, \$ar массивида 1 - 5 марта, 2 – 4 марта, 3,4,5 қийматлари 1 марта учраган.

Массив элементлари билан ишлаш: *In_array()*, *array_search()*, *array_keys()*, *array_values()*, *array_unique()*

***In_array()* функцияси**

in_array() - массив элементларини қидирувчи функция. Берилган массивда қидириладиган қиймат мавжудлигини текширади.

Синтаксиси:***in_array()***(«қидириладиган_қиймат», «массив», [«тип чегараси»]);

Агар учинчи параметр true сифатида берилган бўлса, у ҳолда массивдан фақатгина қиймати бўйича эмас, балки типи бўйича ҳам мос элементни қидиради. Агар қидириладиган қиймат – сатр бўлса, у ҳолда таққосланаётганда регистрга аҳамият беради.

10.8 - мисол. Берилган массив элементлари фанлар рўйхатидан ташкил топган. Берилган массивда Informatika фани мавжудми ёки йўқ қидирамиз.

```
<?php
$dfan = array("Matematika","Fizika","Falsafa","Informatika","Tarix");
if (in_array("Informatika",$dfan))
echo "Informatika fanidan imtihon topshiraman<br>";
// " Informatika fanidan imtihon topshiraman" xabarini chiqazadi
if (in_array("informatika",$dfan))
echo " informatika fanidan imtihon topshiraman<br>";
// xech narsa chiqmaydi,
// chunki " Informatika " satri mavjud, lekin " informatika " mavjud emas.
```

```
// ya`ni bu funksiya registrga sezgir
?>
Natija:Informatika fanidan imtihon topshiraman
```

In_array() функциясида қидирилаётган қиймат массив шаклида ҳам бўлиши мумкин. Фақатгина бу имконият PHP 4.2.0. версиясидан бошлаб қўшилган.

10.9 - мисол.

```
<?php
$dfan = array("Matematika", "Fizika", array("Falsafa", "Informatika"), "Tarix");
if (in_array(array("Informatika", "Falsafa"), $dfan))
    echo "Informatika va Falsafa fanlaridan imtihon topshiraman<br>";
?>
Natija:Informatika va Falsafa fanlaridan imtihon topshiraman
```

array_search() функцияси

Ушбу функция ҳам *in_array()* функцияси каби массив элементларини қидиради. *array_search()* функциясининг *in_array()* функциясидан фарқли тарафи агар элемент топилган бўлса, калитини чиқазади, акс ҳолда false ни қайтаради.

Синтаксиси:

array_search() («қидирилаётган_қиймат», «массив», [«тип чегараси»]);

Сатрларни солиштириш регистрга сезгир, агар учинчи параметр кўрсатилган бўлса, у ҳолда қиймат типини ҳам солиштиради. PHP нинг 4.2.0 версиясигача, агар қидирилаётган қиймат топилмаса, функция хато ёки бўш NULL қийматни қайтаради.

10.10 - мисол. Энди юқоридаги мисолни тескарисини кўриб ўтамиз, имтиҳон топширилган фанлар массиви бор. Массивда имтиҳон топширилган фанлар тартиби массив элементи калити билан мос равишда берилган.

```
<?php
$dfan = array("Matematika", "Fizika", "Falsafa", "Informatika", "Tarix");
if (!array_search("Informatika", $dfan))
    echo "Informatika fanidan imtihon topshirilmagan.<br>";
else {
    $k = array_search("Informatika", $dfan);
    $n=$k+1;
    echo "Informatika imtihon topshirgan fanlarimning $n - si";
}
?>
Natija:Informatika imtihon topshirgan fanlarimning 4 - si
```

Бу функция *in_array()* функциясига нисбатан функционалроқ, чунки бу функция ёрдамида қидирилаётган элемент нафақат массивда мавжудлигини, балки у массивни қаерида жойлашгани ҳақида ҳам билишимиз мумкин. Агар қидирилаётган элемент массивда бир нечта бўлса, нима қиламиз? Бу ҳолатда *array_search()* функцияси биринчи топилган элемент калитини чиқазади. Барча элемент калитларини олиш учун эса *array_keys()* функциясидан фойдаланилади.

array_keys() функцияси

array_keys() функцияси массивнинг барча калитларини чиқазади. Қўшимча параметр сифатида берилган аниқ қиймат ёрдамида элемент калитлари рўйхатини олиш мумкин.

Синтаксиси:

array_keys(«массив», [«қидириш учун қиймат»])

array_keys() функцияси массивнинг сонли калитлари каби сатрли калитлари қийматларини ҳам чиқазади. Бунда қийматлардан ҳосил бўлган янги массив сонли индексли массив кўринишида бўлади.

10.11 - мисол. *array_keys()* функциясининг қўлланилиши:

Фанлар массиви мавжуд. Ушбу массив жуда кўп элементлардан ташкил топган шунинг учун рўйхатдаги баъзи фанлар тасодифан такрорланиши мумкин. Шу фанлардан бири – Fizika фани такрорлангани ҳақида шубҳа туғилди. Такрорланганми ёки йўқлигини текшириб кўрамиз.

```
<?php
$dfan = array("Matematika", "Fizika", "Falsafa", "Informatika", "Tarix", "Fizika");
$fan_keys = array_keys($dfan, "Fizika");
echo "Fizika massivda ". count($fan_keys) . " marta yozilgan.<br>";
foreach ($fan_keys as $val){
    echo " $val - o`rinda <br>";
}
?>
Natija:
Fizika massivda 2 marta yozilgan.
1 - o`rinda
5 - o`rinda
```

array_keys(), функцияси аввалги иккита функция каби регистрга сезгир, масалан FIZIKA элементини массивдан топмайди. *array_keys()* функцияси PHP4 версиясидан бошлаб қўшилган.

Массив элементларининг калитларини олиш учун ишлатиладиган функцияларни кўриб чиқдик, энди массив элементларининг қийматларини олиш учун ишлатиладиган функциялар билан танишамиз.

***array_values()** функцияси*

Синтаксиси:

array_values(массив);

Бу функция массивда берилган барча элемент қийматларини янги массивга бутун сонлар билан индекслаб ёзади, шунинг учун массив элементларининг барча калитлари йўқолади ва фақатгина қийматлари қолади.

***array_unique** функцияси*

Синтаксиси:

array_unique(массив);

Бу функция массивдан такрорланувчи қийматларни ўчириб, янги массив ҳосил қилади. Натижада, бир нечта бир хил қиймат ва уларнинг калитлари ўрнига битта қиймат қолади. Уни калити қандай бўлади? Янги массивга калит бир хил элементларнинг бир нечта калитлари ичидан қандай танлаб олинади? Массивнинг барча элементлари сатрларга акслантирилади ва сараланади. Шундан сўнг сараловчи ҳар бир қийматнинг биринчи калитини эслаб қолади, бошқа калитлар эса инкор этилади.

10.11 - мисолга қайтамиз. Биз биламизки, Fizika фани бизнинг массивда тасодифан икки марта қатнашиб қолди. Ростдан ҳам битта фан икки марта ўқилмайди, («ўқиди, лекин эсдан чиқарди» бу ҳисоб эмас), шунинг учун бу

камчиликни тўғрилаймиз. Ўтилган фанлар рўйхатидан такрорланганларини олиб ташлаймиз.

```
<?php
$dfan = array("Matematika", "Fizika", "Falsafa", "Matematika", "Informatika",
"Tarix", "Fizika", );
print_r(array_unique($dfan));
?>
Natija:Array ( [0] => Matematika [1] => Fizika [2] => Falsafa [4] =>
Informatika [5] => Tarix )
```

10.4. Массивларни саралаш

Турли хил масалаларни ечишда тез-тез массив кўринишида сақланаётган маълумотларни саралаш зарур бўлиб туради. Агар бу масалани Си да ечмоқчи бўлсак, у ҳолда ўнтагача код сатрлари ёзиш керак бўлади, PHP да эса бу битта оддий буйруқ билан амалга оширилади.

10.2 - жадвал. Саралаш функциялари.

Функция	Саралаш	Тескари тартиб	<калит/қиймат>жуфтлиги сақлаш
sort	Киймат	Йўқ	Йўқ
rsort	Киймат	Ҳа	Йўқ
asort	Киймат	Йўқ	Ҳа
arsort	Киймат	Ҳа	Ҳа
ksort	Калит	Йўқ	Ҳа
krsort	Калит	Ҳа	Ҳа
usort	Киймат	?	Йўқ
uasort	Киймат	?	Ҳа
uksort	Калит	?	Ҳа

sort функциясининг ишлатилиши

sort() функцияси массивни саралаб, унинг элементи қийматларини ўсиш тартиби (кичигидан каттасига қараб) бўйича тартиблайди. Бу функция массивнинг барча мавжуд калитларини ўчириб, уларни элементларнинг янги тартибига мос келувчи сонли индексга алмаштиради. Агар функция ишни муваффақиятли якунласа true ни, акс ҳолда – false ни қайтаради.

Синтаксиси:

sort (массив [, байроқ]);

Қўшимча параметр байроқ сифатида қуйидаги константалардан бирортаси ишлатилиши мумкин:

- *SORT_REGULAR* – массив элементларини оддий равишда таққослайди;
- *SORT_NUMERIC* – массив элементларини сон каби таққослайди;
- *SORT_STRING* – массив элементларини сатр каби таққослайди.

10.12 - мисол. **sort()** функциясининг қўлланилиши:

Бизда иккита массив мавжуд: фанлар номи – уларга ажратилган соат ва аксинча, фанларга ажратилган соат – уларнинг номи. Бу массивларни ўсиш тартиби бўйича тартиблаймиз.

```
<?
$soatlar= array("informatika" => 56, "fizika" => 34, "matematika" => 36);
sort($soatlar);
// sonlar o'sish tartibi bo'yicha saralanadi,
// kalitlar tushirib qoldiriladi
print_r($soatlar);
$fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika");
sort($fanlar);
// qiymatlar alifbo tartibida saralanadi
// kalitlar tushurub qoldiriladi
print_r($fanlar);
```

```
?>
Natija:Array ( [0] => 34 [1] => 36 [2] => 56 ) Array ( [0] => fizika [1] =>
informatika [2] => matematika )
```

***asort, rsort, arsort* функциялари**

Бу функциялар синтаксиси ҳам *sort()* функцияси каби бўлиб, вазифалари бир мунча фарқ қилади.

Массив элементларини калитлари сақланган ҳолда саралаш учун *asort()* функцияси ишлатилади.

Синтаксиси:

asort(массив [, байроқ]);

Агар массивни тескари тартибда, яъни катта қийматдан бошлаб кичигига қараб саралаш керак бўлса, у ҳолда *rsort()* функцияси ишлатилади.

Синтаксиси:

rsort(массив [, байроқ]);

Шу ҳолда, яъни тескари тартибда сараланган ҳолда яна калитларини ҳам сақлаш керак бўлса, *arsort()* функциясидан фойдаланилади.

Синтаксиси:

arsort(массив [, байроқ]);

Мос ҳолда байроқ қийматлари ҳам *sort()* функцияси каби бўлади: *SORT_REGULAR, SORT_NUMERIC, SORT_STRING*.

SORT_NUMERIC байроғи фақатгина PHP4 да қўшилган.

10.13 - мисол. *asort, rsort, arsort* функцияларининг қўлланилиши.

```
<?
$fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika");
asort($fanlar);
// kalit qiymatlari saqlangan holda,
// massivni saralaymiz
print_r($fanlar);
echo "<br>";
rsort($fanlar);
// teskari tartibda saralaymiz,
// kalitlar almashadi
print_r($fanlar);

?>
Natija:Array ( [34] => fizika [56] => informatika [36] => matematika )
Array ( [0] => matematika [1] => informatika [2] => fizika )
```

10.14 - мисол. Ўқув манбалари каталогини яратамиз. Ҳар бир манбани муаллифи, номи, чиқарилган санаси ва қисқача изоҳи мавжуд. Бу элементларни массив элементлари сифатида қараб, уларни тескари тартибда саралаймиз. Бунинг учун қуйидаги формани яратамиз ва ушбу саҳифани бирор ном(*insert_book.php*) билан сақлаймиз:

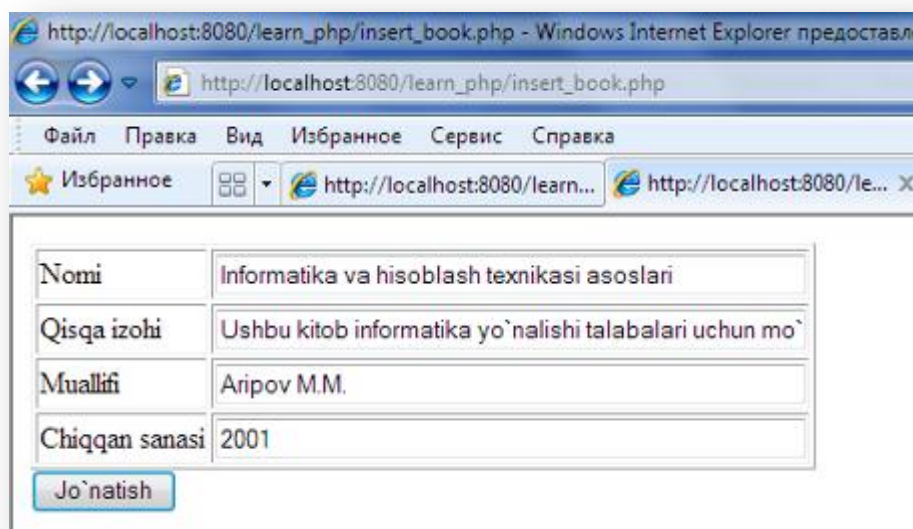
10.14a. *10.14 – мисол учун форма. insert_book.php файли*

```
<form action=test.php>
<table border=1>
  <tr>
    <td>Nomi </td>
    <td><input type=text name=nomi size=50> </td>
  </tr>
```

```

<tr>
<td>Qisqa izohi</td>
<td><input type=text name=izohi size=50> </td>
</tr>
<tr>
<td>Muallifi</td>
<td><input type=text name=muallifi size=50></td>
</tr>
<tr>
<td>Chiqqan sanasi</td>
<td><input type=text name=chiqqan_sana size=50></td>
</tr>
</table>
<input type=submit value="Jo`natish">
</form>

```



10.1 - расм. Ўқув манбаларини киритиш учун форма

Иккинчи саҳифада `insert_book.php` файлидаги формада берилган маълумотларни калитлари сақланган ҳолда кийматларини камайиш(ёки тескари алифбо) тартиби бўйича тартиблаймиз. Бунинг учун **`arsort()`** функциясидан фойдаланиш қулай. Бу функция ёрдамида берилган массивнинг калитлари сақланган ҳолда, янги тартибдаги массивни ҳосил қилиш мумкин. Иккинчи саҳифани `test.php` деб номлаймиз ва қуйидаги скрипти ёзамиз, чунки олдинги форма маълумотлари `test.php` файлига жўнатилади.

10.146. 10.14 – мисол учун берилган формадаги маълумотларни қайта ишловчи дастур. `test.php` файли

```

<?php
print_r($_GET); echo "<br>";
arsort($_GET);
// kalitlari saqlangan holda
// massivni teskari tartibda saralaymiz.
print_r($_GET); echo "<br>";
$ordered_names = array_keys($_GET);
// yangi massiv yaratamiz
foreach($ordered_names as $key => $val)
echo "$key :$val <br>";
// yangi massiv elementlarini chiqazamiz

```


>

Натижа:**10.2 - расм. Киритилган ўқув манбаларини чиқазии****Массивни калити бўйича саралаш: ksort(), krsort()**

Баъзан массивни *калит қийматлари* бўйича саралаш зарурати туғилади. Мисол учун, агар бизда юқоридаги мисолда берилгани каби китоблар ҳақидаги маълумотлар массиви бор бўлса, у ҳолда китобларни муаллифи бўйича саралашимиз мумкин. Бунинг учун PHP да бир қанча сатр кодларни ёзиш шарт эмас, чунки PHP нинг бу вазифаларни бажарувчи функциялари мавжуд.

ksort() функцияси - массив калитларини калитнинг мос қийматлари сақланган ҳолда, ўсиш тартиби (тўғри тартиб) бўйича саралайди.

krsort() функцияси - массив калитларини калитнинг мос қийматлари сақланган ҳолда, камайиш тартиби (тескари тартиб) бўйича саралаш учун ишлатилади. Бу функцияларни синтаксиси ҳам **sort()** функцияси синтаксисига ўхшаш.

10.15 – мисол. Массивни калити бўйича саралаш

```
<?
$fanlar= array(56 => "informatika", 34 => "fizika", 36 => "matematika");
ksort($fanlar);
// kalit qiymatini saqlagan holda,
// massivni saralaymiz
print_r($fanlar);

?>
Natija:Array ( [34] => fizika [36] => matematika [56] => informatika )
```

10.5. Кўп ўлчовли массивларни саралаш

Массивларни саралашда, битта ўзгартириш киритиш ёрдамида уларнинг тартибини алифбо ёки рақамли тартибга ўтказиш мумкин. PHP да иккита сон ёки иккита матнли сатрни таққослаш имконияти мавжуд, бироқ кўп ўлчовли массивларнинг элементлари ҳам массивдан иборат. PHP нинг иккита массивни таққослаш имконияти мавжуд эмас, шунинг учун уларни таққослашда методлар яратиш зарур.

Массивларни фойдаланувчи томонидан аниқланган функция ёрдамида саралаш: *usort()*, *uksort()*, *uasort()*

Массив қийматларини саралашни оддий иккита йўли(камайтиш ёки ўсиш тартиби) дан ташқари PHP да фойдаланувчи томонидан берилган бирор бир критерия(мезон) ёрдамида ҳам саралаш мумкин. Бу критериялар(мезонлар) функция ёрдамида берилиб, функция номи махсус *usort()* ёки *uksort()* саралаш функцияларида параметр сифатида кўрсатилади. Ушбу махсус функциялар фойдаланувчи томонидан аниқланган функциялар ёрдамида мос ҳолда *usort()* массив элементлари қийматларини, *uksort()* эса – массив калити қийматларини саралайди. Бу иккала функция ҳам агар саралаш ижобий яқун топса, true ни, акс ҳолда false ни қайтаради.

Синтаксиси:

usort(массив, сараловчи_функция)

uksort(массив, сараловчи_функция)

Албатта, массивларни фойдаланувчи томонидан аниқланган функция ёрдамида саралаб бўлмайди. Бу функция массив элементларини таққослаш имконини берувчи аниқланган критерияларни қониқтириши керак. Сараловчи функция қандай яратилган бўлиши керак? Биринчидан, иккита параметрга эга бўлиши керак. Уларга интерпретатор *usort()* функцияси учун элемент қиймати ёки *uksort()* функцияси учун массив калити жуфтлигини бериши мумкин. Иккинчидан, сараловчи функция қуйидагилардан бирортасини қайтариши керак:

- нолдан кичик бутун сонни, агар биринчи параметр иккинчисидан кичик бўлса;

- нолга тенг сонни, агар иккала параметр тенг бўлса;

- нолдан катта сонни, агар биринчи параметр иккинчисидан катта бўлса.

Бошқа саралаш функциялари каби *usort()* функциясининг ҳам ўхшаши мавжуд, калит қиймати ўзгармайдиган *uasort()* функцияси.

Қуйида аввал кўрилган массив берилган. Бу массивда талабалар ҳақидаги маълумотлар, уларнинг коди, фамилияси, исми, стипендияси, курси, манзили, туғилган санаси, университети берилган.

10.16 – мисол. Кўп ўлчовли массивни фойдаланувчи томонидан аниқланган функция ёрдамида саралаш

```
<?
$talabalar=array(
    array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
    array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
    array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamDU"),
    array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
);
```

```
?>
```

Бу кўп ўлчовли массивни сараласак, унинг қийматлари қандай тартибда сараланади? Бизга маълумки, массив таркибини иккита тартибда саралаш мумкин. Сатрли маълумотлар(фамилияси, исми, манзили, университети)ни алифбо тартибида, сонлилари(коди, стипендияси, курси)ни рақамли тартибда сараланади. Бунинг учун `usort()` функцияси ва элементларни қандай таққослаш кераклигини кўрсатувчи фойдаланувчи томонидан аниқланган функция ишлатилади.

Қуйидаги код бу массивнинг учинчи устуни(исми) қийматларини алифбо тартибида саралайди.

```
function ftaf($x, $y)
{
if ( $x[2] == $y[2] )
return 0;
else if ( $x[2] < $y[2] )
return -1;
else
return 1;
}
usort($talabalar, ftaf);
```

Фойдаланувчи томонидан аниқланган функциялар қандай яратилишини аввалги мавзуларда кўриб ўтганмиз. Бу мисолда фойдаланувчи томонидан аниқланган функция - `ftaf()`. `ftaf()` функцияси параметр сифатида иккита қийматни қабул қилади: **x** ва **y**. `ftaf()` функциясининг вазифаси ушбу иккита қийматни қабул қилиб, уларнинг тартибини аниқлаш.

Қаралаётган мисолда **x** ва **y** параметрлари ҳар бири алоҳида талабанинг исмини билдирувчи асосий массив ичидаги массивлар. **x** массивининг **name(исми)** элементига мурожаат этиш учун `x[2]` деб ёзиш керак. Чунки массив нолдан бошлаб индексланганлиги учун **name(исми)** – иккинчи элемент. Массивдаги **name(исми)** элементини таққослаш учун функцияга узатиладиган `$x[2]` ва `$y[2]` ўзгарувчилари ишлатилади.

Функция ўз ишини тугатгандан сўнг жавоб сифатида қиймат қайтаради. Функция қийматларини қайтариш учун **return** калит сўзи ишлатилади. Масалан, **return 1;** сатри функцияга мурожаат этганда, 1 кодини қайтаради. `ftaf()` функцияси **x** ва **y** ни таққослаш учун `usort()` функцияси билан ишлатилади. Агар **x** нинг қиймати **y** нинг қийматига тенг бўлса, 0 ни, кичик бўлса, манфий сонни, катта бўлса, мусбат сонни қайтаради. **x** ва **y** нинг қийматига боғлиқ равишда функция 0,1 ёки -1 ни қайтаради.

Код сатрининг охирида `usort()` функциясига саралаш керак бўлган массив(`$talabalar`) ва фойдаланувчи функцияси(`ftaf()`) билан мурожаат этилади.

Агар массивни бошқа тартибда саралаш лозим бўлса, бошқа функция яратилади. Масалан, талабаларнинг курслари бўйича саралаш лозим бўлса, бешинчи устун қаралади ва саралаш учун қуйидаги функция яратилади.

```
function ftaf($x, $y)
{
if ( $x[4] == $y[4] )
return 0;
```

```

else if ( $x[4] < $y[4] )
return -1;
else
return 1;
}
usort($talabalar, ftaf);

```

`usort($talabalar, ftaf);` функциясига мурожаат этгандан, сўнг массив курсларини ўсиш тартиби бўйича тартиблайди.

`usort()` функциясидаги *u* белгиси “user”(«фойдаланувчи») деган маънони англатади, шунинг учун бу функция фойдаланувчи функциясини талаб этади.

`uksort()` ва **`uasort()`** функциялари **`ksort()`** ва **`asort()`** ларга ўхшаш, фақатгина булар ҳам фойдаланувчи функциясини талаб этади.

`asort()`га ўхшаб **`uasort()`** функцияси ҳам ассоциатив массивларни қийматларини саралайди. Қийматлар оддий сонлар ва сатрлардан иборат бўлса, **`asort()`** функцияси ишлатилади. Агар қийматлар массив сингари мураккаб объектлардан ташкил топган бўлса, у ҳолда таққсолаш учун функцияни аниқлаш ва **`uasort()`** функциясини ишлатиш керак.

`sort()`, `asort()` ва `ksort()` функцияларининг уларга мос номида “г” белгиси иштирок этган тескари тартибда сараловчи ўхшаши мавжуд. Саралаш учун фойдаланувчи томонидан аниқланган функцияларнинг тескари версияси йўқ, бироқ кўп ўлчовли массивларни тескари тартибда саралаш мумкин. Модомики, таққсолаш функциялари дастурчилар томонидан яратилган экан, демак қарама-қарши қийматни қайтарувчи таққсолаш функцияларини ҳам яратиш мумкин. Саралаш тескари тартибда бажарилиши учун, агар *x* нинг қиймати *y* дан кичик бўлса 1 ва агар *x* нинг қиймати *y* дан катта бўлса -1 бўлиши керак.

10.17 – мисол. *Кўп ўлчовли массивни фойдаланувчи томонидан аниқланган функция ёрдамидатескари тартибда саралаш*

```

<?
$talabalar=array(
    array("1","Islomov", "Ikrom", 1899, 1, "Andijon", "3/12/1992", "TDPU"),
    array("3","Fozilov", "Fozil", 1870, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
    array("6","Salimov", "Vohid", 1880, 4, "Namangan", "7/06/1989", "NamDU"),
    array("10","Karimov", "Botir", 1885, 2, "Jizzax", "8/12/1991", "TDPU")
);

function aksftaf($x, $y)
{
if ( $x[4] == $y[4] )
return 0;
else if ( $x[4] < $y[4] )
return 1;
else
return -1;
}
usort($talabalar, aksftaf);

for ($i=0; $i<4; $i++) {
echo $talabalar[$i][1]." ".$talabalar[$i][4]."- kurs<br>";
}
?>

```

Natija:Salimov 4- kurs

Fozilov 3- kurs

Karimov 2- kurs

Islomov 1- kurs

Энди `usort($talabalar, aksftaf);` функцияга мурожаат этганда, массив курсларини камайиш тартибида тартиблайди.

10.18 - мисол. Фойдаланувчи томонидан аниқланган функция ёрдамида массивни саралаш:

Бизда манбалар ҳақидаги маълумотлар(номи, муаллифи ва чиқарилган йили) ни ўзида мужассамлаштирган массив мавжуд. Биз манбаларни чиқарилган санаси бўйича тартибламоқчимиз.

```
<?php
// massiv quyidagi ko'rinishda bo'ladi:
$books = array(
    "O'tgan kunlar" => array ("Qahhor", 1840),
    "Shaytanat" => array("Tohir Malik",1998),
    "Mehrobdan chayon" => array ("Qodiriy",1863),
    "Hamsa" => array("Navoiy",1868));
function cmp($x,$y){
    if ($x[1]==$y[1])
        return 0;
    elseif ($x[1] < $y[1])
        return -1;
    else
        return 1;
}
uasort($books,"cmp");
// cmp funksiyasi yordamida massivni saralaymiz
foreach ($books as $key => $book) {
    echo "$book[0]: \"$key\"<br>";
}

?>
```

Natija:
 Qahhor: "O'tgan kunlar"
 Qodiriy: "Mehrobdan chayon"
 Navoiy: "Hamsa"
 Ahmad: "Shaytanat"

10.6. Массивнинг ҳар бир элементи учун ихтиёрий функцияни қўллаш

Баъзан бир хил амални массивнинг барча элементлари учун қўллаш ёки уларни бир хил шаклда ўзгартириш талаб этилади. Бунинг учун `array_walk()` функциясидан фойдаланилади.

Синтаксиси:

`int array_walk(array массив, string функция [, mixed параметр])`

Кўриб турганингиздек, бу функция учта параметр қабул қилади. Биринчиси, *массив* – қайта ишланилиши лозим бўлган массив. Иккинчиси – *функция* – массивнинг ҳар бир элементини қабул қилувчи фойдаланувчи томонидан аниқланган *функция*. Учинчи *параметр*– зарур бўлмаган параметр бўлиб, у ишлатилганда фойдаланувчи томонидан аниқланган функцияга параметр сифатида берилади. Бу функция амаллар муваффақиятли якунланганда `true` ни, акс ҳолда `false` ни қайтаради.

Қоидага кўра, фойдаланувчи томонидан аниқланган функциянинг иккита параметри мавжуд бўлиб, бу икки параметрга навбат билан массивнинг ҳар бир элементининг қиймат ва калитлари берилади.

Агар `array_walk()` функциясига мурожаат этилганда, учинчи параметр кўрсатилса, у ҳолда бу параметр фойдаланувчи томонидан аниқланган функциянинг учинчи параметри сифатида қаралади. Агар фойдаланувчи томонидан аниқланган функция унга берилгандан кўра кўпроқ параметрларни талаб қилса, у ҳолда `array_walk()` га ҳар гал мурожаат этилганда огоҳлантириш ҳақида хабар чиқазилади.

Агар массив қийматларининг нусхаси билан эмас, балки асли билан ишлашга тўғри келинса, параметрни функцияга жўнатма(ссылка) сифатида узатиш керак бўлади. Бироқ, массивнинг элементларини ўчириш ёки қўшиш ва массив таркибини ўзгартирувчи ҳаракатларни амалга ошириш мумкин эмас, чунки бу ҳолда `array_walk()` ишининг натижаси ноаниқ бўлади.

10.19 - мисол. Массивнинг барча элементлари учун функцияни қўллаш

```
<?
$fanlar= array(56 => "informatika",
34 => "fizika",
36 => "matematika");
// massiv elementlariga qo'llash
// uchun funksiya yaratamiz
function fan_soat($val,$key,$data){
    echo "\"$val\" $data $key soat ajratilgan.<br>";
}
// $fanlar massivning barcha elementlari uchun
// fan_soat funksiyasining qo'llanilishi
array_walk($fanlar,"fan_soat","faniga");
?>
Natija: "informatika" faniga 56 soat ajratilgan.
"fizika" faniga 34 soat ajratilgan.
"matematika" faniga 36 soat ajratilgan.
```

Массив элементи қиймати ўзгартирилмаганлигини кўриш мумкин. Уларни ўзгартириш учун `fan_soat` функциясидаги `$val` ўзгарувчисига жўнатма(ссылка) орқали қийматни узатиш керак.

10.20 - мисол. Массивнинг барча элементлари учун функцияни қўллаш. 2–усул.

```
<?
```

```
$fanlar= array(56 => "informatika",  
34 => "fizika",  
36 => "matematika");  
// massiv elementlariga qo'llash  
// uchun funksiya yaratamiz  
print_r($fanlar);  
function fan_soat(&$val,$key){  
    $val= "<p>Fan nomi: \"\" . $val . \"\"<br>";  
    $key = "Ajratilgan soat: \"\" . $key . \"\"</p>";  
    echo $val.$key;  
}  
// $fanlar massivning barcha elementlari uchun  
// try_walk funksiya'sining qo'llanilishi  
array_walk($fanlar,"fan_soat");  
print_r($fanlar);  
?>
```

Natija: Array ([56] => informatika [34] => fizika [36] => matematika)

Fan nomi: "informatika"

Ajratilgan soat: 56

Fan nomi: "fizika"

Ajratilgan soat: 34

Fan nomi: "matematika"

Ajratilgan soat: 36

Array ([56] =>

Fan nomi: "informatika"

[34] =>

Fan nomi: "fizika"

[36] =>

Fan nomi: "matematika"

)

10.7. Қисм массивни белгилаш

array_slice() функцияси

Массив бу – элементлар тўплами. Ушбу тўпландан бирорта тўпламности ажратиб олиш учун PHP да *array_slice()* функциясидан фойдаланилади. Ажратиб олинган массив қисм массив дейилади.

Синтаксиси:

array_slice(массив, элемент_номери[, узунлиги])

Бу функция *элемент_номери* параметрида берилган номердаги массив элементидан бошлаб, *узунлиги* параметрига тенг бўлган узунликдаги *қисм массивни* берилган массивдан ажратиб олади. Мусбат *элемент_номери* элемент номерининг тартибини мос равишда массивнинг бошидан бошлаб кўрсатади, манфий *элемент_номери* эса массивнинг охиридан бошлаб ҳисоблайди.

10.21 – мисол. *array_slice()* функциясининг қўлланилиши

```
<?php
$arr = array(1,2,3,4,5);
$sub_arr = array_slice($arr,2);
print_r($sub_arr);
/*
Array ( [0] => 3 [1] =>4 [2] => 5 ),
3, 4, 5 elementlaridan tashkil topgan qism massivini chiqazadi */
$sub_arr = array_slice($arr,-2);
print_r($sub_arr);
// Array ( [0] => 4 [1] => 5 ) chiqazadi,
// 4, 5 elementlaridan tashkil topgan qism massivini chiqazadi
?>
Natija:Array ( [0] => 3 [1] => 4 [2] => 5 ) Array ( [0] => 4 [1] => 5 )
```

Агар *array_slice()* ни ишлатишда *узунлиги* параметри берилса, у ҳолда белгиланган қисм массив элементлари сони бу параметрда берилган сонга тенг бўлади. *узунлиги* параметри манфий сон сифатида ҳам кўрсатилиши мумкин. Бу ҳолда интерпретатор массив охиридан бошлаб, *узунлиги* параметри модулига тенг сондаги элементни ўчиради.

10.22 - мисол. *array_slice()* функциясининг қўлланилиши. 2-усул.

```
<?php
$arr = array(1,2,3,4,5);
$sub_arr = array_slice($arr, 2, 2);
// massiv 3, 4 elementlaridan tashkil topadi
$sub = array_slice($arr,-3, 2);
// massiv 3, 4 elementlaridan tashkil topadi
$sub1 = array_slice($arr,0, -1);
// massiv 1,2,3, 4 elementlaridan tashkil topadi
$sub2 = array_slice($arr,-4, -2);
// massiv 2,3 elementlaridan tashkil topadi
?>
```

array_chunk функцияси

array_chunk() функцияси *array_slice()* функциясига ўхшаш бўлиб, бу функция массивни берилган узунлигига қараб бир нечта қисм массивга ажратади.

Синтаксиси:

array_chunk(массив, ҳажми [, калитларни_сақлаш])

array_chunk() ишининг натижаси сифатида элементлари қисм массивлардан ташкил топган кўп ўлчовли массивни чиқазади. Агар *калитларни_сақлаш* параметри true сифатида берилса, у ҳолда берилган массивни ажратишда калитлари сақланади. Акс ҳолда элемент калитлари нолдан бошланувчи сонли индексларга алмашади.

10.23 - мисол. ***array_chunk()*** функциясининг қўлланилиши:

Талабалар фамилиясидан ташкил топган массив берилган. Ҳар бир гуруҳга учтадан талаба тўғри келади. Талабаларни гуруҳларга учтадан этиб тақсимлаш керак.

```
<?php
$talabalar = array("Ismoilov", "Fozilov",
    "Samadova", "Karimova", "Vohidov");
$ajrat= array_chunk($talabalar,3);
// massivni uchta elementli
// qism massivga ajratamiz
foreach ($ajrat as $k => $guruh){
echo "$k - nomerli guruhga: <ul>";
    foreach ($guruh as $talaba)
        echo "<li>$talaba";
    echo "</ul>";
}
?>
```

Natija:

0 – nomerli guruhga:

- Ismoilov
- Fozilov
- Samadova

1 – nomerli guruhga:

- Karimova
- Vohidov

10.8. Массив элементлари йиғиндисини ҳисоблаш

Массив элементлари қийматларининг йиғиндисини ҳисоблаш учун *array_sum()* функциясидан фойдаланилади. Бу функциянинг *параметри* сифатида эса элемент қийматлари йиғиндиси ҳисобланиши керак бўлган *массив* кўрсатилади.

Синтаксиси:

mixed array_sum(array массив);

Бу функциянинг ишлатилишига бир мунча қийинроқ мисол кўрамиз. Бу мисолда аввалроқ кўриб ўтган *array_slice()* функциясини ҳам ишлатамиз.

10.24 - мисол. Натурал сонлар массиви берилган. Бу массивдан шундай сон топиш керакки, шу элементнинг ўнг томондаги элементлари йиғиндиси, чап томондаги элементлари йиғиндисига тенг бўлиши керак.

```
<?php
$arr = array(2,1,3,4,5,6,4);
// Sikl ichida $k o'zgaruvchisi massivning joriy kalitidan,
// $v o'zgaruvchisi esa massivning joriy qiymatidan tashkil topgan
foreach ($arr as $k => $v) {
    $p = $k + 1;
    //array_slice (massiv, element_nomeri[, uzunligi])
    // array_slice funksiyasi uzunligi uzunligi parametriga teng
    // element_nomeri parametrida berilgan nomerli
    // elementdan boshlangan
    //qism massivni belgilaydi
    $out_next = array_slice($arr,$p);
    $out_prev = array_slice($arr,0,$k);

    // array_sum funksiyasi
    //berilgan massiv elementlari yig'indisini hisoblaydi
    $next_sum = array_sum($out_next);
    $prev_sum = array_sum($out_prev);

    if ($next_sum==$prev_sum)
        echo "qiymat:$v";
    // qaralayotgan massivning har bir qadamini ko'rish uchun
    // quyidagilarni aktivlashtiriladi
    // print_r($out_next); echo "<br>";
    // print_r($out_prev);
    // echo "$next sum, $prev sum<br>";
    // echo "<hr>";
}
?>
```

Natija:qiymat:5

10.9.Массив элементларини жойлашган тартибини ўзгартириш

Баъзи масалаларда массив элементлари тартибини турли усуллар билан ўзгартириш талаб этилади. **Shuffle()** функцияси массив элементларини тасодикий тартибда жойлаштиради. **array_reverse()** функцияси массив элементлари тескари тартибда жойлашган массив нусхасини қайтаради.

shuffle() функциясининг ишлатилиши

Талабалар порталининг асосий саҳифасига аълочи (энг кўп стипендия олувчи) талабаларнинг расмларини жойлаштирайлик. Бундай талабалар кўпчиликни ташкил этиши мумкин, лекин уларнинг ҳаммасини эмас, тасодикий тўртта аълочи талабанинг расми асосий саҳифада чиқиб турсин. Чунки, ҳар доим бир хил расмни чиқиб туриши портал фойдаланувчиларини зериктириб қўйиши мумкин. Агар барча расмлар массивда сақланса, бу масалани ҳал этиш осон.

Дастур **10.25** - листингда берилган. Ушбу дастур тўртта ихтиёрий танланган расмларни экранга чиқазади. Аввал массив элементлари ихтиёрий тарзда тартибланади, сўнг улардан биринчи тўрттаси тасвирланади.

10.25 - листинг. *for_n_page.php* – талабалар порталини асосий саҳифасини такомиллаштириш учун PHP нинг ишлатилиши.

```
<?
$pictures = array("upload/1.jpg", "upload/3.jpg", "upload/5.jpg",
"upload/10.jpg", "upload/16.jpg") ;
shuffle($pictures);
?>
<html>
<head>
<title>Talabalar portali</title>
</head>
<body>
<center>
<h1> Talabalar portali</h1>
<table width = 100%>
<tr>
<?
for ($i = 0; $i < 4; $i++ )
{
echo "<td align = center><img src=\"";
echo $pictures [ $i] ;
echo "\" width = 200 height = 250></td>" ;
}
?>
</tr>
</table>
</center>
</body>
```



10.4 -расм. *Shuffle()* функцияси тўртта ихтиёрий танланган расмни тасвирлашга имкон яратади.

***array_reverse()* функциясини ишлатилиши**

array_reverse() функцияси параметр сифатида массив қабул қилади ва шу массив элементларини тескари тартибда жойлаштириб янги массивни ҳосил қилади.

10.26 – мисол. 10 дан 1 гача тескари тартибда номерланган элементлардан ташкил топган массивни яратамиз.

range() функцияси кетма-кет ўсиб боровчи сонлар массивини яратади, сонларни камайиш тартибида саралаш учун **rsort()** функцияси ишлатилади. Ёки **for** циклини ишлатиб ҳам, биттадан элемент қўшиш ёрдамида массивни ҳосил қилиш мумкин:

```
<?
$numbers = array ();
for($i=10; $i>0; $i--)
array_push( $numbers,$i );
print_r($numbers);
?>
```

Natija:Array ([0] => 10 [1] => 9 [2] => 8 [3] => 7 [4] => 6 [5] => 5 [6] => 4 [7] => 3 [8] => 2 [9] => 1)

for() цикли ушбу мисолда кўрсатилганидек, бошқарувчи ўзгарувчи ёрдамида камайиш тартибини бажаради. Бошланғич қийматга энг катта қиймат ўрнатилади, сўнг ҳар бир цикл охирида ҳисоблагич қийматни -- амали ёрдамида биттага камайтиради. Аввал бўш массив яратилади, сўнг циклнинг ҳар бир элементига **array_push()** функциясини қўллаб, массивнинг охирига янги элемент қўшиб борилади ва натижада янги массив ҳосил бўлади. **array_push()** функциясига тескари функция бу - **array_pop()**. Бу функция элементни ўчириб, массив охиридан битта элементни қайтаради.

Худди шундай, **range()** функцияси яратган массив элементларини тартибини ўзгартириш учун **array_reverse()** функциясидан фойдаланилади.

```
<?
$numbers = range(1,10);
$numbers = array_reverse($numbers);
print_r($numbers);
?>
Natija:Array ( [0] => 10 [1] => 9 [2] => 8 [3] => 7 [4] => 6 [5] => 5 [6] => 4 [7] => 3 [8] => 2
[9] => 1 )
```

Эътибор беринг, **array_reverse()** функцияси ўзгартирилган массивнинг нусхасини қайтаради. Янги нусха эса эскисини устига сақланади.

10.10. Маълумотларни файлдан массивга юклаб олиш

10.2 – масала. Аввалги мавзуларимизда талабалар ҳақидаги маълумотларни файлда сақлаб қўйгандик. Файлнинг ҳар бир сатри куйидагича кўринишда эди:

1 Islomov Ikrom 150 1 Andijon 3/12/1992 TDPU

Талабалар маълумотларини қайта ишлаш учун уни қайта массивга юклаб олиш мумкин. **10.27 - листинг**да берилган саҳифа жорий файлни таркибини чиқазади.

10.27 - листинг. *view.php. Файл таркибини чиқариш учун PHPнинг ишлатилиши*

```
<? $talaba= file("student.txt") ;
// nmassivdagi talabalar ma'lumotlarini sonini hisoblash
$number_of_talaba = count($talaba);
if ($number_of_talaba==0)
{
echo "<p><strong>Talaba ma'lumotlari kiritilmagan.
Yana urinib ko`ring.</strong></p>";
}
for($i=0;$i<$number_of_talaba;$i++)
{
echo $talaba[$i] . "<br>" ;
}
?>
```

Natija:

1 Islomov Ikrom 150 1 Andijon 3/12/1992 TDPU
 3 Fozilov Fozil 200 3 Qashqadaryo 1/12/1990 TDPU
 6 Salimov Vohid 150 4 Namangan 7/06/1989 NamDU
 10 Karimov Botir 100 2 Jizzax 8/12/1991 TDPU

Бу саҳифа аввалги маърузаларда яратилган 9.14 листингда кўрсатилган натижани чиқазади. Битта фарқи файлнинг барчасини массивга юклаб олиш учун file() функцияси ишлатилади. Файлнинг ҳар бир сатри массивнинг алоҳида элементи бўлади. Бу саҳифада ҳам массив элементлари сонини аниқлаш учун count() функцияси ишлатилади.

Бундан ташқари талабалар маълумотлари сатри бўлимларини кейинчалик қайта ишлаш учун алоҳида массивнинг элементига юклаб олиш мумкин. Бу вазифаларни бажариш учун **10.28 - листинг**да дастур коди берилган.

10.28 – листинг. *vieworders.php файли.*

```
<html>
<head>
<title>Talabalar portali-talabalar haqidagi ma'lumotlar</title>
</head>
<body>
<h1>Talabalar portali</h1>
<h2>Talabalar haqidagi ma'lumotlar</h2>
<?
//faylni hammasini o`qib olamiz
//Har bir talaba ma'lumotlari massiv elementlari bo`ladi
$talaba= file("student.txt") ;
// nmassivdagi talabalar ma'lumotlarini sonini hisoblash
$number_of_talaba = count($talaba);
if ($number_of_talaba==0)
```

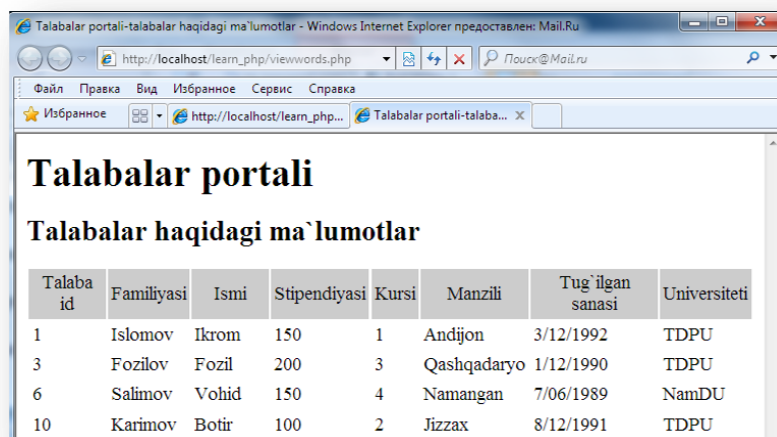


```

{
echo "<p><strong>Talaba ma'lumotlari kiritilmagan.
Yana urinib ko'ring.</strong></p>";
}
echo "<table border = 0 cellpadding = 3>";
echo "<tr><td bgcolor = \"#CCCCCC\" align = center>Talaba id</td>
      <td bgcolor = \"#CCCCCC\" align = center>Familiyasi</td>
      <td bgcolor = \"#CCCCCC\" align = center>Ismi</td>
      <td bgcolor = \"#CCCCCC\" align = center>Stipendiyasi</td>
      <td bgcolor = \"#CCCCCC\" align = center>Kursi</td>
      <td bgcolor = \"#CCCCCC\" align = center>Manzili</td>
      <td bgcolor = \"#CCCCCC\" align = center>Tug'ilgan sanasi</td>
      <td bgcolor = \"#CCCCCC\" align = center>Universiteti</td>
      <tr>";
for($i=0;$i<$number_of_talaba;$i++)
{
//har bir satrni ajratish
$result = explode( "\t", $talaba[$i] );
//string tipidagi ma'lumotlarni integer tipiga o'tkazish
$result[0] = intval( $result[0] );
$result[3] = intval( $result[3] );
$result[4] = intval( $result[4] );
//talabani har bir ma'lumotini chiqazish
echo "<tr><td>$result[0]</td>
      <td>$result[1]</td>
      <td>$result[2]</td>
      <td>$result[3]</td>
      <td>$result[4]</td>
      <td>$result[5]</td>
      <td>$result[6]</td>
      <td>$result[7]</td>
      <tr>";
}
echo "</table>";
?>

```

10.28 - листингда берилган дастур коди файлнинг барчасини массивга юклаб олади. **10.27 - листингдан** фаркли равишда бу листингда `explode()` функцияси ҳар бир сатрни бўлимларга(талабанинг ҳар бир маълумотини) кейинчалик улар устида ишлаш учун ажратади. Бу саҳифанинг натижаси **10.5 - расм**да кўрсатилган.



Talaba id	Familiyasi	Ismi	Stipendiyasi	Kursi	Manzili	Tug'ilgan sanasi	Universiteti
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	TDPU
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

10.5 - расм. Талаба маълумотларини файлдан массивга юклаб олиб тасвирлаш

Ушбу мисолнинг дастур кодини бироз бошқачароқ қилиб ҳам ёзишимиз мумкин:

10.29 – листинг. *vieworders2.php* файли.

```
<html>
<head>
<title>Talabalar portali-talabalar haqidagi ma`lumotlar</title>
</head>
<body>
<h1>Talabalar portali</h1>
<h2>Talabalar haqidagi ma`lumotlar</h2>
<table border = 0 cellpadding = 3>
  <tr>
    <td bgcolor = "#CCCCCC" align = center>Talaba id</td>
    <td bgcolor = "#CCCCCC" align = center>Familiyasi</td>
    <td bgcolor = "#CCCCCC" align = center>Ismi</td>
    <td bgcolor = "#CCCCCC" align = center>Stipendiyasi</td>
    <td bgcolor = "#CCCCCC" align = center>Kursi</td>
    <td bgcolor = "#CCCCCC" align = center>Manzili</td>
    <td bgcolor = "#CCCCCC" align = center>Tug`ilgan sanasi</td>
    <td bgcolor = "#CCCCCC" align = center>Universiteti</td>
  </tr>
  <?

$х = fopen("student.txt", "r");
while (!feof($х)) {
$хalaba = fgetcsv($х, filesize("student.txt"), "\t");
echo "<tr>";
//echo is_array($хalaba);
  while (list( $key, $value) = @each($хalaba) )
  {
    print "<td>".$value. "</td>";
  }
  echo "</tr>";
} ?>
</table>
</body>
</html>
```

10.11. Массивлар устида бошқа амалларни бажариш

Биз массивларни қайта ишловчи функциялар билан танишиб чиқдик. Бошқа массивлар билан ишлашда фойдали функцияларни ҳам кўриб ўтамиз.

Массив ичида кўчириш: *each*, *current()*, *reset()*, *end()*, *next()*, *pos()* ва *prev()*

Аввал айтиб ўтганимиздек, ҳар бир массив ичида массивнинг жорий элементини кўрсатиб турувчи ички кўрсаткичи бўлади. Биз ушбу кўрсаткични *each()* функцияни ишлатганимизда кўриб ўтдик, бироқ уни бевосита ишлатиш ва қўллаш мумкин.

Янги массивни яратишда жорий кўрсаткич массивнинг биринчи элементини кўрсатиш учун ўрнатилади. ***current(\$array_name)*** функциясига мурожаат этганда, биринчи элементни чиқазади.

next() ёки ***each()*** функциясига мурожаат этганда, кўрсаткич битта элемент олдинга кўчади. ***each(\$array_name)*** функциясига мурожаат этганда, кўрсаткич кўчмасдан аввал, жорий элементни қайтаради. ***Next()*** функцияси эса бироз бошқача, ***next(\$array_name)*** функциясига мурожаат этганда, кўрсаткични кўчириб, сўнг янги жорий элементни қайтаради.

reset() функцияси кўрсаткични массивнинг биринчи элементига қайтаради. Шунга ўхшаш, ***end(\$array_name)*** функциясига мурожаат этганда, кўрсаткич массивнинг охирига кўчади. ***reset()*** ва ***end()*** функциялари мос равишда массивнинг биринчи ва охириги элементларини қайтаради.

Массив ичида кўчиришни тескари йўналишда бажариш учун ***end()*** ва ***prev()*** функцияларидан фойдаланилади. ***prev()*** функцияси ***next()*** функциясига нисбатан тескари функция. У жорий кўрсаткични битта элемент орқага кўчиради, сўнг янги жорий элементни қайтаради.

Масалан, қуйидаги код массив элементларини тескари тартибда чиқазади:

10.30 – мисол.

```
<?
// $array массиви quyidagicha elon qilingan
$array = array (1, 2, 3) ;
$value = end ($array) ;
while ($value)
{
    echo "$value<br>";
    $value = prev($array);
}
?>
Natija:
3
2
1
```

each(), *current()*, *reset()*, *end()*, *next()*, *pos()* и *prev()* функциялари ишлатилганда массивда ихтиёрий тартибда ўзгартириш учун код яратиш мумкин.

Массивни скаляр ўзгарувчига алмаштириш: *extract()*

Калит-қиймат жуфтликларидан ташкил топган ассоциатив массивни **extract()** функцияси ёрдамида скаляр ўзгарувчилар тўпламига алмаштириш мумкин.

Синтаксиси:

extract (array var_array [, int extraot_type] [, string prefix]);

extract() функцияси скаляр ўзгарувчиларни яратишда уларнинг номларини массив калитлари номи билан номлайди. Ўзгарувчилар массивдаги мос қийматларга тенгланадилар.

10.31- мисол..

```
<?
$array = array( "key1" => "value1", "key2" => "value2", "key3" => "value3" );
extract($array);
echo "$key1 $key2 $key3";

?>
Natija: value1 value2 value3
```

Массив калитлари key1, key2 ва key3 дан иборат учта элементдан ташкил топган. **extract()** функцияси ишлатилганда учта \$key1, \$key2 ва \$key3 скаляр ўзгарувчилар яратилди. Натижада кўриб турганингиздек, \$key1, \$key2 ва \$key3 ўзгарувчиларнинг қийматлари мос равишда "value1", "value2" "value3" лар бўлади.

extract() функцияси иккита зарур бўлмаган параметрга эга: *extract_type* ва *prefix*. *extract_type* параметри **extract()** функциясига калит номи билан бир хил ўзгарувчилар мавжуд бўлганда юзага келадиган конфликтларни қайта ишлаш усулини беради. Одатга кўра мавжуд ўзгарувчининг қиймати янгисига алмаштирилади. *extract_type* параметрининг тўртта мумкин бўлган қийматлари жадвалда келтирилган.

Жадвал. **extract()** функциясининг *extract_type* параметрини мумкин бўлган қийматлари

Қиймати	Вазифаси
EXTR_OVERWRITE	Конфликт юзага келганда мавжуд ўзгарувчини қайта ёзади.
EXTR_SKIP	Конфликт юзага келганда элементни ташлаб ўтади(қолдириб кетади).
EXTR_PREFIX_SAME	Конфликт юзага келганда \$prefix_key ўзгарувчисини яратади. Бу ҳолда функцияда prefix параметри кўрсатилиши керак.
EXTR_PREFIX_ALL	Барча ўзгарувчиларга prefix олд қўшимчасини беради. Бу ҳолда функцияга бу параметр берилиши керак.

Одатга кўра кўп ишлатиладиган қийматлар: (EXTR_OVERWRITE) ва (EXTR_PREFIX_ALL). Қолган иккита қиймат аввалдан конфликт юзага

келадиган жойи маълум бўлса ва калит тушириб қолдирилган ва олд кўшимча ишлатилиши талаб этилаётган ҳолларда ишлатилади.

Қуйида EXTR_PREFIX_ALL нинг параметр сифатида ишлатилишига мисол кўрсатилган. Кўриб турганингиздек, яратилган ўзгарувчи *олд_қўшимча_калит_номи* номини олади.

10.32 – мисол.

```
<?
$array = array( "key1" => "value1", "key2" => "value2", "key3" => "value3");
extract($array, EXTR_PREFIX_ALL, "old qushimcha");
echo "$old_qushimcha_key1 $old_qushimcha_key2 $old_qushimcha_key3";
?>
```

Natija: value1 value2 value3

10.3 - Масала. Формани динамик яратиш

Ташриф буюрувчиларга юқори даражадаги дизайнга эга сайтлар рўйхатини тавсия этиш лозим бўлсин. Бунинг учун сайтлар рўйхатидан ташкил топган очилувчи рўйхатни яратамиз. Бу рўйхатдаги сайт номлари массивнинг элементлари бўлади.

10.33 - листинг. Очилувчи рўйхатни динамик яратиш (қуриш)

```
<? if (isset($site)) : header("Location: http://$site");
exit;
else :
?>
<html>
<head>
<title>Formani dinamik yaratish</title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74" vlink="#808040" alink="#808040">
<?
$favsites = array ("www.tdpu.uz",
"www.pedagog.uz",
"www.ziyonet.uz",
"www.nuu.uz",
"www.edu.uz");
// forma yaratamiz
?>
<form action = "learn.php" method="post">
<select name="site">
<option value = "">Saytni tanlang:
<?
$x = 0;
while ( $x < sizeof ($favsites) ) :
print "<option value='$favsites[$x]'>$favsites[$x]";
$x++;
endwhile;
?>
</select>

<input type="submit" value="go!">
</form>
</body>
</html>
<? endif;?>
```

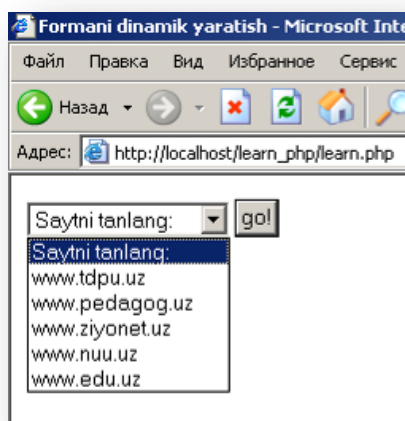
Аввал \$site ўзгарувчисининг қиймати бор-йўқлиги текширилади. Агар

текшириш ижобий натижани берса, PHP нинг махсус header() функцияси параметр сифатида берилган \$site ўзгарувчисининг қиймати билан чақирилади. header() функциясига мурожаат этилганда, функция браузерни \$site ўзгарувчисига кўрсатилган URL манзилига йўналтиради. Агар \$site ўзгарувчиси қиймати берилмаган бўлса, форма браузерга чиқазилади.

Очилувчи рўйхат циклда қурилади, итерациялар сони \$favsites массиви элементлари сонига боғлиқ бўлади.

header() функциясига саҳифанинг юқори қисмида мурожаат этиш керак. Унга PHP сценарийсининг ихтиёрий нуқтасида мурожаат этиш мумкин эмас. Ўз вақтида мурожаат этилмаган header() функцияси муаммо туғдириши мумкин.

Натижа



10.6-расм. Очилувчи рўйхатни динамик яратиш(қуриш). Танлангандан сўнг мос сайтга ўтилади.

Формани динамик яратиш айниқса катта ҳажмдаги маълумотларни қайта ишлашда, уларни ихтиёрий вақтда ўзгартиришга қулай.

ХИ БОБ. САТРЛАР ВА ДОИМИЙ ИФОДАЛАР (РЕГУЛЯРНЫЕ ВЫРАЖЕНИЕ)

Бу бобда сатрлар билан ишлаш, турли хил амалий масалаларни ечишда фойдали функцияларни ўрганиш тўлиқроқ муҳокама қилинади. Сатрларни чиқаришни турли усуллари, сатрларни бўлиш ва бирлаштириш(*explode*, *implode* функциялари), сатр узунлигини аниқлаш(*strlen*), қисм сатрларни белгилаш(*strstr*, *substr*) кўриб чиқилади.

11.1. Сатрлар билан ишлаш

Сатрли маълумотлар типи ва бундай типдаги ўзгарувчини қандай яратиш ҳақида олдинги маърузаларда айтиб ўтдик. Олдинги маърузаларда айтиб ўтганимиздек сатрли ўзгарувчилар учта усул: битталиқ кўштирноқ ёрдамида, иккиталиқ кўштирноқ ёрдамида ва *heredoc*—синтаксиси ёрдамида берилади.

Доимий ифодалар - шаблон бўйича қидиришнинг барча замонавий технологияларини асоси ҳисобланади. Доимий ифодалар қидирилаётган матнда ёзилган оддий ва хизматчи белгиларни кетма-кетлигини сақлаган ҳолда тасвирлайди. Баъзан доимий ифодалар оддий ва тушунарли(масалан, *dog сўзи*) бўлади, лекин кўпинча уларнинг синтаксисида алоҳида мазмунга эга бўлган хизматчи белгилар қатнашади, масалан`<(?)>.*<\/.?.>`.

Доимий ифодалар билан ишловчи функциялардан ташқари РНР нинг сатрлар устида мантикий амалларни бажарувчи 70 дан ортиқ функциялари мавжуд.

Катта ҳажмдаги маълумотларни қайта ишлашда, доимий ифодалар билан ишловчи функцияларнинг ишлаш тезлиги дастур бажарилишида пасаяди. Бу функцияларни доимий ифодалар ҳақиқатда зарур бўлган, нисбатан мураккаброқ сатрларни фақатгина қайта ишлашда қўллаш керак бўлади. Агар матнни таҳлил этиш нисбатан оддий қоидага кўра бажарилса, қайта ишлашни сезиларли даражада тезлаштирадиган РНР нинг стандарт функцияларидан фойдаланиш мумкин.

11.1 - Масала. Фойдаланувчи маълумотларини берилган электрон почта манзилига автоматик жўнатиш

Маълумотларни қайта ишлашнинг яна бир усулида маълумотлар электрон почта манзилига масалан, сайт администраторига автоматик равишда жўнатилади. *mailto* гипермуурожаати ёрдамида хабарни браузернинг ўзидан ҳам жўнатиш мумкин.

Қуйидаги мисолда, фойдаланувчи Талабалар портали ҳақидаги фикр ва мулоҳазаларини жўнатиш мумкин бўлган форма яратилади. Бу форма порталнинг **Бизга ёзинг!** саҳифасида яратилади. Маълумотлар мос шаклда форматланиб, РНР нинг стандарт *mail()* функцияси ёрдамида жўнатилади.

11.1 – листинг.

```
<html>
<head>
<title>Bizga yozing!</title>
</head>
```



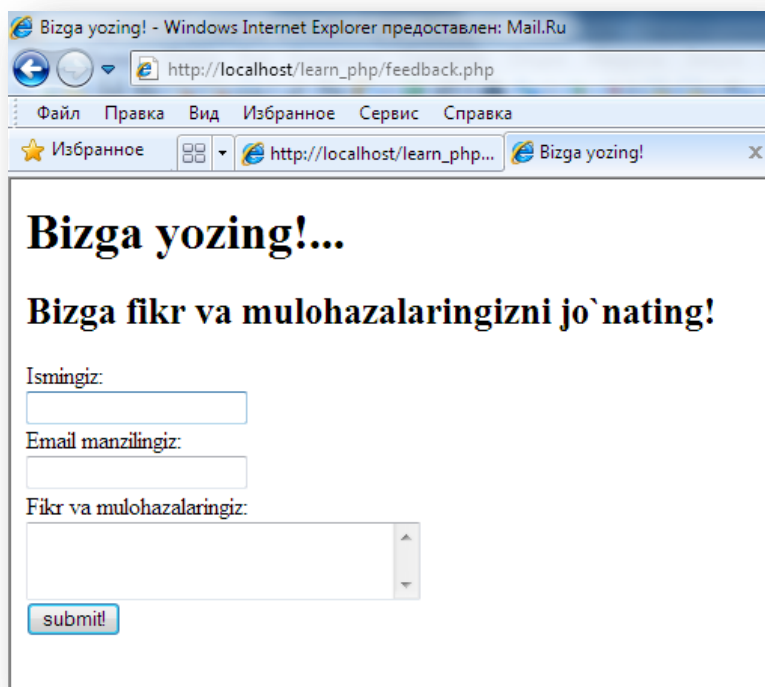
```

<body bgcolor="#ffffff" text="#000000" link="#cbda74" vlink="#808040" alink="
#808040">
<?
// $form ichidagi qo'shtirnoqda joylashganlar hammasi ekranga chiqishi kerak
// aks holda xatolik yuzaga keladi.
$form = "
<form action=\"feedback.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<h1>Bizga yozing!...</h1>
<h2>Bizga fikr va mulohazalaringizni jo'nating!</h2>
Ismingiz:<br>
<input type=\"text\" name=\"name\" size=\"20\" max-
length=\"20\" value=\"\"><br>
Email manzilingiz:<br>
<input type=\"text\" name=\"email\" size=\"20\" maxlength=\"40\" value=\"\"><
br>
Fikr va mulohazalaringiz:<br>
<textarea name=\"comments\" rows=\"3\" cols=\"30\"></textarea><br>
<input type=\"submit\" value=\"submit!\">
</form>;
// agar forma avval chiqazilmagan bo'lsa, uni chiqazishni
// tekshirish uchun $seenform yashirin o'zgaruvchisining qiymati ishlatiladi

if (!isset($seenform)) : print "$form"; else :
// $oluvchi o'zgaruvchisi forma ma'lumotlarini oluvchisini aniqlaydi
$oluvchi = "admin@mail.ru";
// Xabar mavzusi
$mavzu = "($name) ismli foydalanuvchining xabari";
// Qo'shimcha sarlavha
$sarlavha = "From: $email";
mail($recipient, $mavzu, $xabar, $sarlavha) or die("Xabarlaringiz jo'natilmadi
!");
print "Xabarlaringiz jo'natildi!";
endif;?>

```

Натижа



11.1 – расм. Маълумотларни электрон почта манзилига юбориши

Шу каби масалаларда форма маълумотларини жўнатишдан аввал форма майдонлари тўлдирилганлигини текшириш лозим.

Фойдаланувчи томонидан формага маълумот киритилганда ортиқча бўш белгилар(пробеллар), ортиқча теглар ва бошқалар ишлатилиши мумкин. Бу ҳолатда форма майдонлари ёрдамида киритилган сатрларни таҳрирлаш ва ўзгартириш керак бўлади. Бунда бизга РНР нинг сатрлар билан ишловчи функциялари ёрдам беради.

11.2. Сатрларни таҳрирлаш

Сатрни қисқартириш ва кенгайтириш: *trim()*, *ltrim()*, *chop()*, *str_pad()*

Форматлаш жараёнида, сатрга белгиларни қўшиш ва ўчириш натижасида тез-тез сатр узунлигини ўзгартиришга тўғри келади. РНР да бундай масалаларни ҳал этиш учун бир нечта функциялари мавжуд.

***trim()* функцияси**

trim() функцияси сатрни иккала томонидаги барча бўш жойларни ўчиради ва ҳосил бўлган сатрни чиқазади. Ушбу функция бўш жойлардан ташқари \n, \r, \t, \v и \0 махсус белгиларни ҳам ўчиради.

Синтаксиси:

string trim(string camp)

***ltrim()* функцияси**

ltrim() функцияси барча бўш жойларни ва махсус белгиларни сатрнинг ўнг томонидан ўчиради ва ҳосил бўлган сатрни чиқазади. Бу функция ҳам бўш жойлардан ташқари trim() функцияси каби \n, \r, \t, \v и \0 махсус белгиларни ўчиради.

Синтаксиси:

string ltrim (string camp)

***chop()* функцияси**

chop() функцияси берилган сатрдан яқунловчи белгилар ва янги қатор белгиларини ўчиради.

Синтаксиси:

string chop(string camp)

11.2 - мисол. Сатрдан ортиқча янги қатор белгилари(\n)ни **chop()** функцияси ёрдамида ўчириш.

```
<?
$header = "Table of Contents\n\n";
$header = chop($header);
// $header = "Table of Contents"
echo $header;
?>
Natija:Table of Contents
```

***str_pad()* функцияси**

str_pad() функцияси берилган сатрни кўрсатилган белгилар кетма-кетлиги билан тўлдиради.

Синтаксиси:

string str_pad (string camp, int узунлиги [, string қўшимча [, int қўшимча_мини]])

бу ерда *сатр* параметрида берилган сатр, *узунлиги* параметрида эса ҳосил бўладиган сатр узунлиги кўрсатилади. Агар «қўшимча» параметри кўрсатилмаган бўлса, сатр пробеллар билан тўлдирилади, акс ҳолда сатр берилган белгилар билан тўлдирилади. Одатга кўра сатрнинг ўнг томони тўлдирилади; сатрни кўрсатилган йўналиш бўйича тўлдириш учун «қўшимча_типи» параметрида STR_PAD_RIGHT(ўнг), STR_PAD_LEFT(чап) ёки STR_PAD_BOTH(иккала тарафи) константаларини кўрсатиш керак.

11.3 - мисол. str_pad() функциясини қўллаб сатрни тўлдириш.

```
<?
$food = "osh";
$a="paz" ;
print str_pad($food, 6, $a); // "oshpaz" satrini chiqazadi
?>
```

Natija:oshpaz

11.4 - мисол. str_pad() функциясининг зарур бўлмаган параметрларининг қўлланилиши

```
<?
$header = "Table of Contents";
print str_pad($header, 27, "+++", STR_PAD_BOTH);
// +++= Table of Contents=+++ satrini chiqazadi
?>
```

Natija:+++=Table of Contents=+++=

Сатрларни турли усуллар билан қайта таҳрирлаш учун ишлатиш мумкин бўлган РНР нинг функциялар тўплами мавжуд.

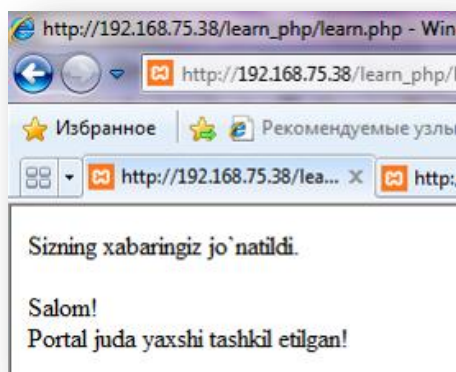
HTML-таҳрирлашнинг ишлатилиши: nl2br()

nl2br() функцияси параметр сифатида сатрларни қабул қилади ва ундаги барча янги сатр белгиларини
 HTML дескрипторига алмаштиради. Бу узун сатрларни браузерга чиқазишда фойдали.

Масалан, фойдаланувчи хабарини таҳрирлаш учун экранга чиқазамиз:

```
<p>Sizning xabaringiz jo`natildi.</p>
<p><? echo nl2br ($comments) ; ?> </p>
```

Натижа расмда кўрсатилган.



11.2 – расм. HTML-таҳрирлашнинг ишлатилиши

Сатрларни наирга бериш учун таҳрирлаш

Биринчи маърузадан бошлаб *echo* функциясини бир неча бор ишлатдик. *echo* – бу функция эмас, балки конструкция шунинг учун унга мурожаат

этишда юмалоқ қавс ишлатиш шарт эмас. **echo** экранга параметрлар билан берилган сатрларни чиқазишда ёрдам беради. **echo** параметрларини турлича: вергул билан ажратиб ёзиш ёки конкатенация амали билан бирлаштириш мумкин, лекин қавс ичига олиб ёзиш мумкин эмас.

11.5 – мисол. echo функциясининг ишлатилиши

```
<?
echo "O`yla", "izla", "top";
// "O`yla izla top"
// echo ni bir necha parametrlarini konkatentsiya
// amali yordamida berish mumkin
echo "O`yla " . "izla " . "top ";
// bu ham " O`yla izla top" yozuvini chiqazadi
echo ("O`yla ", "izla ", "top ");
// unexpected xatoligini chiqazadi
?>
Natija:
Parse      error:      parse      error,      unexpected      ', '      in
D:\xampplite\htdocs\learn_php\learn.php on line 8
```

echo буйруғининг қисқартирилган **синтаксиси** мавжуд:

<?=чиқазиш_учун_сатр?>

Бу ерда, **чиқазиш_учун_сатр** параметри экранга чиқарилиши лозим бўлган ихтиёрий усулда берилган сатрни ўз ичига олади.

Мисол учун, куйидаги скрипт экранга қизил ёзув билан «Mening ismim Lola» сатрини чиқазади:

```
<? $name="Lola"?>
<font color=red> Mening ismim <?=$name?></font>
Natija: Mening ismim Lola
```

Сатрларни экранга чиқазиш учун **echo** конструкциясидан ташқари бир қанча функциялар мавжуд. Улардан бири **print()** функцияси ва уни бошқа хил турлари **printf**, **sprintf** ва ҳ.

Биз шу вақтгача сатрларни браузер ойнасига чиқазиш учун **echo** конструкциясидан фойдаландик. РНР ушбу вазифани бажарувчи **print()** функциясини ҳам ишлатади, бироқ унинг битта фарқли тарафи натижа қийматларини (бажариш натижасига қараб 0 ёки 1 ни) қайтаради.

Ушбу иккала конструкция сатр қандай бўлса, ўша ҳолда экранга чиқазади. **printf()** ва **sprintf()** функциялари ёрдамида эса бирмунча қийинроқ таҳрирлаш ишларини олиб бориш мумкин. Бу иккала функция амалда асосан бир хил ишлайди, лекин **printf()** таҳрирланган сатрни браузер ойнасига чиқазади, **sprintf()** эса таҳрирланган сатрни қайтаради.

Ушбу функцияларнинг **синтаксиси**:

string sprintf (string format [, mixed args ...])

int printf (string format [, mixed args ...])

ушбу иккала функцияда биринчи параметр сифатида таҳрирлаш учун сатр берилади, бу ерда агар сатр ўзгарувчилардан иборат бўлса, ўзгарувчилар ўрнига таҳрирланадиган код ишлатилади. Қолган параметрлар эса сатрни таҳрирлашда кодлар ўрнига қўйиладиган ўзгарувчилардан иборат бўлади.

11.6 – мисол. echo конструкциясида керакли сатрни чиқазувчи ўзгарувчини куйидагича ишлатдик:

```
<?
```

```
$sum="2 so`m 50 tiyin";
echo "Daftar $sum turadi.";
?>
```

худди шундай натижани **printf()** функцияси ёрдамида олиш учун қуйидагича ўзгартириш киритилади:

```
<?
$sum="2 so`m 50 tiyin";
printf ("Daftar %s turadi.", $sum);
?>
```

Natija:Daftar 2 so`m 50 tiyin turadi.

Сатрни таҳрирлашда `%s` кетма-кетлиги *алмаштириш таснифи* деб юритилади.

Берилган тасниф "сатрни алмаштириш" маъносини англатади. Ушбу ҳолда бу кетма-кетлик сатр сифатида берилган `$sum` ўзгарувчисининг қийматига алмашади.

Агар `$sum` ўзгарувчисида сақланаётган қиймат 2.5 бўлса, иккала берилган ифода натижага 2.5 ни чиқазади.

printf() функциясида ҳақиқий сонларни тасвирлаш учун бошқачарок алмаштириш таснифини ишлатиш мумкин:

```
<?
$sum=2.5;
printf ("Daftar %.2f turadi.", $sum);
?>
```

Natija:Daftar 2.50 turadi.

Сатрларни таҳрирлаш учун ишлатиладиган бир қанча алмаштириш таснифлари мавжуд. Сатрларни таҳрирлашда ***n*** та алмаштириш таснифлари ишлатилганда, ***n*** та параметрлар кўрсатилиши зарур. Ҳар бир алмаштириш таснифи ўрнига берилган тартиби бўйича параметрлар қўйиб борилади.

11.7 - мисол:

```
<?
$sum=2.5;
$um_sum=3.8;
printf("Daftar %.2f(umumiy daftar %.2f) turadi.", $sum, $um_sum);
?>
```

Natija:Daftar 2.50(umumiy daftar 3.80) turadi.

Ушбу ҳолда биринчи алмаштириш таснифи **\$sum** ўзгарувчиси учун, иккинчиси эса — **\$um_sum** ўзгарувчиси учун ишлатилди.

Барча алмаштириш таснифлари бир хил форматга эга:

%['қўшимча_белги'][-][кенглиги][аниқлиги]мини

Барча алмаштириш таснифлари `%` белгиси билан бошланади. Агар `%` белгисини чиқазиш талаб этилса, `%%` кетма-кетлиги қўлланилади.

қўшимча_белги шарт бўлмаган параметр. Бу параметр кўрсатилган кенгликкача ўзгарувчиларни қўшиш мумкинлигини билдиради. Ўзгарувчилар нолдан бошлаб ушбу параметрда кўрсатилган сонгача ҳисоблагич қиймати сифатида қўшиб борилиши мумкин.

"-" белгиси ҳам шарт бўлмаган параметр. Бу параметр маълумотлар ўнг тарафдан эмас, чап тарафдан бошлаб тўғриланиши учун кўрсатилади.

кенглиги параметри **printf()** функциясида ўзгарувчи қийматларини алмаштириш учун қанча жой(белги) зарурлигини кўрсатиш учун ишлатилади.

аниқлиги параметри ўнлик нуқтадан бошланиши зарур. Бу параметр вергулдан сўнг тасвирланадиган ўнлик белгиларнинг миқдорини аниқлайди.

Таснифлашнинг охириги параметри код типини билдиради.

Ишлатиладиган кодларнинг қисқача таснифи **11.1 - жадвалда** берилган.

11.1 – жадвал. Алмаштириш таснифлари типи кодлари

Типи	Вазифаси
b	Бутун сон сифатида шарҳланади ва иккилик санок системасида чиқазилади
c	Бутун сон сифатида шарҳланади ва белги сифатида чиқазилади
d	Бутун сон сифатида шарҳланади ва ўнлик санок системасида чиқазилади
f	Икки карра аниқликдаги сон сифатида шарҳланади ва ўнли каср кўринишида чиқазилади
o	Бутун сон сифатида шарҳланади ва саккизлик санок системасида чиқазилади
s	Сатр сифатида шарҳланади ва сатр сифатида чиқазилади
x	Бутун сон сифатида шарҳланади ва a-f кичик ҳарфлари билан ифодаланган ўн олтилик санок системасида чиқазилади
X	Бутун сон сифатида шарҳланади ва A-F катта ҳарфлари билан ифодаланган ўн олтилик санок системасида чиқазилади

Сатр регистрини(катта ва кичик белгиларни) алмаштириш

Сатр регистрини ўзгартириш учун PHP да тўртта функция мавжуд:

- `strtolower()`;
- `strtoupper()`;
- `ucfirst()`;
- `ucwords()`.

strtolower() функцияси

strtolower() функцияси сатрдаги барча алифбо белгиларини кичигига (кичик регистрга) алмаштиради.

Синтаксиси:

string strtolower(string camp)

Алифбо таркибида бўлмаган белгиларни функция алмаштирмайди.

11.8 – мисол. strtolower() функциясининг қўлланилиши.

```
<?php
$gap = "TENNIS O`YNASH va PHPDA DASTURLASH mening sevimli mashg`ulotim!";
$gap = strtolower($gap);
Echo $gap;
?>
Natija:tennis o`ynash va phpda dasturlash mening sevimli mashg`ulotim!
```

Сатр белгиларини нафақат кичигига, балки каттасига ҳам алмаштириш мумкин.

strtoupper() функцияси

Сатр белгиларини каттасига алмаштириш **strtoupper()** функцияси ёрдамида амалга оширилади.

Синтаксиси:

string strtoupper (string camp)

Алифбо таркибида бўлмаган белгиларни функция алмаштирмайди.

11.9 – мисол. strtoupper() ёрдамида сатр белгиларини каттасига алмаштиришни кўриб ўтамиз.

```
<?php
$gap = "tennis o`ynash va PHPda dasturlash mening sevimli mashg`ulotim!";
$gap = strtoupper($gap);
echo $gap;
// funksiya ishlatilgandan so`ng $gap satri
/* « TENNIS O`YNASH VA PHPDA DASTURLASH MENING SEVIMLI MASHG`ULOTIM!» */
?>
Natija:TENNIS O`YNASH VA PHPDA DASTURLASH MENING SEVIMLI
MASHG`ULOTIM!
```

ucfirst() функцияси

ucfirst() функцияси шартга кўра сатрдаги биринчи белгини, каттасига алмаштиради.

Синтаксиси:

string ucfirst (string camp)

Алифбо таркибида бўлмаган белгиларни функция алмаштирмайди.

11.10 – мисол. ucfirst() функциясини ишлатилиши

```
<?php
$gap = "tennis o`ynash va PHPda dasturlash mening sevimli mashg`ulotim!";
$gap = ucfirst($gap);
echo $gap;
// funksiya ishlatilgandan so`ng $gap satri
/* « Tennis o`ynash va PHPda dasturlash mening sevimli mashg`ulotim!» */
?>
Natija:Tennis o`ynash va PHPda dasturlash mening sevimli mashg`ulotim!
```

ucwords() функцияси

ucwords() функцияси сатр таркибидаги ҳар бир сўзнинг биринчи белгисини каттасига алмаштиради.

Синтаксиси:

string ucwords (string camp)

Алифбо таркибида бўлмаган белгиларни функция алмаштирмайди. «Сўз» сатрнинг бошқа элементларидан пробел билан ажратилган белгилар кетма-кетлиги сифатида аниқланади.

11.11 – мисол. ucwords() функциянинг ишлатилиши

```
<?php
$gap = "tennis o`ynash va PHPda dasturlash mening sevimli mashg`ulotim!";
$gap = ucwords($gap);
echo $gap;
// funksiya ishlatilgandan so`ng $gap satri
/* « Tennis O`ynash Va PHPda Dasturlash Mening Sevimli Mashg`ulotim!» */
?>
Natija:Tennis O`ynash Va PHPda Dasturlash Mening Sevimli Mashg`ulotim!
```


Сатрларни сақлаш учун тахрирлаш

PHP да белгиларни бекор қилиш учун махсус белгиланган иккита функция мавжуд. Ихтиёрий сатрни маълумотлар базасига ёзишдан аввал уларни **AddSlashes()** функцияси ёрдамида тахрирлаш лозим.

11.12 – мисол. AddSlashes() функциясининг ишлатилиши

```
<?
$comments = AddSlashes($comments);
?>
```

Бошқа сатрлар билан ишловчи функциялар каби AddSlashes() функцияси ҳам параметр сифатида сатрни қабул қилади ва тахрирланган сатрни қайтаради.

AddSlashes() функцияси ишлатилганда сатр маълумотлар базасида тескари чизик белгиси билан бирга сақланади. Сатрни олишда тескари чизик белгисини ўчиришни унутмаслик лозим. Тескари чизик белгисини ўчириш StripSlashes() функцияси ёрдамида амалга оширилади:

11.13-мисол. StripSlashes() функциясининг ишлатилиши

```
<?
$comments = StripSlashes($comments);
?>
```

11.3. Сатрларни таққослаш

Сатр узунлигини текшириш(аниқлаш): strlen() функцияси

Белгилардан иборат сатр узунлигини strlen() функцияси ёрдамида аниқлаш мумкин.

Синтаксиси:

int strlen(string camp)

11.14 - мисол. Берилган сатр узунлигини strlen() функцияси ёрдамида ҳисоблаш.

```
<?
$string = "hello";
$length = strlen($string);
// $length = 5
print $length;
?>
Natija:5
```

Бу функция ёрдамида киритилган маълумотларнинг тўғрилигини ҳам текшириш мумкин. \$email ўзгарувчисида сақланадиган, формада яратилган электрон почта манзилини кўриб ўтамыз. \$email ўзгарувчисида сақланаётган электрон почта манзилини тўғрилигини текширишнинг асосий усулларида бири – унинг узунлигини текшириш. Ўйлаб қараганда энг кичик электрон почта манзилининг узунлиги олтига белгига тенг – масалан, a@a.to каби. Шунга кўра, дастур агар манзил узунлиги олтидан кичик бўлса, хато ҳақидаги хабарни чиқазсин:

```
<?
if (strlen($email) < 6)
{
echo "email manzil noto`g`ri kiritilgan";
}
exit; // PHP sahifani bajarilishini to`xtatish
?>
```

Бу албатта маълумот тўғрилигини текширишнинг энг сода йўли. Кейинроқ бир мунча мураккаброқ усуллари ҳам кўриб ўтамиз.

Иккита сатрни таққослаш: *strcmp()*, *strcasemp()*, *strspn()*, *strcspn()*

Иккита сатрни таққослаш ихтиёрий дастурлаш тилининг сатрлар билан ишлаш амалларининг асосий қисми ҳисобланади. Бу масалани бир нечта турли усуллар билан ечиш мумкин, РНР да сатрларни таққослаш учун тўртта функция мавжуд:

- *strcmp()*;
- *strcasemp()*;
- *strspn()*;
- *strcspn()*.

***Strcmp()* функцияси**

strcmp() функцияси иккита сатр белгиларини таққослайди. Бу функция сатрларнинг регистрига(катта-кичиклигига) аҳамият беради.

Синтаксиси:

int strcmp(string camp1, string camp2)

- 0, агар *camp1* ва *camp2* мос тушса;
- < 0, агар *camp1* *camp2* дан кичик бўлса;
- > 0, агар *camp2* *camp1* дан кичик бўлса.

11.15 - мисол. *strcmp()* функциясини қўллаб иккита сатрни таққослаш.

```
<?
$string1 = "non";
$string2 = "non";
if ((strcmp($string1, $string2)) == 0) :
print "$string1 va $string2 satrlari ekvivalent!<br>";
endif;
// if buyrug'i true qiymatni beradi
$string1 = "non";
$string2 = "tuz";
if ((strcmp($string1, $string2)) == 0) {
echo "$string1 va $string2 satrlari ekvivalent!";}
else {echo "$string1 va $string2 satrlari ekvivalent emas!";}
?>
```

Natija:

non va non satrlari ekvivalent!

non va tuz satrlari ekvivalent emas!

***Strcasemp()* функцияси**

strcasemp() функцияси ***strcmp()*** каби ишлайди, битта фарқли томони— таққослашда белгилар регистрига(катта-кичиклигига) аҳамият қаратилмайди.

Синтаксиси:

int strcasemp(string camp1, string camp2)

11.16 - мисол. *strcasemp()* функциясини қўллаб иккита сатрни солиштириш.

```
<?
$string1 = "non";
$string2 = "Non";
if ((strcasemp($string1, $string2)) == 0) :
print "Satrlar ekvivalent!";
endif;
// if buyrug'i true qiymatni beradi
```

```
?>
```

Natija:Satrlar ekvivalent!

***strspn()* функцияси**

strspn() функцияси биринчи *camp1* сегменти таркибидаги белгиларни *camp2* да иштирок этганларини узунлигини чиқазади.

Синтаксиси:

```
int strspn (string camp1, string camp2)
```

11.17 - мисол. **strspn()** функциясини паролларни текширишда ишлатилиши.

```
<?
```

```
$password = "12345";
if (strspn($password, "1234567890")!=strlen($password)) :
print "Password cannot consist solely of numbers!";
endif;
?>
```

Natija:

***Strcspn()* функцияси**

strcspn() функцияси биринчи сегмент *camp1* таркибидаги белгиларни *camp2* да иштирок этмаганларини узунлигини чиқазади.

Синтаксиси:

```
int strcspn (string camp1, string camp2)
```

11.18 - мисол. **strcspn()** функциясини паролларни текшириш учун ишлатилиши.

```
<?
```

```
$password = "12345";
if (strcspn($password, "1234567890") == 0) :
print "Password cannot consist solely of numbers!";
endif;
```

```
?>
```

Natija:Password cannot consist solely of numbers!

11.4. Сатрлар билан ишловчи функциялар ёрдамида сатрларни ажратиш ва бирлаштириш

Жуда ҳам керакли функциялардан бири булар – сатрни қисмларга ажратиш функцияси ва унинг акси сатрларни битта сатрга бирлаштириш функцияси. Нима учун бу функциялар керак? Масалан, агар фойдаланувчи хохишига кўра формани динамик тарзда генерация қилмоқчи бўлсак, танлаш рўйхатини яратиш учун элементларни бирор-бир белги ёрдамида ажратиб киритишни таклиф этиш мумкин. Ва ҳосил бўлган қийматлар рўйхатини саралаш учун сатрни қисмларга ажратиш керак бўлади. Сўнг олинган рўйхатлар қийматларини қайта ишлаш учун сатрни бўлакларга бўлиш керак бўлади. Бу каби бўлакларга ажратишни амалга ошириш учун PHP нинг бир нечта функцияларидан фойдаланиш мумкин:

```
explode(ажратувчи, берилган_сатр  
[, элементларнинг_энг_катта_сони])  
split (шаблон, берилган_сатр  
[, элементларнинг_энг_катта_сони])  
preg_split (шаблон, берилган_сатр  
[, элементларнинг_энг_катта_сони  
[, байроқ]])
```

Охирги иккита функция доимий ифодалар билан ишлашади, шунинг учун биз уларни кўриб ўтмаймиз. Бирмунча соддароқ бўлган функция– **explode()** функцияси билан танишамиз.

explode() функцияси сатрни элементларга ажратади ва бу элементларни массив сифатида чиқазади.

Синтаксиси:

```
array explode (string ажратувчи, string берилган_сатр [,int  
элементларнинг_энг_катта_сони])
```

explode() функцияси берилган_сатрни ажратувчи параметрида кўрсатилган белги ёрдамида қисм сатрларга ажратади ва бу қисм сатрлардан ҳосил бўлган массивни чиқазади. Агар қўшимча параметр **элементларнинг_энг_катта_сони** берилган бўлса, у ҳолда массив элементлари сони бу сондан катта бўлмайди, охирги элементда сатрнинг барча қолдиғи ёзилади. Агар ажратувчи параметри сифатида бўш сатр « » кўрсатилса, у ҳолда **explode()** функцияси false ни қайтаради. Агар ажратувчи белгиси берилган сатрда бўлмаса, у ҳолда берилган сатр ўзгартирилмаган ҳолда чиқазилади. **Масалан,**

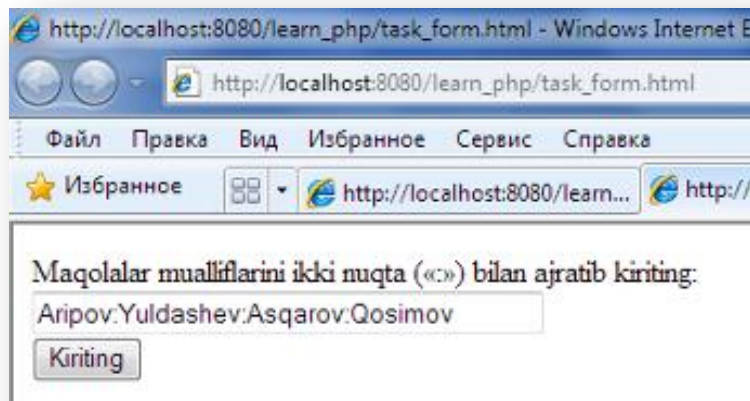
```
<?php  
$info = "wilson | baseball | indians";  
$user = explode("|", $info);  
// $user[0] = "wilson";  
// $user[1] = "baseball";  
// $user[2] = "Indians";  
print_r($user);  
?>  
Natiya:Array ( [0] => wilson [1] => baseball [2] => indians )
```

11.19 - мисол. **explode()** функциясининг қўлланилиши.

Очиловчи рўйхат форма элементини яратамиз. Бу форма элементини қийматлари фойдаланувчи томонидан киритилади. Бунинг учун қуйидагича дастур кодини ёзамиз:

```
<form action=test.php>
Maqolalar mualliflarini ikki nuqta («:») bilan ajratib kiriting:
<br>
<input type=text name=muallif size=40>
<br>
<input type=submit value=Kiriting>
</form>
```

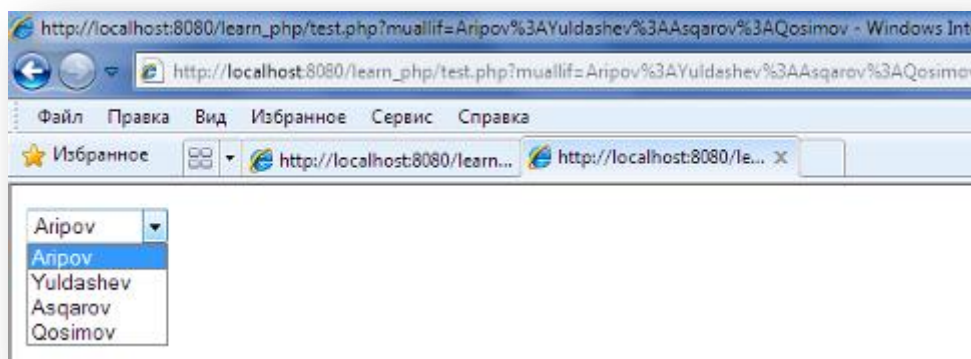
Натижада қуйидаги формага эга бўламиз:



11.3 - расм. Очиловчи рўйхат ҳосил қилиш учун қийматларни киритиш формаси

Бу киритилган маълумотлар test.php саҳифасида қайта ишланади:

```
<?php
$str = $_GET["muallif"];
$names = explode(":", $str);
// ":" yordamida foydalanuvchi
// tomonidan kiritilganlarni ajratamiz
$s = "<select name= muallif >";
// ochiluvchi ro'yhatni yaratamiz
foreach ($names as $k => $name) {
    $s .= "<option value=$k>$name";
    // ro'yhatga elementlarni qo'shamiz
}
$s .= "</select>";
echo $s;
?>
```



11.4 - расм. Формани қайта ишлаш натижасида ҳосил бўлган очилувчи рўйхат

***implode()* функцияси**

Сатрни қисмларга ажратишдан ташқари баъзида, бунинг акси, бир нечта сатрни битта бутун сатрга бирлаштириш зарурати туғилади. PHP да бунинг учун ***implode()*** функция таклиф этилади. ***explode()*** функцияси сатрни массив элементларига ажратса, ҳосил бўлган массивни ***implode()*** функцияси ёрдамида битта сатрга бирлаштириш мумкин.

Синтаксиси:

implode (*сатр_массиви*, *бирлаштирувчи*)

Бу функция *бирлаштирувчи* белгиси ёрдамида берилган массив элементларини бирлаштиради. ***explode()*** функциясидан фарқли равишда ***implode()*** функциясида параметрлар тартибининг аҳамияти йўқ.

11.20 – мисол. ***implode()*** функциясининг қўлланилиши.

Бирор бир шахснинг манзили: республикаси, шаҳар ёки вилояти, тумани, маҳалласи алоҳида алоҳида майдонларда сақланган. Уларни бирлаштириб манзил сифатида кўрсатилиши керак. Бу майдонлардаги сатрларни битта сатрга бирлаштириш учун ***implode()*** функциясини ишлатиш мумкин:

```
<?php
$a_j_manzil = array ("O`zbekiston", "Toshkent shaxri", "Yunusobod tumani", "Qo`shchinor mahallasi");
$manzil = implode(", ", $a_j_manzil);
Echo $manzil;
/* $manzil = "O`zbekiston, Toshkent shaxri, Yunusobod tumani, Qo`shchinor mahallasi"; */
?>
```

Natija: O`zbekiston, Toshkent shaxri, Yunusobod tumani, Qo`shchinor mahallasi

join() функцияси ***implode()*** функциясининг ўхшаши бўлиб, у бажарган барча вазифаларни бажаради, бу иккала функция бир-биридан фақатгина номи билан фарқ қилади.

***strtok()* функцияси**

strtok() функцияси *сатр*ни иккинчи параметрда берилган *ажратувчи*га асосан лексемаларга ажратади.

Синтаксиси:

string strtok (*string camp*, *string ажратувчи*)

strtok() функциясининг битта ноқулай тарафи бор: сатрни тўлиқ ажратиш учун функцияга бир неча марта кетма-кет мурожаат этиш керак. Навбатдаги мурожаатда функция *сатр*дан кейинги лексемани белгилайди. Шу билан бирга «*сатр*» параметри функцияга бир марта берилади, функция эса ҳар гал сатрни ажратгандан сўнг кетма-кет жорий позицияни белгилаб боради. Бу жараён сатр лексемаларга тўлиқ ажратилгунча бажарилади.

11.21 - мисол. Бир нечта *ажратувчилар* ёрдамида берилган *сатр*ни ажратиш.

```
<?
$info = "TDPU:tdpu_info@edu.uz | Uzbekiston, Toshkent";
// Chegaralar - ikki nuqta (:), vertikal chiziq (|) va vergul (,)
$ajrat = ":", "|", ",";
$ajratilgan = strtok($info, $ajrat);
while ($ajratilgan) :echo "Element = $ajratilgan<br>";
$ajratilgan = strtok($ajrat);endwhile;
?>
```

Natija:

Element = TDPU

Element = tdpu_info@edu.uz

Element = Uzbekiston

Element = Toshkent

***substr()* функцияси**

Баъзида биз қидирилайётган сатр қайси белгидан бошланишини билмаймиз, лекин масалан, бу сатр бешинчи элементдан бошланиб, берилган сатр охирига қараб иккита белгидан сўнг тугатилади. Бундай ҳолатда қисм сатрни белгилаш қандай амалга оширилади? Бунинг учун ***substr()*** функциясидан фойдаланамиз.

substr() функцияси *берилган_сатр*дан *бошланиш_белгисининг_ўрни*дан бошлаб *узунлиги* *узунлиги* параметрига тенг бўлган қисм сатрни ажратиб олади.

Синтаксиси:

string ***substr()***(*string* берилган_сатр, *int* бошланиш_белгисининг_ўрни [, *int* узунлиги])

Агар *узунлиги* параметри кўрсатилмаса, у ҳолда функция кўрсатилган *бошланиш_белгиси_ўрни*дан бошлаб *берилган_сатр* охиригача бўлган қисм сатрни чиқазади. Бу функцияни ишлатишда қуйидагича ҳолатлар юзага келади. *бошланиш_белгисининг_ўрни* параметри ҳам мусбат бутун сон, ҳамда манфий бутун сон бўлиши мумкин.

- агар «*бошланиш_белгисининг_ўрни*» параметри мусбат бутун сон бўлса, қисм сатр берилган сатрнинг «*бошланиш_белгисининг_ўрни*» параметрида берилган номер ўрнидан бошланади;

- агар «*бошланиш_белгисининг_ўрни*» параметри манфий бутун сон бўлса, қисм сатр (*сатр узунлиги*- *бошланиш_белгисининг_ўрни* =) ўрнидан бошланади;

- агар «*узунлиги*» параметри мусбат бутун бўлса, қисм сатр «*бошланиш_белгисининг_ўрни*»дан бошлаб (*бошланиш_белгисининг_ўрни* + *узунлиги* =) ўрнигача бўлган барча белгиларни чиқазади. Агар бу қиймат

сатр узунлигидан катта бўлса, у ҳолда сатр охиригача бўлган белгилар чиқазилади;

- «узунлиги» параметри манфий сон ҳам бўлиши мумкин. Агар «узунлиги» параметрида манфий сон берилса, у ҳолда сатр охиридан кўрсатилган сонга тенг белги ташлаб юборилади.

11.22 – мисол. **substr()** функциясининг қўлланилиши

```
<?
$car = "1998 Nexia";
$model = substr($car, 5);
// 5 belgidan boshlab satr oxirigacha bo`lgan belgini chiqazadi
echo $model;
// $model = "Nexia"
?>
Natija:Nexia
```

11.23 – мисол. **substr()** функциясининг қўлланилиши

«Узунлиги» параметри мусбат бўлган ҳолат:

```
<?
$car = "1998 Nexia";
$model = substr($car, 0, 4);
/* 0 belgidan boshlab
4 ta belgili qism satrni belgilaydi*/
// $model = "1998"
echo $model;
?>
Natija:1998
```

11.24 – мисол. **substr()** функциясининг қўлланилиши

«Узунлиги» параметри манфий бўлган ҳолат:

```
<?php
$car= "<b>1998 Nexia</b>";
echo $car, "<br>";
$model = substr($car, 3, -4);
/* 3 belgidan boshlanib oxiridan
4 ta belgi tushirib qoldirilgan
qism satrni belgilaydi*/
echo $model;
?>
Natija:
1998 Nexia
1998 Nexia
```

11.24 - мисолдаги каби мисолларни ечишда **strip_tags()** функциясидан фойдаланиш **substr()** функциясига нисбатан қулайроқ.

strip_tags() функцияси

Синтаксиси:

strip_tags(*camp* [, қолдириладиган_теглар])

Бу функция *camp*даги барча html ва php тегларни ўчиради. Қўшимча параметр - *қолдириладиган_теглар* ёрдамида сатрдан ўчирилиши керак бўлмаган тегларни бериш мумкин. Теглар рўйхати ҳеч қандай белги билан ажратилиши керак эмас. Агар нотўғри ёки тўлиқ бўлмаган тег учраса функция бу ҳақида хабар беради.

11.25 - мисол. **strip_tags()** функциясининг қўлланилиши

```
<?php
$string = "<b>Bold text</b>
<i>Italic text</i>";
$str = strip_tags($string);
```

```
// satrdan barcha teglarni o`chiradi
$str1 = strip_tags($string, '<i>');
// <i> dan tashqari barcha teglarni o`chiradi
$str2 = strip_tags($string, '<i><b>');
// <i> va <b> dan boshqa barcha teglarni o`chiradi
echo $str, "<br>", $str1, "<br>", $str2;
?>
```

Natija:

Bold text *Italic text*

Bold text *Italic text*

Bold text *Italic text*

substr() функцияси ёрдамида бошқа мисолни кўрамиз.

11.26 - мисол. Саломлашув ва имзоси билан берилган қандайдир хабар бор. Биринчи навбатда саломлашувни, сўнг эса имзони ўчирамиз ва фақатгина хабарнинг асосий қисмини қолдирамиз.

```
<?php
$text = "Salom! Bugun biz satrlar bilan ishlashni o`rganamiz. Muallif.";
$no_hello = substr($text, 7);
// salomlashuvni o`chiramiz
$content = substr($text, 7, 45);
// substr($text, 7, -9) kabi
// imzoni o`chiramiz
echo $text, "<br>", $no_hello,
"<br>", $content;
?>
```

Natija:

Salom! Bugun biz satrlar bilan ishlashni o`rganamiz. Muallif.

Bugun biz satrlar bilan ishlashni o`rganamiz. Muallif.

Bugun biz satrlar bilan ishlashni o`rganamiz.

Агар сатрдан аниқ бир белги керак бўлса ва бу белгини тартиб номерини билсак, бу ҳолда *substr()* типдаги функцияларни ишлатиш шарт эмас. Бундан ҳам оддийроқ синтаксисдан фойдаланиш мумкин – бунинг учун сатрий ўзгарувчидан сўнг белги номерини фигурали қавс ичига олиб ёзсак кифоя. Мисол учун берилган сатрда «а» ҳарфи ҳисоб бўйича иккинчида турибди, у ҳолда:

```
$text = "Salom! Bugun biz satrlar bilan ishlashni o`rganamiz. Muallif.";
echo $text{1}; // «a» belgisini chiqazamiz
```

Бу белги номери икки эмас, балки бир, бундан билиш мумкинки, сатр белгилари тартиб номери нолдан бошланади.

Сатрни белгилари ва уларни тартиб номери ҳақида гапирар эканмиз, у ҳолда сатрда нечта белги бор ва уларни қандай ҳисоблаш мумкин каби саволлар юзага келади. Сатр белгилари сони – бу *camp узунлигидир*. *Camp узунлигини strlen()* функцияси ёрдамида ҳисоблаш мумкин.

substr_count() функцияси

substr_count() функцияси берилган сатрдаги қисм сатрларни неча марта учраганлиги сонини чиқазади.

Синтаксиси:

int substr_count (string берилган camp, string қисм camp)

11.27 - мисол. *substr_count()* функцияси ёрдамида айн қисм сатри неча марта учраганлигини ҳисобланг:

```
<?
$tnng_twist = "The rain falls mainly on the plainsof Spain";
```

```
$count = substr_count($tng_twist, "ain");  
// $count = 4  
print $count;  
?>
```

Natija:4

Демак, қисм сатрларни қандай топиш ва белгилашни кўриб ўтдик. Энди *берилган сатр*даги қисм сатрларни бошқа янги қисм сатрларга алмаштиришни ўрганамиз.

11.5. Қисм сатрларни қидириш ва алмаштириш

Қисм сатрларни ўрнини аниқлаш: `strpos()`, `strrpos()`

`strpos()` функцияси

Берилган сатр таркибида қисм сатр мавжудлигини текшириш учун *`strpos()`* функцияси ишлатилади.

Синтаксиси:

`strpos` (*берилган_сатр*, *қидирилаётган_қисм_сатр*
[*қайси_белгидан_бошлаб_қидириш_керак*])

*Қидирилаётган_қисм_сатр*ни *берилган_сатр*даги ўрнини чиқазади, акс ҳолда агар *қидирилаётган_қисм_сатр* *берилган_сатр* таркибидан топилмаса, `false` матикий жавобни қайтаради. Кўшимча *параметр* қидирув сатрнинг қайси белгисидан бошланиши кераклигини кўрсатиш учун имконият яратади. Мантикий `false` дан ташқари функция `false` (масалан, 0 ёки «») ни ўрнини босувчи бошқа қийматларни ҳам чиқазиши мумкин. Шунинг учун *қидирилаётган_қисм_сатр* мавжудлигини текшириш учун «===» эквивалент амалини қўллаш тавсия этилади.

11.28 - мисол. *`strpos()`* функцияси ёрдамида сатр таркибидаги қисм сатрни мавжудлигини текшириш.

```
<?
$str = " Albatta aytish kerak-
ki, PHP dasturlash tillari orasida eng yaxshisiga aylandi va ulardagi yetishm
ovchiliklarni maksimal darajada bartaraf qila oldi. PHP texnologiyasining vaz
ifasi - Webda ishlovchilarga html-
sahifani oson va tez, dinamik holda ishlashi va o'zgartirish uchun imkoniyatl
ar yaratishdan iborat. PHP texnologiyasi bularni a'lo darajada bajaradi.";
$pos = strpos($str,"PHP");
if ($pos!== false) echo "Qidirilayotgan satr $pos o'rinda topildi";
else echo "Qidirilayotgan satr topilmadi";
?>
Natija:Qidirilayotgan satr 26 o'rinda topildi
```

Агар *қидирилаётган_қисм_сатр* параметри қиймати сатрли бўлмаса, у ҳолда бутун типга акслантирилади ва ASCII-код белгиси сифатида қаралади. PHP да ихтиёрий белгини ASCII-кодини олиш учун *`ord()`* («белги») функциясидан фойдаланиш мумкин.

Масалан, агар `$pos = strpos($str,116);` каби ёзилса интерпретатор «t» белгиси қидирилаётган деб ҳисоблайди. Агар бу сатрни юқоридаги мисолга қўйилса ва натижа чиқазилса, у ҳолда қидирилаётган сатр 5 - ўринда деб чиқазилади.

`Ord()` функциясига тескари функция, – бу *`chr()`* («код») функциясидир. Бу функция берилган кодга мос ASCII-коди белгисини чиқазади.

`strpos` функцияси *берилган_сатр* таркибидаги *қидирилаётган_қисм_сатр*ни фақатгина биринчи учраганини жойлашган ўрнини чиқазади. *Берилган_сатр* таркибидаги *қидирилаётган_қисм_сатр*ни охирги учраганини жойлашган ўрнини аниқловчи функция ҳам мавжуд. Бу функция *`strrpos()`*.

`strrpos()` функцияси

Синтаксиси:

`strrpos` (*берилган_сатр*, *қидирилаётган_белги*)

`strrpos()` дан фаркли равишда бу функция *қидирилаётган_белгини*

*берилган_сатр*даги охирги учраган ўрнини топиш учун имкон яратади. Имкониятларига кўра бу функциянинг `strpos()` функциясидан фарқи шундаки - бу функция бутун сатрни ўрнини эмас, балки фақатгина алоҳида белгини ўрнини аниқлай олади. Агар `strrpos()` да иккинчи параметр сифатида сатр берилса, у ҳолда қидириш вақтида унинг фақатгина биринчи белгиси ишлатилади.

Шундай ҳолатлар бўладики, қидирилаётган сатр каерда жойлашганлигини ўрнини билиш зарур бўлмайди, балки шу ўриндан бошлаб сатрда жойлашган барча белгиларни олишга тўғри келади. Бундай ҳолатларда `strpos()` ва `strrpos()` функцияларини эмас, балки қисм сатрларни белгилаш учун ёрдам берувчи функциялардан фойдаланиш керак.

Сатрдаги қисм сатрларни қидириш: `strstr()`, `strchr()`, `strrchr()`, `stristr()`
***Strstr()* функцияси**

PHP да берилган сатрдан қисм сатрни ажратиб олиш ҳақида гап кетганда биринчи ўринда `strstr()` функциясига тўхталишимиз мумкин.

Синтаксиси :

`string strstr(string берилган_сатр, string қидирилаётган_сатр)`

`strstr()` функцияси *қидирилаётган_сатр*нинг биринчи учраганидан бошлаб *берилган_сатр* охиригача бўлган қисм сатрни ажратиб олади.

Масалан,

```
<?
$url = "http://www.apress.com";
$domain = strstr($url, ".");
echo $domain;
// $domain = ".apress.com"
?>
Natija: .apress.com
```

Агар *қидирилаётган_сатр* топилмаса, у ҳолда функция `false` ни қайтаради. Агар *қидирилаётган_сатр* сатрли маълумотлар типи кўринишида бўлмаса, у ҳолда бутун типга ўтказилади ва *белги коди* сифатида қаралади. Бундан ташқари, бу функция регистрга(катта-кичиклигига) аҳамият қаратади, агар биз «PHP» ва «php» сўзларини параллел равишда қидирмоқчи бўлсак, у ҳолда натижа турлича бўлади. `strstr()` билан биргаликда унга жуда ўхшаш бўлган функция `strchr()` ни ҳам ишлатиш мумкин.

11.29 - мисол. `strstr()` функциясининг ишлатилиши:

Номи, муаллифидан иборат бўлган сатрдан «Nomi» сўзи билан бошланувчи қисм сатрни ажратиб оламиз.

```
<?
$str = "Muallif: Azimov Akrom(<a
href=mailto:akrom@mail.ru>xat yozing</a>),
Nomi: 'Web dasturlash tili' ";
echo "<b>Berilgan satr: </b>", $str;
if (!strstr($str, "Nomi"))
echo "Satr topilmadi<br>";
else echo "<p><b>Olingan satr: </b>",
strstr($str, "Nomi");
?>
Natija:
Berilgan satr: Muallif: Azimov Akrom(xat yozing), Nomi: 'Web dasturlash tili'
```

Olingan satr: Nomi: 'Web dasturlash tili'

Қидиришда қисм сатрни регистрга(катта-кичиклигига) аҳамият қаратмаслиги учун ушбу функцияга ўхшаш функция - **stristr()** (*берилган_camp*, *қидирилаётган_camp*) функциясидан фойдаланиш мумкин. Бу функция худди **strstr()** каби ишлайди, фақатгина қидиришда *қидирилаётган_camp* белгилари регистрга(катта-кичиклигига) аҳамият қаратмайди. **strchr()** функцияси **strstr()** каби ишлайди, унинг иккинчи варианты **strchr()** эса **strstr()** функциясига ўхшаш битта фарқи *берилган_camp*даги қисм сатрни *қидирилаётган_camp*ни охириги учраганидан бошлаб чиқазади.

Амалиётда жуда кам ҳолатларда аниқ сўз ёки сатр билан бошланувчи қисм сатрни олиш керак бўлади, шунинг учун **strstr()** функцияси тез-тез ишлатилмайди. Бундан ташқари, РНР нинг қидириш учун қулай бошқа функциялари ҳам мавжуд.

Қисм campларни алмаштириш: str_replace(), substr_replace()

str_replace() функцияси

Бу функция оддий ва қулай бўлиб, алмаштириладиган қисм сатрга алоҳида талаблар бўлмаган ҳолда, кўпгина масалаларни ҳал қилишга имкон беради. Мураккаб шартлар билан берилган алмаштиришларда доимий ифодалар усули ва мос функциялар **ereg_replace()**, **preg_replace()** ишлатилади.

Синтаксиси:

string str_replace (string қисм_camp, string янги_қисм_camp, string camp)

str_replace() функцияси *camp* таркибида учрайдиган *қисм_camp*ни бошқа *янги_қисм_camp*га алмаштириш учун ишлатилади.

ёки

str_replace (қиймат, янги қиймат, объект)

str_replace() функцияси қаралаётган объектдан қийматни қидиради ва ушбу қийматни белгиланган янги қийматга алмаштиради.

Кейинги синтаксисда параметрлар сатр эмас, балки объект ва қиймат деб юритилганлигининг сабаби, РНР 4.0.5 версиясидан бошлаб, бу функциянинг ихтиёрий параметри массив бўлиши мумкин.

Агар қисм сатр берилган сатрда учрамаса, берилган сатр ўзгармайди.

11.30 - мисол. str_replace() функциясининг қўлланилиши

```
<?php
$sevimli_fan= "Mening sevimli fanim matematika va fizika";
$sevimli_fan= str_replace("matematika", "informatika", $sevimli_fan);
// $sevimli_fan= "Mening sevimli fanim informatika va fizika"
echo $sevimli_fan;
?>
```

Natija:Mening sevimli fanim informatika va fizika

Агар алмаштириш бажариладиган объект массив бўлса, у ҳолда бу амал массивнинг барча элементлари учун бажарилади ва натижа янги массив сифатида чиқазилади.

11.31 - мисол. str_replace() функциясининг қўлланилиши. 1 – усул.

```
<?php
$salomlashuv = array("Salom", "Salom hammaga!",
```

```

"Salom, do`stlar!"); // obyekt
$yangi_salomlashuv = str_replace("Salom",
"Hayrli tong", $salomlashuv);
// almashtirishdan so`ng
print_r($yangi_salomlashuv);
/* natija: Array ([0]=> Hayrli tong
[1]=> Hayrli tong hammaga!
[2]=> Hayrli tong, do`stlar!) */
?>
Natija:
Array (
[0] => Hayrli tong
[1] => Hayrli tong hammaga!
[2] => Hayrli tong, do`stlar! )

```

Агар кидирилаётган қиймат ва алмаштириладиган қиймат – массивлар бўлса, у ҳолда ҳар бир массивдан биттадан қиймат олиниб, объектда кидириш ва алмаштириш бажарилади. Агар алмаштирилидиган қиймат кидирилаётган қийматдан кичик бўлса, у ҳолда янги қиймат сифатида бўш сатр олинади.

11.32 - мисол. str_replace() функциясининг қўлланилиши. 2 – усул.

```

<?php
$greeting = array("Salom", "Salom hammaga!",
"Salom, do`stlar!", "Assalomu alaykum",
"Assalomu alaykum, o`rtoqlar", "Hi");
// obyekt
$search = array ("Salom",
"Assalomu alaykum", "Hi");
// qidirilayotgan qiymat
$replace = array ("Hayrli tong",
"Hayrli kun");
// almashtirilishi kerak bo`lgan qiymat
$new_greet = str_replace($search, $replace,
$greeting);
// almashtirishni bajarish
print_r($new_greet);
//hosil bo`lgan massivni chiqarish
?>
Natija:
Array (
[0] => Hayrli tong
[1] => Hayrli tong hammaga!
[2] => Hayrli tong, do`stlar!
[3] => Hayrli kun
[4] => Hayrli kun, o`rtoqlar
[5] => )

```

Агар кидирилаётган қиймат массив, алмаштириладиган қиймат эса сатр бўлса, у ҳолда бу сатр барча топилган қийматларни алмаштириш учун ишлатилади.

11.33 - мисол. str_replace() функциясининг қўлланилиши. 3 – усул.

```

<?php
$greeting = array("Salom", "Salom hammaga!",
"Salom, do`stlar!", "Assalomu alaykum",
"Assalomu alaykum, o`rtoqlar");
// obyekt
$search = array ("Salom", "Assalomu alaykum");
// almashtirilishi kerak bo`lgan qiymat

```



```
$replace = "Hayrli kun";
// almashtirilishi kerak bo'lgan qiymat
$new_greet = str_replace($search,
    $replace, $greeting); // almashtirishni bajarish
print_r($new_greet);
// hosil bo'lgan massivni chiqarish
?>
Natija:
Array ( [0] => Hayrli kun
[1] => Hayrli kun hammaga!
[2] => Hayrli kun, do'stlar!
[3] => Assalomu alaykum
[4] => Assalomu alaykum, o'rtoqlar )
```

str_replace() функцияси регистрга(белгиларнинг катта-кичиклигига) аҳамиятли, лекин регистрга аҳамиятли бўлмаган унга ўхшаш функция мавжуд, бу функция - ***str_ireplace()***. Лекин бу функция PHP нинг ҳамма версияларида ҳам қўлланилмайди.

***substr_replace()* функцияси**

Бу функция кўриб ўтилган иккита функция ***str_replace()*** ва ***substr()*** ларнинг хусусиятини ўзида жамлаган.

Синтаксиси:

substr_replace (*берилган_satr*, *алмаштириладиган_satr*,
белгининг_бошлангич_ўрни[, *узунлиги*])

бу функция параметрларга асосан *берилган_satr*ни ҳаммасини ёки қисмини *белгининг_бошлангич_ўрни* параметрида кўрсатилган ўридан бошлаб *алмаштириладиган_satr*га алмаштиради. Қўшимча параметр сифатида берилган *узунлиги* ёрдамида алмаштириладиган белгилар сонини чегаралаш мумкин. Аслида, сатрдаги алмаштирилиши керак бўлган қисм сатр кўрсатилмайди, фақатгина у қаерда жойлашгани ва қандай узунликка эга эканлиги кўрсатилади ва шу хусусияти билан ***substr_replace()*** функцияси ***str_replace()*** функциясидан фарқ қилади.

substr() функциясидаги каби *белгининг_бошлангич_ўрни* ва *узунлиги* параметрлари манфий бўлиши мумкин:

- агар «*белгининг_бошлангич_ўрни*» параметри мусбат бўлса, алмаштириш кўрсатилган позициядан бошланади;

- агар «*белгининг_бошлангич_ўрни*» параметри манфий бўлса, алмаштириш (*satr узунлиги – белгининг_бошлангич_ўрни*=) позициядан бошланади;

- агар «*узунлиги*» параметри мусбат бўлса, сатр берилган узунлик бўйича алмаштирилади;

- агар «*узунлиги*» параметри манфий бўлса, сатр охиридан бошлаб нечта белги алмаштириш керак эмаслигини кўрсатади ёки алмаштириш (*satr_узунлиги-узунлиги*=) позицияда тугатилади.

- агар «*узунлиги*» параметри кўрсатилмаса, у ҳолда алмаштириш сатр охиригача бажарилади.

11.34 – мисол. *substr_replace()* функциясининг ишлатилиши

```
<?php
$sevimli = "ning sevimli radiosini";
$name = "Lola";
// "0, 0" parametri almashtiriladigan fragment
```

```
//sattrning birinchi o`rnidan boshlanadi va tugatiladi
$sevimli = substr_replace($sevimli, $name, 0, 0);
print $sevimli;
?>
```

Natija:Lolaning sevimli radiosi

11.35 – мисол. *substr_replace()* функциясининг ишлатилиши

```
<?php
$text = "Mening ismim Komil.";
echo "Berilgan satr: $text<hr>\n";
/* quyidagi berilgan satr barcha berilgan satrlarni
'Meniki esa- Farhod' satriga almashtiradi*/
echo substr_replace($text, 'Meniki esa - Farhod',
0) . "<br>\n";
echo substr_replace($text, 'Meniki esa - Farhod',
0, strlen($text)) . "<br>\n";
// quyidagi satr berilgan satr
// boshiga 'Salom! ' so`zini qo`shadi
echo substr_replace($text, 'Salom! ',
0, 0) . "<br>\n";
// Quyidagi ikkita satr berilgan satrdagi Komil ismini
// Ilhom ismiga almashtiradi
echo substr_replace($text, 'Ilhom', 13,
-1) . "<br>\n";
echo substr_replace($text, 'Ilhom', -6,
-1) . "<br>\n";
?>
```

Natija:

Berilgan satr: Mening ismim Komil.

```
Meniki esa - Farhod
Meniki esa - Farhod
Salom! Mening ismim Komil.
Mening ismim Ilhom.
Mening ismim Ilhom.
```

11.6. html-кодлардан таркиб топган сатрлар

html-тегларида таркиб топган сатрлар билан тез-тез ишлаймиз. Агар бундай сатрни браузерга *echo()* ёки *print()* ёрдамида чиқазсак, у ҳолда биз html-теглари ўзини кўрмаймиз, балки шу тегларга мос форматланган сатрни оламиз. Браузер барча html-теглари HTML тили стандартларига мос ҳолда қайта ишлайди. Баъзан биз сатрни қайта ишланмаган ҳолда браузерда кўришни ҳохлаймиз. Бунинг учун *htmlspecialchars()* функциясида фойдаланилади.

***htmlspecialchars()* функцияси**

htmlspecialchars (*сатр* [, *қўштирноқ_стили* [, *кодировка*]])

htmlspecialchars() функцияси «<», «>», «&», «'», «"» каби махсус белгиларини HTML тилидаги мос «<», «>», «&», «"», «'» белгиларга алмаштиради.

Қўшимча параметр *қўштирноқ_стили*да иккиталик ва битталик қўштирноқларни қандай интерпретация қилиниши кераклиги кўрсатилади. *қўштирноқ_стили* учта қиймат (ENT_COMPAT, ENT_QUOTES, ENT_NOQUOTES)дан бирига тенг бўлиши керак. ENT_COMPAT константаси иккиталик қўштирноқ махсус белгиси сифатида таржима қилинишини, битталик қўштирноқ эса ўзгаришсиз қолишини англатади. ENT_QUOTES эса ҳам иккиталик қўштирноқ, ҳам битталик қўштирноқ ўзгариши кераклигини, ENT_NOQUOTES эса барча қўштирноқлар ўзгаришсиз қолиши кераклигини билдиради.

Кодировкалар параметрида UTF-8, ISO-8859-1 ва бошқа кодировкалар берилиши мумкин, лекин бирорта рус кодировкалари бу ерда қўлланилмайди.

11.32 - мисол. *htmlspecialchars()* функциясининг қўлланилиши

```
<?php
$new = htmlspecialchars("<a
href='mailto:au@mail.ru'>
Xat yozish</a>", ENT_QUOTES);
echo $new;
/* berilgan satr quyidagicha kodirovkalanadi:
&lt;a href=&#039;mailto:au@mail.ru&#039;&gt;
Xat yozish &lt;/a&gt; */
?>
```

Natija: Xat yozish

htmlspecialchars() функцияси фақатгина тез-тез ишлатиладиган махсус белгиларни қайта кодировкалайди. Агар HTML таркибидаги барча белгиларни ўзгартириш зарур бўлса, у ҳолда *htmlentities()* функциясида фойдаланилади. Бу функцияни ишлатишда ҳам русча ҳарфлар махсус кетма-кетликларга кодланади. Масалан, «А» ҳарфи «À» комбинациясига алмашади. Бу функциянинг синтаксиси ва ишлатилиши қонун-қоидаси *htmlspecialchars()* функциясининг синтаксиси ва ишлатилиш қонун-қоидасига ўхшаш.

11.7. Сатрлар билан ишлашга мисоллар

11.2 - Масала. Хатоларни текшириш

Фойдаланувчи маълумотларини қайта ишлаш фақатгина маълумотлар тўғри структурага эга бўлгандагина кутилган натижани беради. Киритилган маълумотларни ишончилигини, аниқлигини текшириш мумкин эмас, бироқ уларни тўлиқлигини (масалан, электрон почта манзили стандарт шаблонга мос келишини аниқлаш мумкин) текшириш мумкин. Гарчи маълумотларни текшириш учун JavaScript технологиясидан тез-тез фойдаланилса ҳам, браузерда мос келмаслик каби муаммолар ҳам вужудга келиши мумкин. Модомики, РНР коди сервер томонидан бажарилар экан, ҳар доим форма маълумотларини текшириш керакли натижа (албатта, тузилган дастурнинг шартлари тўғрилигига қараб) бериши мумкинлигига ишонч ҳосил қилиш мумкин.

Дастур фойдаланувчи киритган маълумотларида хато борлигини билиши билан унга бу ҳақида хабар ва уни тузатишга таклиф бериши лозим. Бу ҳолларда ечимнинг мумкин бўлган бир нечта вариантлари мавжуд - хатолик ҳақидаги хабарни оддий чиқазиш, альтернатив вариантларни таклиф этиш (масалан, агар фойдаланувчи бошқа фойдаланувчи томонидан танлаб бўлинган логинни танласа) ва ҳ.

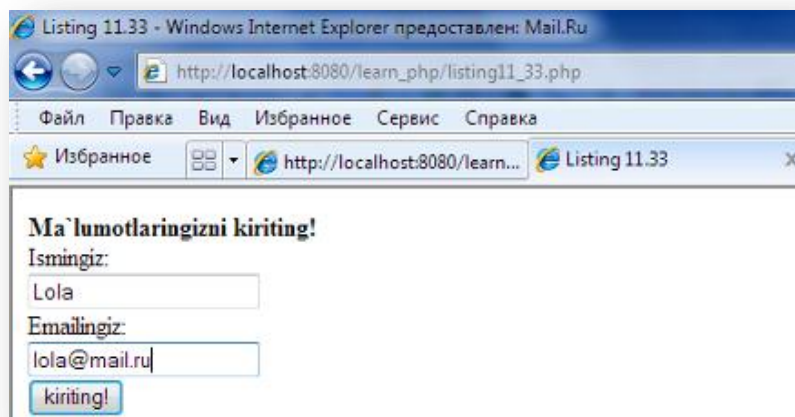
Форманинг барча майдонларини бўш қолмаганлигини аниқлаш учун кетма-кет текшириб чиқамиз. Керак бўлган жойда киритилган маълумотларни структурасини тўғрилиги ҳам текширилади (масалан, электрон почта манзили майдонида). Агар текшириш муваффақиятли якунланса, кейинги майдонга ўтилади; акс ҳолда дастур хатолик ҳақидаги хабарни чиқазади, формани қайта экранга чиқазиш учун кейинроқ ишлатиладиган байроқни ўрнатади ва кейинги майдонга ўтади. Жараён форманинг барча майдонлари текшириб бўлгунча давом этади. (11.33 - листинг).

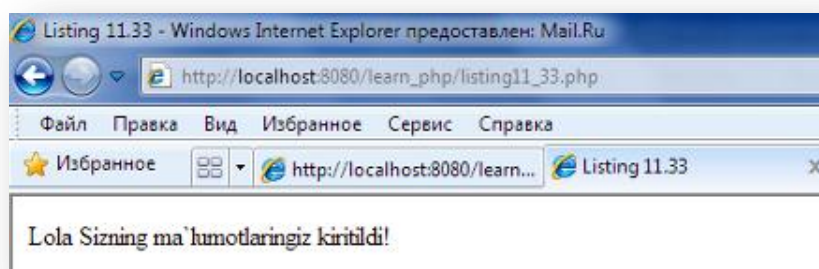
11.33 - листинг. Форма маълумотларини текшириш ва хатолар ҳақида хабар чиқазиш

```
<html>
<head>
<title>Listing 11.33</title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74" vlink="#808040" alink="#808040">
<?
$form = "
<form action=\"listing11_33.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<b>Ma'lumotlaringizni kiriting!</b><br>
Ismingiz:<br> <input type=\"text\" name=\"name\" size=\"20\" maxlength=\"20\" value=\"\"><br>
Emailingiz:<br>
<input type=\"text\" name=\"email\" size=\"20\" maxlength=\"40\" value=\"\"><br>
<input type=\"submit\" value=\"kiriting!\"></form>;
// forma avval to'ldirilganmi?
if (!isset($seenform)):
print "$form";
```

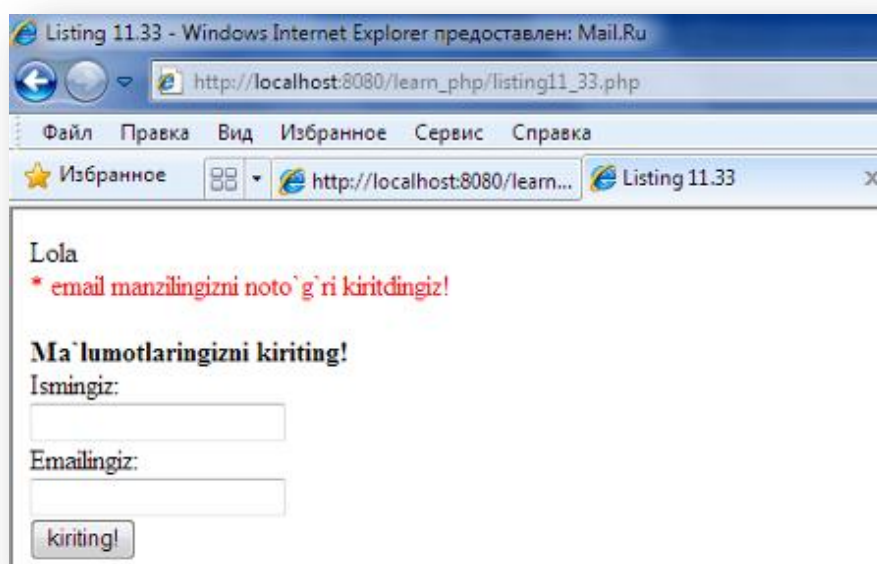
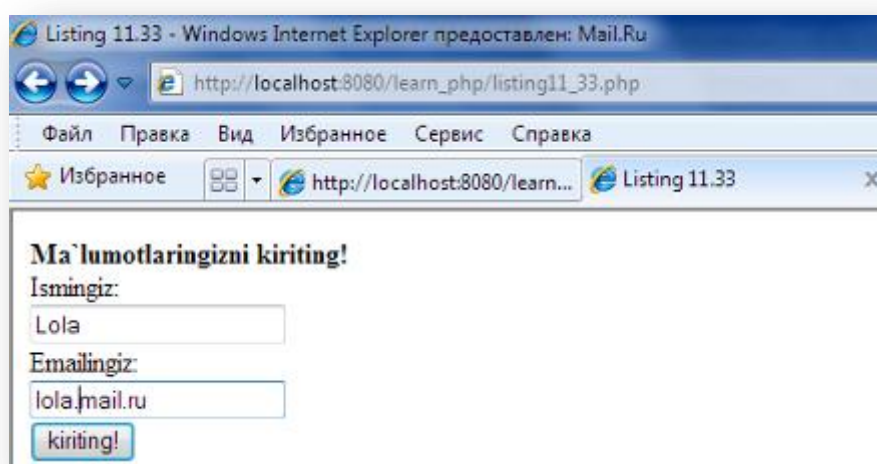
```
// foydalanuvchi formani to'ldirdi
// kiritilgan ma'lumotlarni tekshirish
else :
$error_flag = "n";
// ism maydoniga ma'lumot kiritilganligini aniqlash
echo $name;
if ($name == "") :
print "<font color=\"red\">* Ismingizni kiritmadingiz!</font> <br>";
$error_flag = "y";
endif;
// elektron pochta manzili maydoniga ma'lumot kiritilganligini aniqlash
if ($email == "") :
print "<font color=\"red\">* email manzilingizni kiritmadingiz !</font>
<br>";$error_flag = "y";
else :
// elektron pochta manzilidagi barcha alifbo belgilarini kichik harflarga alm
ashtirish
$email = strtolower(trim($email));
// elektron pochta manzilini sintaksisi to'g'riligini tekshirish
if (!@ereg('^[a-zA-Z0-9_]|\\-|\\.')+'. '@'. '([a-zA-Z0-9_]|\\-|\\.')+'. '[a-zA-
Z]{2,4}$', $email)) :
print "<br><font color=\"red\">* email manzilingizni noto'g'ri kiritdingiz!<
/font> <br>" ;
$error_flag = "y";
endif;
endif;
// agar xato bayrog'i $error_flag o'rnatilgan bo'lsa, formani qayta chiqazish
if ($error_flag == "y") :
print "$form";else : // foydalanuvchi ma'lumotlarini qayta ishlash
print " Sizning ma'lumotlaringiz kiritildi!";
endif;
endif;?>
</body>
</html>
```

Тўғри маълумот киритилганда:





Нотўғри маълумот киритилганда:



11.5–расм. Форма маълумотларини текшириш ва хатолик ҳақидаги хабарни чиқариш

11.33 - листингдаги дастур исм майдони ва электрон почта манзили бўш қолмаганлиги, ҳамда киритилган манзил синтаксиси тўғрилигини кўрсатмоқда. Бирор-бир текшириш натижасида формада хатолик учраса,

дастур мос хабарни беради ва формани қайта экранга чиқазади. Шу билан бирга фойдаланувчи хатоларни тўғрилаши учун формада аввал киритилган маълумотлар сақлаб қолинади.

11.3 - Масала. Фойдаланувчи маълумотларини матнли файлда сақлаш

Фойдаланувчига маълумотларини тўртта объектга киритиш таклиф этилади: исми, электрон почта манзили, тили ва касби. Киритилган маълумотлар user_information.txt матнли файлда сақланади. Маълумот элементлари <вертикал чизик> (|) белгиси билан ажратилади.

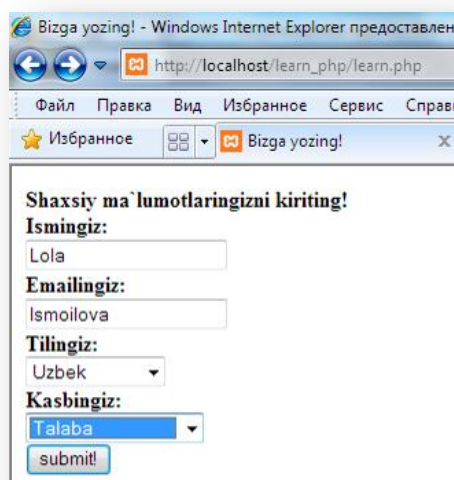
11.34 - листинг. Фойдаланувчи маълумотларини матнли файлда сақлаш

```
<html>
<head>
<title>Bizga yozing! </title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#cbda74" vlink="#808040" alink=
#808040">
<?
$form = " <form action=\"learn.php\" method=\"post\">
<input type=\"hidden\" name=\"seenform\" value=\"y\">
<b>Shaxsiy ma'lumotlaringizni kiriting!</fb><br>
Ismingiz:<br> <input type=\"text\" name=\"name\" size=\"20\" max-
length=\"20\" value=\"\"><br>
Emailingiz:<br> <input type=\"text\" name=\"email\" size=\"20\" maxlength=\"2
0\" value=\"\"><br>
Tilingiz:<br>
<select name=\"language\">
<option value=\"\">Tilni tanlang:
<option value=\"English\">English
<option value=\"Spanish\">Spanish
<option value=\"Italian\">Italian
<option value=\"French\">French
<option value=\"Japanese\">Japanese
<option value=\"uzbek\">Uzbek
</select><br>
Kasbingiz:<br>
<select name=\"job\">
<option value=\"\">Nima ish qilasiz?:
<option value=\"student\">Talaba
<option value=\"programmed\">Dasturchi
<option value=\"manager\">Menejer
<option value=\"teacher\">O'qituvchi
</select><br>
<input type=\"submit\" value=\"submit!\">
</form>"; // forma avval to'ldirilganmi?
if (!isset($seenform)) :
    print "$form";
else : $fd = fopen("useinformation.txt", "a");
    // kiritilgan ma'lumotlarda vertikal chiziq
    // uchramaganligiga ishonch hosil qilish
    $name = str_replace("|", "", $name);
    $email = str_replace("|", "", $email);
    // foydalanuvchi ma'lumotlaridan satr tuzish
    $user_row = $name." ".$email."|".$language." ".$job."\n"; fwrite($fd, $user_r
ow) or
    die("faylga yoza olmadi!");
    fclose($fd);
    print "Bizda ma'lumotlaringizni qoldirganingiz uchun rahmat!";
endif;?>
```

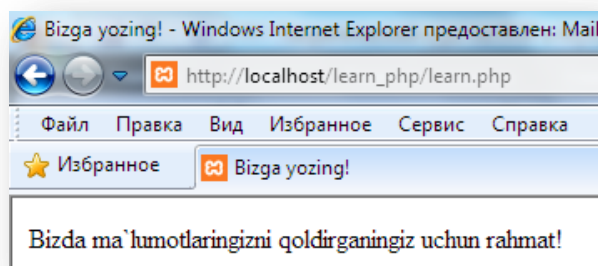


```
</body>
</html>
```

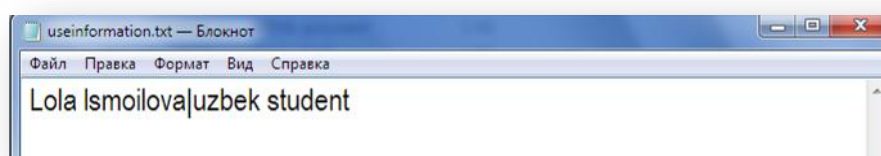
Натижа



11.6 - расм. Маълумотларни матнли файлда сақлаш



11.7 – расм. Маълумотлар жўнатилгандан сўнг *learn.php* файли



11.8 – расм. *useinformation.txt* файли

Фойдаланувчи исм ёки электрон почта манзилида <вертикал чизик> (|) белгисини ишлатмаганлигини текшириш фрагментига эътибор қаратинг. `str_replace()` функцияси бу белгиларни ўчириб, уларни бўш сатрга алмаштиради. Агар бу иш бажарилмаса, фойдаланувчи белгиси (|) маълумотлар файли структурасини бузади ва уни тўғри қайта ишланишига ҳалақит беради.

Бир мунча кичик ҳажмдаги маълумотларни қайта ишлаш жараёнида матнли файллардан тўла фойдаланиш мумкин. Бироқ кўп сондаги фойдаланувчи маълумотлари билан ишлаш (сақлаш ва қайта ишлаш) учун маълумотлар базасидан фойдаланилган маъкул.

ХИ БОБ. ОБЪЕКТГА ЙЎНАЛТИРИЛГАН ДАСТУРЛАШ

Бу бобда РНР нинг объект моделини кўриб чиқамиз. Классларни аниқлаш ва ишлатиш. Класслар ва объектлар ҳақида тушунча, масалалар турлари ва уларни ишлатилиши. Класслар кенгайтмаси усуллари, класс конструкторлари ва бошқалар. Бундан ташқари РНР5 да қўйилган объект моделларининг янги имкониятлари билан танишамиз.

РНР қуйидаги объектга йўналтирилган имкониятларни қўлламайди:

- кўп миқдордаги ворисликни(множественное наследование);
- конструкторларни автоматик чақиришни;
- мавҳум классларни;
- ортиқча методларни;
- ортиқча операторларни;
- виртуал функциялардан очик ва ёпиқ фойдаланиш;
- деструкторларни;
- кўп шакллиликни.

12.1. Класслар ва объектлар

Объектга йўналтирилган дастурлашнинг асосий тушунчаларидан бири бу – **класслар ва объектлар**.

Объект – бу дастурда қўлланладиган тушунча ёки бирор физик предмет ҳақидаги маълумотларни ўзида жамлаган структуралашган ўзгарувчи.

Класс – эса бу объектларнинг тавсифи ва улар устида бажариладиган ҳаракатлар.

Класслар объектга йўналтирилган дастурлашни синтаксис базасини яратади. Класслар дастурда ишлатиладиган аниқ нусхаларни яратадиган шаблон сифатида ифодаланади. Класслар нусхаси объект деб аталади. Классларни маълумотлар типи сифатида, объектларни – эса ўзгарувчилар сифатида қараш мумкин. Дастур бир вақтнинг ўзида худди бутун типдаги бир нечта ўзгарувчи билан ишлагани каби битта класснинг бир нечта объектлари билан ишлаши мумкин.

Классларни эълон қилиш `class` калит сўзи билан бошланади(худди, функцияни эълон қилиш `function` калит сўзи билан бошлангани каби).

Класснинг синтаксиси:

```
<?
class Class_nomi
{
}
?>
```

Классларни номлашда бир нечта мулоҳазаларга эътибор бериш керак. Класс номи РНР тилининг объектларни номлаш қоидаcига тўғри келиши керак, лекин дастурчилар томонидан захира сифатида сақланган номлар рўйхати мавжуд. Биринчи навбатда бу номлар, «_» белгиси билан бошланади. Бундан ташқари, РНР да бир неча маротаба ишлатиладиган `stdClass` деб захираномланган ном. Класслар ва функцияларни яратишда бундай номларни ишлатиш мумкин эмас.

Класслар фойдали бўлиши учун **атрибутлар** ва **методларга** эга бўлиши керак. Класс объектларининг ҳар бир атрибути ўзгарувчиларни эълон қилиш йўли билан аниқланган класс ичида var калит сўзи ёрдамида эълон қилинади. Атрибутлар PHP да қўлланиладиган ихтиёрий маълумотлар типига бўлади; улар унча катта фарқи бўлмаган ўзгарувчилар сифатида қаралади. Қуйидаги код Class_nomi класс ичида иккита атрибутни яратади:

```
<?
class Class_nomi
{
    var $atribut1;
    var $atribut2;
}
?>
```

Атрибутлар эълон қилингандан сўнг методлар эълон қилинади, бу функцияни эълон қилишга ўхшаш. Методлар функция эълон қилиш йўли билан аниқланган класс ичида яратилади.

Қуйидаги код ҳеч қандай ҳаракатни бажармайдиган иккита методдан ташкил топган Class_nomi классини яратади. metod1 методи ҳеч қандай параметрни қабул қилмайди, metod2 методи эса иккита параметрни қабул қилади.

```
<?
class Class_nomi
{
    function metod1()
    {
    }

    function metod2($param1, $param2)
    {
    }
}
?>
```

Аниқланган класс ичида жорий класснинг вакилига мурожаат этиш, яъни ўзгарувчини метод ичида атрибутларга жўнатиш учун \$this калит сўзи ишлатилади.

Бу синтаксисни ишлатаётганда икки хил ҳолатга аҳамият бериш керак:

- методга жўнатилаётган атрибут функция параметри сифатида узатилиши керак эмас;
- доллар \$ белгиси атрибут номи олдида(оддий ўзгарувчи каби) эмас, балки \$this ўзгарувчисидан олдин туриши керак.

```
<?
class Class_nomi
{
    var $atribut;
    function metod($param)
    {
        $this->atribut=$param;
        echo $this->atribut;
    }
}
?>
```

Баъзи дастурлаш тилларида маълумотларни ҳимоялаш мақсадида атрибутларга кириш чегаралаб қўйилади. PHP да бундай хусусият мавжуд эмас, шунинг учун барча атрибутлар ва методлар класс ташқарисида қўлланилмайди.

```
<?
class Class_nomi
{
    var $atribut;
}

$a = new Class nomi ();
$a->atribut = "qiymat" ;
echo $a->atribut;

?>
```

класс ташқарисида атрибутларга кутилмаган мурожаат бу унча яхши бўлмаган фикр.

```
<?
class Class_nomi
{
    var $atribut ;
    function get_atribut()
    {
        return $this->atribut;
    }
    function set_atribut ($yangi_qiymat)
    {
        $this->atribut = $yangi_qiymat;
    }
}

?>
```

Бу код \$atribut атрибутларга кириш учун функцияни тасвирлайди. Унда get_atribut() функцияси аниқланган бўлиб, \$atribut нинг қийматини қайтаради ва set_atribut() функцияси \$atribut га янги қийматни беради.

12.1 – мисол.

```
<?
class Web_page
{
    var $bgcolor;
    function getbColor()
    {return $this->bgcolor;}}
    function setbColor($color)
    { $this->bgcolor = $color;}

?>
```

12.2. Ўзгарувчиларни инициализацияси (лавҳаси)

Баъзан класс вакили яратилиши билан класс атрибутларини қийматга тенглаштириш зарур бўлади. Класс атрибутларини бошланғич қийматлари масалалари учун махсус иккита стандарт метод мавжуд. РНР4 да қийматни var оператори ёки конструктор функциялари ёрдамида инициализация қилинади. var ёрдамида фақатгина константа(ўзгармас) қийматларни инициализация қилиш мумкин. Қиймати константа бўлмаган масалалар учун конструктор функциялари ишлатилиб, бу функция объект классда лойиҳалаштирилганда автоматик тарзда чақирилади. Конструктор функцияси аниқланган барча класслар номига мос тушувчи номга эга бўлиши керак.

Мисол келтирамиз. Талабалар портали саҳифасини яратишда саҳифа элементларини атрибутлар сифатида эълон қиламиз. Саҳифа сарлавҳаси, калит сўзлари, меню тугмаларини қийматга тенглаштирамиз.

РНР4 да классни қуйидагича ёзиш тўғри ҳисобланмайди:

```
<?
class Page
{
//Page klassining atributlari
var $content = $newcontent;
var $title = "Talabalar portali";
var $keywords = "Talabalar, o`quv jarayoni, maqolalar, darsliklar, baholar";
var $buttons = array( "Asosiy" => "home.php",
                     "Ta`lim resurslari" => "resurs.php",
                     "Talabalar uchun" => "talaba.php",
                     "Portal haqida" => "about.php"
                     );
// klass atributlariga qiymatni tenglash
...
}
?>
```

Parse error: parse error, unexpected T_VARIABLE in **D:\xampplite\htdocs\learn_php\learn.php** on line **3**

Шунинг учун дастурни РНР4 да қуйидагича ёзилиши, керак бўлган натижани беради:

```
<?
class Page
{
//Page klassining atributlari
var $content ;
var $title = "Talabalar portali";
var $keywords = "Talabalar, o`quv jarayoni, maqolalar, darsliklar, baholar";
var $buttons = array( "Asosiy" => "home.php",
                     "Ta`lim resurslari" => "resurs.php",
                     "Talabalar uchun" => "talaba.php",
                     "Portal haqida" => "about.php"
                     );
// Page klassining metodlari
function SetContent($newcontent)
{
    $this->content = $newcontent;
}
...
}
?>
```

РНР3 ва РНР4 да конструкторлар турлича ишлайди. РНР3 да **функция**

класс билан бир хил номга эга бўлса, конструктор сифатида қаралади, РНР4 да эса **функция аниқланган класс** билан бир хил номга эга бўлса, конструктор сифатида қаралади. РНР5 да класс конструкторлари `_construct` каби номланади. Бундан ташқари РНР5 да объектлар ўчирилишидан олдин автоматик тарзда чақириладиган функция деструкторлари ҳам ишлатилади. РНР5 да функция деструкторлари `_destruct` деб номланади.

12.3. Объект яратиш ва у билан ишлаш

Класс яратилгандан сўнг класс аъзоси бўлган алоҳида элемент – объектни яратиш зарур бўлади. Бу жараён классни нусхасини яратиш деб аталади. Класс – бу маълумотларни битта типдаги – объект типдаги ёзилиши. Класслар реал ўзгарувчилар учун шаблон демакдир. Объект яратишда, классни ёзишда аниқланган барча атрибутларни олиш ва барча методларни қабул қилиш мумкин.

Синтаксиси:

\$obyekt_nomi->atribut_nomi ёки

\$obyekt_nomi-> metod_nomi(argumentlar ro`yhati)

Эслатиб ўтамиз, атрибут ёки методдан олдин \$ белгиси қўйилмайди.

Объектлар new оператори ёрдамида яратилади.

Қуйидаги код `Class_nomi` классини шу класс типдаги учта объект билан бирга яратади:

```
<?
class Class_nomi
{
    function metod($param)
    {
        echo "Konstruktorga $param parametri bilan murojaat etiladi";
    }
}
$a= new Class_nomi("Birinchi");
$b= new Class_nomi("Birinchi");
$c= new Class_nomi("Birinchi");

?>
```

Хар бир объект яратилганда конструкторга мурожаат этилади ва юқоридаги натижани экранга чиқазади.

Ўзимиз учун тасаввур қилиб кўрамиз, объект бу файллар системасидаги директория, унинг характеристикаси эса – бу директориядаги файл. Хар бир директорияда бир хил файл туриши мумкин, лекин улар турли директорияларда сақланганлиги учун турли хил ҳисобланади. Худди шундай агар атрибутлар ва методлар турли объектларда қабул қилинсалар, улар ҳам турли ҳисобланади. Юқори ўриндаги директориядан керакли файлни олмоқчи бўлсак, бу файлга тўлиқ йўлни(путь) ёзишимиз керак.

```
<?
class Class_nomi
{
    function metod1()
    {
    }
    function metod2($param1, $param2)
    {
    }
}

?>
```

`Class_nomi` типдаги \$a номли объектни қуйидагича яратамиз:

```
$a=new Class_nomi;
```

Энди методларга мурожаат этиш мумкин, методларга мурожаат этиш функцияга мурожаат этиш каби: метод номи, қавс ичида керакли

параметрлар жойлашади. Бироқ бу методлар функцияда эмас, объектда яратилади, шунинг учун улар жойлашган объект кўрсатилиши зарур. Объект номлари куйидагича ишлатилади:

```
$a->method1 ( ) ;
$a->method2 (12, "test") ;
```

Агар метод бирор нимани қайтарса, қайтарилган маълумотларни куйидагича олиш мумкин:

```
$x = $a->method1 ( ) ;
$y = $a->method2 (12, "test");
```

Юқори ўриндаги директория PHP да глобал ўзгарувчилар майдони дейилади, унга йўл эса `—>` ажратувчиси ёрдамида кўрсатилади. У ҳолда, `$a->atribut` `$a->` атрибут номлари иккита турли ўзгарувчини билдиради. PHP да ўзгарувчилар олдида фақат битта доллар белгиси қуйилади, шунинг учун `$a->$atribut` деб ёзиш мумкин эмас. Бу конструкция `$a` объектининг атрибут атрибутига қандай мурожаат этишини эмас, балки `$atribut` ўзгарувчиси номидаги атрибутга мурожаат этишни билдиради(масалан, `$a->`”).

12.2 – мисол. Атрибут қийматларини ўрнатиш

```
<?php
$a->atribut= "Internet ga kirish";
// obyekt atributi qiymatini shunday o`rnatish mumkin
$a->$atribut = "Internet ga kirish";
// obyekt atributi qiymatini bunday o`rnatish mumkin emas
$х= "atribut";
$a->$х= "Internet ga kirish";
// obyekt atributi qiymatini shunday o`rnatish mumkin
?>
```

Класс яратаётган вақтда ушбу класснинг объекти номи қандай бўлишини билиб бўлмайди, бундан ташқари объект кўпгина номга эга ва барчасининг номи турлича бўлиши мумкин. Шунга кўра класс аниқланаётганда унинг ичида объектга қандай мурожаат этишни билиш қийин. Класс аниқланаётганда унинг ичида ўзгарувчи ва функцияга мурожаат этиш рухсатига эга бўлиш учун, `$this` псевдоўзгарувчини ишлатиш керак.

`$this` ўзгарувчиси мурожаат этилаётган метод объект нусхасига жўнатилади. Ихтиёрий классда бир нечта объект нусхалари мавжуд бўлиши мумкин, `$this` ўзгарувчиси бу ҳолда жорий объектдаги атрибутга жўнатилади.

Масалан, `$this->atribut` берилган класснинг жорий объектини атрибут атрибутини қийматини чиқазади.

Масалан, **12.1 – мисолдаги** `Web_page` классини объекти куйидаги буйруқ ёрдамида яратилади:

```
$some_page = new Web_page;
```

`$some_page` номи янги объект `Web_page` классига ўтказилган метод ва атрибутлар шахсий тўпламига эга бўлади. Ушбу объектда жойлашган `$bgcolor` атрибутини қийматини ўзгартириш учун классда аниқланган `setbColor()` методидан фойдаланиш мумкин.

```
$some_page->setbColor("black");
```

PHP объект ва атрибут номи билан кўрсатилган атрибут қийматларини

олиш учун имкон яратади:

```
$some_page->bgcolor;
```

```
<?
class Web_page
{
var $bgcolor;
function setbColor($color)
{ $this->bgcolor = $color;}
function getbColor()
{return $this->bgcolor;}}
$some_page = new Web_page;
$some_page->setbColor("black");
$some_page->bgcolor;
?>
```

12.4. Ворислик(наследование)

Оддий ва иерархик ворислик

Ворислик механизми – объектга-йўналтирилган ёндошувнинг энг муҳим қисми. Ворислик дастурлашда ягона муҳим йўл, у кодларни бир неча бор нусхалашни олдини олади ва шу билан бирга маълумотлар структураси билан ишлашга имкон яратади.

Мисол ёрдамида моҳиятини тушунтирамиз. Web сайт саҳифалари элементларини кўриб чиқайлик. Сайтнинг бош саҳифаси сайт сарлавҳаси, калит сўзлар, менюлар, саҳифанинг асосий қисмдаги матн, сайтнинг қуйи қисмидаги маълумотлар ва ҳ. элементларга эга. Энди сайтнинг талабалар саҳифасидаги маълумотларни қараймиз, талабаларга тегишли менюлар, саҳифанинг асосий қисмидаги матн каби элементлардан иборат. Бироқ, талабалар саҳифаси ҳам сайт сарлавҳаси, калит сўзлар, менюлар, умуман олганда бош саҳифа эга бўлган элементларга эга. Агар ушбу мулоҳазаларни объектга йўналтирилган дастурлаш терминига кўчирсак, у ҳолда айтиш мумкинки, ҳар бири ўзининг атрибути ва методига эга бўлган иккита – **бош саҳифа класс** ва **талаба саҳифаси класс** яратилади. Бундан кўринадики, талаба саҳифаси класс бош саҳифа классини барча атрибутларига эга ва бунга қўшимча равишда ўзига хос хусусиятга эга, шунинг учун **талаба саҳифаси класс** **бош саҳифа класс**ини қисм класс ҳисобланади. Худди шундай агар бош саҳифанинг менюлар қатори бўлса, у ҳолда талабалар саҳифасининг ҳам менюлар қатори бўлиши керак, аксинча эмас. Талабалар саҳифаси классидан ташқари бош саҳифа классига алоқадор кўпгина классларни яратиш мумкин ва уларнинг барчаси бош саҳифа классини қисм класс бўлади. Одатда амалиётда тезда бир қанча проектларда(масалан, бош ёки асосий класс) ишлатиш мумкин бўлган умумий классни аниқлаш ва уни ҳар бир проектнинг ўзига хос талабига кўра(масалан, талабалар класс каби) соддалаштирмоқ кулай. Буни қандай амалга ошириш мумкин? Кенгайтирилган механизм ёрдамида. Ихтиёрий класс бошқасидан кенгайтирилган(тўлиқроқ) бўлиши мумкин. Кенгайтирилган класс у аниқланганда ёзилган атрибут ва методдан ташқари таянч (асосий)класснинг барча атрибут ва функцияларига эга бўлади. Ҳозирги мисолда талабалар саҳифаси класс – кенгайтирилган класс, бош саҳифа класс эса- таянч класс ҳисобланади. Классдан мавжуд функция ва атрибутларни ўчириш мумкин эмас, уни фақатгина кенгайтириш мумкин. РНР 4 да кенгайтирилган класс фақатгина битта таянч классга боғлиқ, шунинг учун РНР кўп миқдордаги ворисликни қўлламайди.

РНР да класслар **extends** калит сўзи ёрдамида кенгайтирилади.

Синтаксиси:

```
<?
class Class_nomi2 extends Class_nomi1
{
    var $atribut2;
    function metod2()
    {
    }
}
```

>

Агар Class_nomi1 классы куйидагича эълон қилинган бўлса:

```
<?
class Class_nomi1
{
    var $atribut1;
    function metod1()
    {
    }
}

?>
```

Class_nomi2 типдаги объектнинг атрибут ва методларига мурожаат этганда :

```
<?
$b=new Class_nomi2();
$b->metod1();
$b->atribut1=10;
$b=metod2();
$b->atribut2=10;
?>
```

Эътибор беринг, Class_nomi2 классы Class_nomi1 классининг кенгайтирилгани, metod1() методи ва \$atribut атрибутлари Class_nomi1 классда эълон қилинган бўлса ҳам Class_nomi2 классда мурожаат этиш мумкин. Class_nomi1 классининг вориси Class_nomi2 классы Class_nomi1 классининг барча имкониятлари ва маълумотларини олади ва бундан ташқари ўзининг атрибут ва методларига эга бўла олади.

Фақатгина ворислик бир томонлама амалга оширилади.

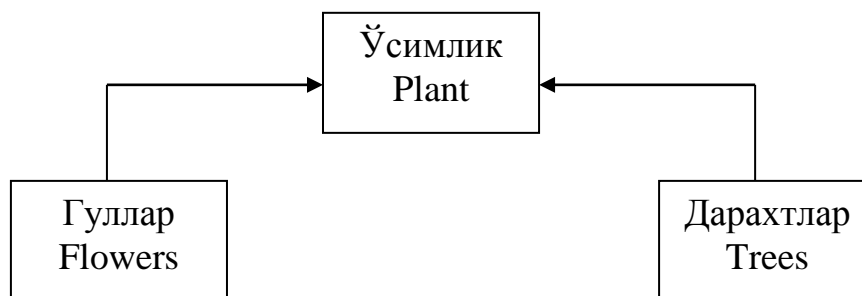
Class_nomi2 классы барча характеристикаларини Class_nomi1 классига ўтказди. Бундан ташқари ворислик кодларини бир неча бор ишлатилиши яна бир муҳим асосга – дастурни ўзгартиришда юзага келадиган хатолар эҳтимолини камайтиришга имкон яратади.

Кенгайтирилган класс конструкторига мурожаат этилганда таянч класс конструкторига автоматик тарзда мурожаат этилмайди.

Яратилган Class_nomi1 ва Class_nomi2 класслари ўртасида ўрнатилган муносабат – таянч-авлодлар (родитель-потомок) муносабати дейилади. Class_nomi1 – классы таянч класс, унинг авлодлари эса Class_nomi2 классы каби кенгайтириш ёрдамида унга асосланиб яратилган класслар. Ихтиёрий класс таянч класс бўлиши ва унга асосланиб унинг авлод класслари яратилиши мумкин.

Классларни аниқлаш тартиби аҳамиятга эга. Биринчи кенгайтирилган Class_nomi2 классини, сўнг Class_nomi1 классини яратиш мумкин эмас. Класслар қандай аниқланса, шу тартибда яъни класс аниқланиб, кейин унинг кенгайтирилган формаси яратилади.

12.3 - мисол



12.1- расм. Иерархия моделига асосланган класс

12.1- расмдаги иерархия моделига асосланган классни берилиши:

```

<?
// O`simliklar
class Plant
{
var $name;
var $age;
function setAge($mph)
{
    $this->age = $mph;
}
function getAge()
{
    return $this->age;
}
}
// Gullar
class Flowers extends Plant
{
var $FColor;
function setFColor($col)
{
    $this->FColor = $col;
}
function getFColor()
{
    return $this->FColor;
}
}
// Daraxtlar
class Trees extends Plant
{
var $growth;
function setGrowth($gro)
{
    $this->growth = $gro;
}
function getGrowth()
{
    return $this->growth;
}
}
// Bu klasslarni obyektлари quyidagicha yaratiladi
$plan = new Plant;
$gul = new Flowers;
?>
    
```

Берилган буйруқ иккита объектни яратади. Биринчи \$plan объекти Plant классига мурожаат этади. Иккинчи \$gul объекти эса, Flowers классига

мурожаат этади ва бу класс Plant классига умумий хусусиятларга эга.

Агар класс бир нечта таянч классларнинг вориси бўлса, бу ҳолат **кўп микдордаги ворислик** (множественное наследование) дейилади. Афсуски, PHP **кўп микдордаги ворисликни** қўлламайди.

12.5. Кўп даражали ворислик(Многоуровневое наследование)

Дастурни ҳажмини ошириш ва мураккаблаштириш **кўп даражали ворислик** зарурлигига олиб келиши мумкин. Класс ўзини хусусиятларини бошқа классга узатади, ўзини навбати билан бу класс ҳам ўз хусусиятларини учинчи бир классга узатади. **Кўп даражали ворисликда** таянч классни кенгайтириб класс ҳосил қилиш ва ҳосил бўлган классни яна кенгайтириб бошқа бир класс ҳосил қилиш мумкин. **Кўп даражали ворислик** дастурнинг модулли структурасини ривожлантиришда оддий бошқарувни ва аниқроқ мантикий структурани таъминлайди.

```
<?
class Plant {
//atributlarni elon qilish...
//metodlarni elon qilish...
}
class Street extends Plant {
//atributlarni elon qilish...
//metodlarni elon qilish...
}
class Home extends Street {
//atributlarni elon qilish...
//metodlarni elon qilish...
}
$nis = new Home;
?>
```

\$nis объекти Plant, Street ва Home классларининг барча атрибутлари ва методларидан ташкил топган.

Кенгайтирилган(ворис) класс таянч класснинг барча хусусиятларини занжир шаклида олади. Таянч класс конструкторларига эса ворис-класс объекти яратилганда автоматик тарзда чақирилмайди. Бу конструкторлар ворис-классларни метод кўринишида чақиради.

12.6. Конструкторлар

Конструктор бир қанча атрибутлар бераётган қийматлар методи каби тасвирланади (бундан ташқари бошқа методларни ҳам чақириш мумкин). Конструкторлар янги объектлар яратилганда автоматик тарзда чақирилади. Конструктор методи номи мавжуд класс номи билан мос тушиши керак.

12.4-мисол.

```
<?
class Web_page
{ var $bgcolor;
function Web_page($color)
{ $this->bgcolor = $color;}}
// Web_page klassi konstruktoriga murojaat etish
$page = new Web_page("brown");
?>
```

Конструкторлар бир этапда объект яратиш ва уни атрибутларини инициализациялаш амалларини бажариш учун имкон яратади.

Объект оддий яратилганда ҳам конструкторларни чақириш мумкин, лекин уни атрибутлари инициализацияланмайди:

```
$page = new Web_page;
```

Классда аниқланган конструкторлар ёрдамида ҳам объект яратиш мумкин. Бу ҳолда Web_page класс объекти яратилади ва уни атрибутлари қиймати bgcolor га тенглаштирилади:

```
$page = new Web_page("brown");
```

12.7. Деструкторлар

Аввал эслатиб ўтганимиздек, PHP деструкторларни тўғридан-тўғри қўлламайди. Шунга қарамай, PHP нинг `unset()` функциясига мурожаат этиб, деструктор ишига осонгина тақлид қилиш мумкин. Бу функция мавжуд ўзгарувчиларни ўчиради ва у системада ишлатаётган ресурсларни чиқазади. `unset()` объектлар билан ҳам ўзгарувчилар билан ишлагани каби ишлайди. Фараз қилайлик, `$Web_page` объекти билан ишляпмиз. Бу объект иши тугатилгандан сўнг функция чақирилади:

```
unset($Web_page);
```

бу буйруқ хотирадан `$Web_page` нинг барча таркибини ўчиради. Инкапсуляция хусусиятига амал қилган ҳолда, `unset()` га мурожаат этишни `destroy()` номли методга жойлаштириш керак ва сўнг унга мурожаат этиш керак:

```
$Website->destroy();
```

Фақатгина катта ҳажмдаги хотира талаб қиладиган объектлар билан иш олиб боргандагина деструкторга мурожаат қилиш зарурати туғилади. Сабаби одатда php сценарий бажарилиб бўлгандан сўнг барча ўзгарувчи ва объектлар автоматик тарзда хотирадан тозаланади.

12.8. Мавҳум класслар

Баъзи ҳолатларда объекти умуман яратилмайдиган классни яратиш қулай бўлади (бу класс кенгайтирилган классни яратиш учун таянч класс сифатида керак ҳолос). Бундай класслар мавҳум (абстракт) класслар дейилади. Мавҳум класслар одатда дастур яратувчилари таянч абстракт класслардан ясалган барча классларни баъзи функционал имкониятларини қўллашни таъминлаш керак бўлган ҳолатларда қабул қилинадилар.

PHP да тўғридан-тўғри мавҳум классларни қўллаш инкор этилади, лекин оддий айланма ечим мавжуд – <мавҳум> классларда конструкторни аниқлаш ва унга die() чакирувини қўшиш кифоя. Реал объектни тасаввур қилиш учун бу класслардан кенгайтирилган классни яратиш керак бўлади. Бинобарин, Home ва Street класслари объектларини имкон қадар яратишни бартараф этиш учун уларнинг конструкторларига die() чакирувини қўшиш керак.

```
<?
class Plant
{
    Atributlarni elon qilish...
    function Plant() {
        die ("Cannot create Abstract Plant class!");
    }
    Boshqa metodlarni elon qilish...
}
class Street extends Plant
{
    Atributlarni elon qilish...
    function Land() {
        die ("Cannot create Abstract Street class!");
    }
    Boshqa metodlarni elon qilish...
}
class Home extends Street
{
    Atributlarni elon qilish...
    Boshqa metodlarni elon qilish...
}
?>
```

Бу мавҳум классларни нусхасини яратишга уриниш хатолар ҳақида маълумот чиқазиш ва дастурларни якунлашга олиб келади.

12.9. Методларни ортиқчалиги(Перегрузкой методов)

Бир хил номли, лекин турли миқдордаги ёки параметр типдаги бир нечта методларни аниқланишига - методларни ортиқчалиги(Перегрузкой методов) дейилади. Абстракт класслар каби, PHP бу имкониятларни қўлламайди, лекин буни оддий айланиб ўтиш ечими мавжуд.

12.5 – мисол.

```
<? class Sahifa
{
var $bgcolor;
var $textcolor;
function Sahifa()
{
// jo`natilgan argumentlar sonini aniqlashva
// kerakli nomdagi metodni chaqirish
$name = "Sahifa".func_num_args();
if ( func_num_args() == 0 ) :
    $this->$name();
else :
    $this->$name(func_get_arg(0));
endif;
}
function Sahifa0() {
    $this->bgcolor = "white";
    $this->textcolor = "black";
    print "Created default page";
}
function Sahifa1($bgcolor) {
    $this->bgcolor = $bgcolor;
    $this->textcolor = "black";
    print "Created custom page";
}
}
$html_page = new Sahifa("red");
?>
Natija:Created custom page
```

Албатта, бундай натижани ортиқча уриниш деб аташ мумкин, лекин бу ОЙД нинг асосий аспектларисиз ишлай олмайдиганлар учун жуда мос усулдир.

12.10. Масаланинг қўйилиши

12.1 – Масала. Талабалар порталининг саҳифаларини қайта ишлашни объектга йўналтириш йўли билан ҳал этамиз.

Класс номлари мантиқий номлар бўлиши кераклиги талаб этилади. Шунинг учун саҳифани тасвирловчи классни Page деб номлаймиз.

```
class Page
{
}
```

Класслар бир нечта атрибутларни талаб қилади. Саҳифалар биридан иккинчисига алмашганда алмаштирилиши керак бўлган элементларни класс атрибутлари сифатида аниқлаймиз. Саҳифанинг асосий таркиби HTML дескрипторлар ва матнларни \$content деб номлаймиз. Бу атрибут аниқланган класс ичида эълон қилинади:

```
var $content;
```

худди шундай саҳифа сарлавҳасини сақлаш учун атрибут аниқлаймиз. Сарлавҳа сатри ҳам ўзгариши мумкин, шунинг учун унинг номини одатга кўра қуйидагича эълон қиламиз.

```
var $title = "Talabalar portali";
```

Кўпгина тижорат порталларининг WEB саҳифалари қидирув механизмларига ёрдам сифатида метадескрипторларни ҳам ўз ичига олади. Метадеккрипторлар ҳам турли саҳифаларда турлича бўлади, шунинг учун уни ҳам одатга кўра ёзамиз.

```
var $keywords = "Talabalar, o'quv jarayoni, maqolalar, darsliklar, baholar";
```

Табалар саҳифасини тасвирлаганимизда иштирок этган меню тугмалари, фойдаланувчилар чалғимаслиги учун улар барча саҳифаларда ўзгармай қолиши зарур. Бироқ уларга кичкина ўзгартиришлар киритилиши мумкин, улар ҳам атрибут сифатида эълон қилинади. Кейинчалик тугмалар сони ўзгариши мумкин, шунинг учун уларни массив сифатида эълон қиламиз ва уларда тугма номи ва WEB манзилини сақлаймиз:

```
var $buttons = array( "Asosiy" => "home.php",
                     "Ta'lim resurslari" => "resurs.php",
                     "Talabalar uchun" => "talaba.php",
                     "Portal haqida" => "about.php",
                     "Admin" => "admin.php"
);
```

Аниқланган функционал ҳаракатларни бажариш учун классларда методларни яратиш талаб этилади. Уларни аниқлашни ўрнатиш учун рухсат берувчи функцияларни аниқлаш ва аниқланган атрибутларни қийматларини олишдан бошланади.

```
function SetContent($newcontent)
{
    $this->content = $newcontent;
}
```

бу класснинг асосий вазифаси – HTML саҳифани тасвирлаш, бунинг учун эса функция талаб этилади. Бу функция display()деб номланиб, қуйидагича бўлади:

```
function Display ()
{
    echo "<html>\n<head>\n" ;
    $this -> DisplayTitle () ;
}
```

```

$this -> DisplayKeywords() ;
$this -> DisplayStyles () ;
echo "</head>\n<body>\n" ;
$this -> DisplayHeader();
$this -> DisplayMenu($this->buttons) ;
echo $this->content ;
$this -> DisplayFooter() ;
echo "</body>\n</html>\n" ;
}

```

Бутун классни бошқа файлларга камраб олиш мумкин бўлиши учун page.inc файлида сақлаймиз.

12.6-листинг. page.inc файли —Page классни Талабалар портални саҳифасини яратишнинг оддий мукамал усулини таъминлайди.

```

<?
class Page
{
//Page klassining atributlari
var $content;
var $title = "Talabalar portali";
var $keywords = "Talabalar, o'quv jarayoni, maqolalar, darsliklar, baholar";
var $buttons = array( "Asosiy" => "home.php",
                      "Ta'lim resurslari" => "resurs.php",
                      "Talabalar uchun" => "talaba.php",
                      "Portal haqida" => "about.php",
                      "Admin" => "admin.php"
                      );
// Page klassining metodlari
function SetContent($newcontent)
{
    $this->content = $newcontent;
}
function SetTitle($newtitle)
{
    $this->title = $newtitle;
}
function SetKeywords ($newkeywords)
{
    $this->keyWords = $newkeywords ;
}
function SetButtons ($newbuttons)
{
    $this->buttons = $newbuttons ;
}
function Display ()
{
    echo "<html>\n<head>\n" ;
    $this -> DisplayTitle () ;
    $this -> DisplayKeywords() ;
    $this -> DisplayStyles () ;
    echo "</head>\n<body>\n" ;
    $this -> DisplayHeader();
    $this -> DisplayMenu($this->buttons) ;
    echo $this->content ;
    $this -> DisplayFooter() ;
    echo "</body>\n</html>\n" ;
}
function DisplayTitle ()
{
    echo "<title> $this->title </title>";
}
function DisplayKeywords()
{
}

```

```

echo "<META name=\"keywords\" content=\"\${this->keywords}\">";
}
function DisplayStyles ()
{
?>
<style>
h1 (color: white; font-size :24pt; text-align: center; font-
family : arial , sans-serif)
.menu { color : white ; font-size :12pt; text-align : center;
font-family : arial , sans-serif ; font-weight : bold)
td (background: black)
p (color: black; font-size :12pt; text-align: justify ; font-
family : arial , sans-serif)
p. foot (color: white; font-size :9pt; text-align : center ;
font- family: arial, sans-serif ; font-weight: bold)
a: link, a: visited, a: active (color: white)
</style>
<?
}
function DisplayHeader()
{
?>
<!-- page header -->
<table width="100%" cellpadding = 12 cellspacing = 0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "images/logo.png "></td>
<td>
<h1>Talabalar portali</h1>
</td>
<td align = right><img src = "images/logo.png "></td>
</tr>
</table>
<!-- menu -->
<?
}
function DisplayMenu($buttons)
{
echo "<table width = \"100%\" bgcolor = white cellpadding = 4 cellspacing
= 4>\n";
echo "<tr>\n";
//tugmalar o`lchamini hisoblash
$width = 100/count($buttons) ;
while (list($name, $url) = each ($buttons) )
{
$this -> DisplayButton($width, $name, $url,
!$this->IsURLCurrentPage($url) ) ;
}
echo "</tr>\n" ;
echo "</table>\n" ;
}
function IsURLCurrentPage($url)
{
if(strpos($GLOBALS["SCRIPT_NAME"], $url)==false)
{
return false;
}
else
{
return true;
}
}
function DisplayButton($width, $name, $url, $active = true)
{
if ($active)

```



```

{
echo "<td width = \"\$width%\">
    <a href = \"\$url\">
    <img src = \"images/plus.png\" alt = \"\$name\" border =0></a>
    <a href = \"\$url\"><span class=menu>$name</span></a></td>";
}
else
{
echo "<td width = \"\$width%\">
<img src = \"images/minus.png\">
<span class=menu>$name</span></td>" ;
}
}
function DisplayFooter()
{
?>
<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr>
<td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@mail.ru">fmahbuba@mail.ru</a>manziliga jo`nating </p>
</td>
</tr>
</table>
<?
}
?>

```

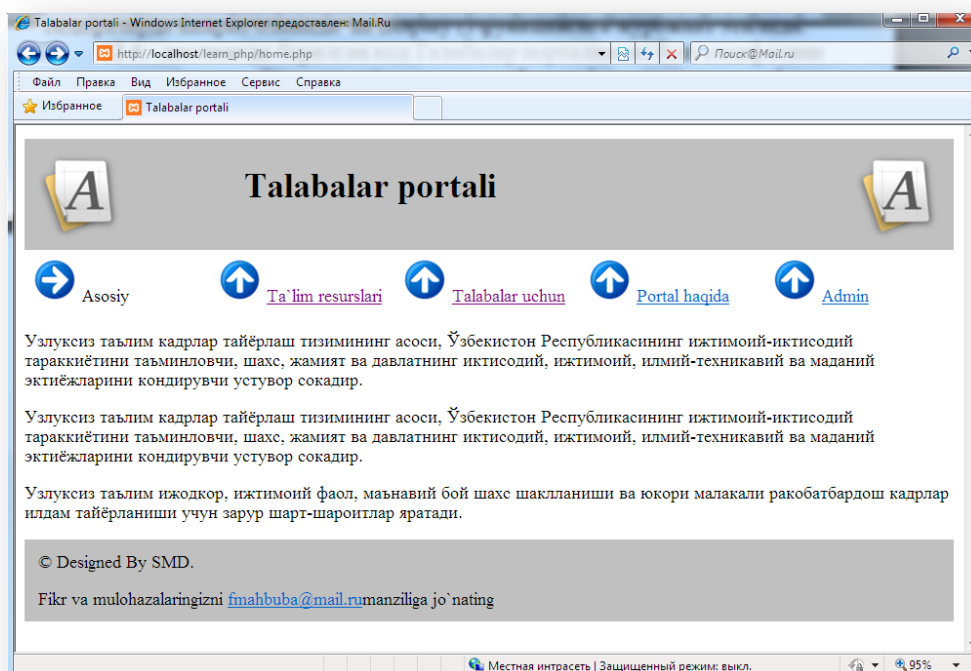
12.6 - листингни кўриб чиқишда DisplayStyles(), DisplayHeader() ва DisplayFooter() функцияларига эътибор қаратинг, улар катта блоклардан иборат HTML-кодларни бирор-бир PHP-кодлар ёрдамида қайта ишламаяпти. Шунинг учун функция ичида тугалловчи PHP-дескриптори (?>)ни ишлатдик, HTML-кодларни киритиб, сўнг яна очилувчи PHP-дескрипторини (<?) ишлатиб PHP га қайтдик.

Ушбу классда яна иккита метод аниқланган. DisplayButton() методи ягона меню тугмаларини чиқазади. Агар тугма жорий саҳифани кўрсатаётган бўлса, унинг ўрнида актив бўлмаган тугма тасвирланади. Бу саҳифанинг ягона ташқи кўринишини тасвирлайди ва фойдаланувчиларга қайси саҳифада турганлигини визуал тарзда аниқлашга ёрдам беради.

IsURLCurrentPage() методи жорий саҳифадаги Internet-манзил тугмаси билан боғлиқлигини кўрсатаяптими ёки йўқлигини аниқлайди. Бу ерда сатрлар билан ишловчи strpos() функцияси серверда ўрнатилган ўзгарувчилардан бирортасида берилган Internet-манзил мавжудлигини аниқлаш учун ишлатилади. strpos(\$GLOBALS["SCRIPT_NAME"], \$url) оператори ёки агар \$url ўзгарувчисида сақланаётган сатр ички глобал ўзгарувчи(SCRIPT_NAME)да жойлашган бўлса, номерни акс ҳолда false қайтаради.

Ушбу классни саҳифаларда ишлатиш учун page.inc файлини саҳифаларда қамраб олинади ва Display () функциясига мурожаат этилади.

12.6 - листингда берилган код Талабалар порталининг бош саҳифасини яратади ва аввалги бўлимда кўрсатилган Web саҳифага ўхшаш саҳифани экранга чиқазади(12.2 - расм).



12.2-расм. Талабалар порталнинг Web саҳифасини класслар ёрдамида яратиш

12.6 –листинг. home.php — бу Page классини ишлатувчи бош саҳифа

```
<?
require("page.inc") ;
$homepage = new Page( );
$homepage -
> SetContent ("<p>Ўзлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекист
он Республикасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-
техникавий ва маданий эктиёжларини кондирувчи устувор соқадир. </p>
<p>Ўзлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Республикаси
нинг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-техникавий ва маданий эктиёжларини кондирувчи устувор соқадир.
</p>
<p>Ўзлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шаклланиши ва юк
ори малакали рақобатбардош кадрлар илдам тайёрланиши учун зарур шарт-
шароитлар яратади.</p>
");
$homepage -> Display ( ) ;
?>
```

12.6 - листингдаги кодлар қуйидаги амалларни бажаради:

1. require операторини таркибида Page классини аниқланган page.inc файл таркибини қамраб олиш учун ишлатади.
2. Page классини нусхасини \$homepage номи билан яратади.
3. \$homepage объекти ичида SetContent() методини чақиради ва бир қанча маълумотларни ва саҳифада тасвирланиши лозим бўлган HTML-дескрипторларни узатади.
4. \$homepage объекти ичида фойдаланувчи браузерни ойнасида саҳифани тасвирловчи Display() методини чақиради.

12.6-листингдан кўриниб турибдики, Page классини ишлатиб янги саҳифани генерация қилиш унча катта иш талаб этилмайди.

Агар сайтнинг баъзи бўлимларида стандарт саҳифа варианты ишлатилса, page.inc файлидан нусха олиб page1.inc янги файлга қўйилади ва озгина ўзгартириш киритилади. Бу ҳолда page.inc файлини ҳар бир янгилашда ва ўзгартиришда pageZ.inc файлига ҳам ўзгартириш киритиш ёддан чиқмаслиги зарур.

Янги класс яратиш учун ворисликдан фойдаланиш – бир мунча рационал йўл бўлиб, унда Page классини функционал имкониятларини катта қисмини ворис классга ўтказиш мумкин.

Талабалар порталининг Талабалар учун саҳифаси ички меню сатрига эга бўлсин.

Бу масала **12.7-листинг**да ечилади. Page классининг вориси TalabaPage классини яратилади. Бу классда \$row2buttons номли янги массив аниқланиб, у иккинчи сатрда чиқувчи тугмалар ва боғланишлардан таркиб топади.

Бу класс таянч класс каби бўлиб, фақатгина Display() методи қисмига ўзгартириш киритилади.

12.7 – листинг. talaba.php —Page классининг маълумолари ва методларини олади, бироқ Display() методига ўзгартириш киритилади.

```
<?
require ("page.inc") ;
class TalabaPage extends Page
{
    var $row2buttons = array ( "Dars jadvali" => "d_jadval.php" ,
                                "Talabalar reytingi" => "reyting.php" ,
                                "Talabalar hayoti" => "about_talaba.php" ,
                                "Fotogalereya" => "foto.php" ,
                                );

    function Display ()
    {
        echo "<html>\n<head>\n" ;
        $this-> DisplayTitle() ;
        $this-> DisplayKeywords () ;
        $this-> DisplayStyles () ;
        echo "</head>\n<body>\n";
        $this-> DisplayHeader() ;
        $this-> DisplayMenu($this->buttons) ;
        $this-> DisplayMenu($this->row2buttons) ;
        echo $this->content;
        $this-> DisplayFooter() ;
        echo "</body>\n</html>\n" ;
    }
}
$stalaba = new TalabaPage ();
$content ="<p>Bu bo`limda talabalarining o`quv jarayoni, dars jadvallari haqid
a tanishasiz.</p>";
$stalaba -> SetContent ($content) ;
$stalaba -> Display();
?>
```

Display() методи юқори класснинг методига жуда ўхшаш бўлиб, унда фақатгина қўшимча иккинчи DisplayMenu() методини чақириш ва иккинчи меню панелини яратиш учун

`$this -> DisplayMenu($this->row2buttons) ;` сатри мавжуд.

Класс аниқлангандан сўнг класс ташқарисида `ServicesPage` классининг нусхасини яратамиз, аввалгисидан фарқли қиймат ўрнатамиз, ва `Display()` методига мурожаат этамиз.

Натижада 12.3 -расмдаги стандарт саҳифанинг янги кўринишини оламиз.



12.3-расм. Талабалар порталининг Web саҳифасини класслар ёрдамида яратиш

2 ҚИСМ. МАЪЛУМОТЛАР БАЗАСИ БИЛАН ИШЛАШ

ХIII БОБ. Маълумотлар базаси

Бу бобда маълумотлар базаси ва МББТ тушунчалари кўрилади. SQL асосий сўров тили: сатрларни саралаш, қўйиш, ўзгартириш ва ўчириш амаллари ҳамда жадвалларни яратиш, ўзгартириш ва ўчириш амаллари кўрилади. Маълумотлар базаси назариясининг асосий тушунчалари, MySQL маълумотлар базасини бошқариш тизими, у билан ишлаш усуллари ва бу МББТ да SQL сўров тилини амалга ошириш кўриб чиқилади.

13.1. Маълумотлар базаси ҳақида асосий тушунчалар

Маълумотлар базаси ўзи нима? **Маълумотлар базаси** ўзида структуралашган маълумотларнинг тизимини сақлайди. Бу маълумотлар турли хил, яъни оддий сотиб олиниши керак бўлган нарсалар рўйхатидан тортиб то расмлар галереясидаги расмлар рўйхати, бирор ташкилот ходимларининг маълумотлари ёки бўлмаса корпоратив тармоқдаги катта ҳажмдаги маълумотлар бўлиши мумкин.

Маълумотлар базасини яратиш, уларга маълумотларни киритиш ва мавжуд маълумотлар базасидан фойдаланиш, яъни саралаш, ишлов бериш, сақлаш, қидириш ва қабул қилиш каби вазифаларни бажарувчи тизим **маълумотлар базасини бошқариш тизими (МББТ)** деб юритилади.

Реляцион маълумотлар базаси – бу реляцион алгебрани мустаҳкам назарий базисига асосланган, кучли назарий фундаментга қурилган, объектлар ва уларнинг ўзаро алоқалари икки ўлчовли жадвал кўринишида тасвирланган маълумотлар базаси. Маълумотларнинг бундай кўринишда тасвирланиши объектларнинг ўзаро алоқаларини яққол тасвирланишига асос бўлади. Реляцион маълумотлар базаси бугунги кунда энг кўп фойдаланадиган база ҳисобланади. Реляцион маълумотлар базасида маълумотлар бир тўда бўлган ҳолда эмас, балки алоҳида жадвалларда жойлашади. Ва бунинг ҳисобига тезкорлик билан эгилувчанлик хусусиятларига эга.

Реляцион маълумотлар базаси қуйидаги параметрлар билан баҳоланади:

1. Соддалик (Простота).
2. Мосланувчанлик (Гибкость).
3. Аниқлилиқ (Точность), математик аниқ усуллар билан МБ манипуляция қилинади.
4. Махфийлик—(Секретность).
5. Боғлиқлик—(Связанность).
6. Боғлиқсизлик—(Независимость).
7. Маълумотлар билан мураккаб амалларни бажариш тили.

Реляцион маълумотлар базаси машинадан ташқаридаги ахборот таъминотининг тизимлари ёрдамида ҳосил қилинади. Реляцион маълумотлар базаси тизими деганда, соф ахборот ишлари учун, яъни ўзаро боғланган катта ҳажмдаги ахборотни сақлаш, ўзгартириш ва ишлатиш учун махсус тизим тушунилади. Қандайдир битта ахборотнинг жуда кичик ўзгариши турли

жойларда муҳим ўзгаришлар бўлишига олиб келиши мумкин. Инсоният томонидан катта миқдордаги билимнинг тўпланиши, турли хил ахборотларни сақлаш масаласини ҳал қилишни талаб этади. Бунда ахборотларни сақлаш ягона мақсад ҳисобланмайди, балки у керакли маълумотлардан керакли вақтда фойдалана олиш, турли ҳужжатларни қайта ишлашга мўлжалланган. Баъзан МБни ишлатилиш самарадорлигини ошириш мақсадида унинг тузилиши ҳам ўзгартирилиб турилади. Бу ҳолда МБнинг шажаравий ва тармоқли моделлари вужудга келади.

Реляцион маълумотлар базасини бошқариш тизими (РМББТ) қуйидаги муаммоларни ҳал этади:

- Иккилик файлга нисбатан, РМББТ маълумотлар билан тез ишлаш имконини беради.
- РМББТ маълумотларни кидириш учун сўровларни жўнатиш ва натижани чиқаришни тез ва оддий ҳал қилади.
- РМББТ бир вақтда бир неча фойдаланувчига хизмат кўрсатишнинг ички механизмига эгадир.
- РМББТ маълумотларга паралел мурожаат қилишни таъминлайди, яъни бир вақтни ўзида бир нечта жадвалга мурожаат қила олади.
- РМББТ рухсатномалар бериш ва таъминлашнинг ички тизимларига эга.
- РМББТ ваколат(привилегия)ларни қўллайдиган тизим.

Шуни таъкидлаш лозимки, ҳозирги вақтда деярли барча МББТлари, асосан, реляцион моделлар асосида ташкил қилинмоқда.

Қуйидаги жадваллардан ташкил топган ўқув маълумотлар базасини лойиҳасини тузиб чиқамиз.

13.1 - жадвал. student(талаба)

stud_id	surname	name	stipend	kurs	city	birthday	univ_id
1	Islomov	Ikrom	150	1	Andijon	3/12/1992	10
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	10
6	Salimov	Vohid	150	4	Namangan	7/06/1989	22
10	Karimov	Botir	0	2	Jizzax	8/12/1991	10
12	Zoitova	Ozoda	250	2	Termiz	1/05/1991	10
265	Fayziyev	Alijon	0	3	Toshkent	5/11/1989	10
32	Komilov	Farhod	150	5	Samarqand	NULL	14
654	Lazizov	Anvar	200	3	Toshkent	1/12/1991	10
276	Fozilov	Ahmad	200	4	NULL	5/08/1991	22
55	Boltayeva	Vasila	250	5	Toshkent	7/01/1990	10
...

stud_id – сонли код, талаба идентификатори

surname – талабанинг фамилияси

name – талабанинг исми

stipend – талаба оладиган стипендияси

kurs – талаба ўқийдиган курси

city - талабанинг яшаш манзили(вилояти)

birthday – талабанинг туғилган санаси

univ_id - сонли код, талаба ўқийдиган университет идентификатори

13.2 - жадвал. lecturer(ўқитувчи)

lecturer_id	Surname	name	city	univ_id
24	Qobulov	Botir	Toshkent	10
46	Nasriddinov	Islom	Toshkent	10
74	Hasanov	Farhod	Namangan	22
108	Sobirov	Nizom	Jizzax	22
276	Nosirova	Vasila	Toshkent	10
328	Saidazimov	Akrom	Andijon	10
...

lecturer_id – сонли код, ўқитувчининг идентификатори

surname – ўқитувчининг фамилияси

name – ўқитувчининг исми

city – ўқитувчининг яшаш манзили (вилояти)

univ_id – ўқитувчи ишлайдиган университет идентификатори

13.3 - жадвал. subject(ўқув фани)

subj_id	subj_name	hour	semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5
...

subj_id – ўқув фани идентификатори

subj_name – ўқув фани номи

hour – фаннинг ўқув соати

semestr – фан ўтиладиган семестр

13.4 - жадвал. university(университет)

univ_id	univ_name	rating	city
22	Texnika davlat universiteti	34	Toshkent
10	Toshkent davlat pedagogika universiteti	29	Toshkent
11	Andijon davlat universiteti	49	Andijon
32	Termiz davlat universiteti	41	Termiz
14	Samarqand davlat universiteti	36	Samarqand
15	O`zbekiston Milliy universiteti	28	Toshkent
18	Toshkent davlat axborot universiteti	36	Toshkent
...

univ_id – университет идентификатори
 univ_name – университет номи
 rating – университет рейтинги
 city – университет жойлашган вилоят ёки шаҳар

13.5 - жадвал. exam marks(имтиҳон баҳоси)

exam_id	stud_id	subj_id	mark	exam_date
145	12	10	5	12/01/2010
34	32	10	4	23/01/2010
75	55	10	5	05/01/2010
238	12	22	3	17/06/2009
639	55	22	NULL	22/06/2009
43	6	22	4	18/01/2010
.....

exam_id – имтиҳон идентификатори
 stud_id – талабанинг идентификатори
 subj_id – ўқув фани идентификатори
 mark – имтиҳон баҳоси
 exam_date - имтиҳон санаси

13.6 - жадвал. subj_lect(ўқитувчининг ўқув дисциплинаси)

lecturer_id	subj_id
24	24
46	46
74	74
108	108
276	276
328	328
...

lecturer_id – ўқитувчининг идентификатори
 subj_id – ўқув фани идентификатори

Жадваллар. Реляцион моделлар асосига қурилган маълумотлар базаси жадвал кўринишида акс этади. Бундай жадвалдаги устунлар *майдон* деб, сатрлар эса *ёзув* деб аталади. Жадвал номи – *student (талабалар)*, ҳар бирида аниқ маълумотлар жойлашган бир қанча устунлар ва талаба ҳақидаги маълумот жойлашган сатрлардан иборат.

Масалан, 13.1-жадвал.

Устун — маълумотларни ташкил этишнинг оддий бирлиги бўлиб, маълумотнинг алоҳида, бўлинмас бирлигига эгалик *реквизит* мос келади.

Демак, майдон МБнинг асосий тузилмали элементи бўлиб, қуйидаги параметрлар билан ифодаланади:

- узунлиги (белги ва символларда ифодаланиб, байтларда ўлчанади),
- номи (майдоннинг ўзига хос алоҳида хусусияти);
- имзо (подпись) (устун сарлавҳаси ҳақида маълумот).

Устунлар

Жадвалдаги ҳар бир майдон(устун) ўзининг уникал номига эга ва турли хил маълумотлар қабул қилади. Устунларни баъзан майдонлар деб ҳам юритилади. Ҳар бир майдон турлича типга эга, масалан:

1. Оддий матн майдони. Белгилар сони 255 дан ошмаслиги керак.
2. MEMO-катта ўлчамли матн майдони. Белгилар сони 65535дан ошмаслиги шарт. Оддий матн ва MEMO майдонида ҳисоб ишларини бажариб бўлмайди.
3. Сонли майдон. Сонли маълумотларни киритишга хизмат қилади ва ҳисоб ишларини бажаришда фойдаланилади. Бу майдон 1,2,4,8 ва 16 байтли бўлиши мумкин.
4. Сана ва вақт майдони. Бу майдон сана ва вақтни бичимланган ҳолда сақлаб қуйиш имконини беради (01. 06. 01 20:29:59). 8байт ўлчамга эга.
5. «Пул бирлиги» номи билан аталувчи майдон. Бу майдондан ҳисоб китоб ишларини юритишда фойдаланилади.
6. Ҳисоблагич майдони. Бу майдон 4 байт узунликка ва автоматик равишда маълум сонга ошиб бориш хусусиятига эга. Ушбу майдондан ёзувларни номерлашда фойдаланиш қулайдир.
7. Мантикий амал натижасини сақловчи майдон. Бу майдон «рост» (true) ёки «ёлғон» (false) қийматни сақлайди. Майдон ўлчами 1 байт.
8. OLE-номи билан юритилувчи майдон. Бу майдон Excel жадвалини, Word хужжатини, расм, овоз ва бошқа шу каби маълумотларни иккилик санок системасида сақлайди. Майдон ўлчами 1 Гбайтгача.
9. Гиперссылка майдони. Бу майдон белги ва сонлардан иборат бўлиб, бирор файл ёки сайтга йўл кўрсатади.
10. Қийматлар рўйхатидан иборат бўлган майдон. Бу майдон бир қанча қийматлардан иборат бўлган рўйхатдан танланган аниқ бир қийматни сақлайди.

Жадваллар орасидаги муносабатлар ишончли ишлаши ва бир жадвалдаги ёзув орқали иккинчи жадвалдаги ёзувни топиш учун жадвалда алоҳида майдон-уникал майдон бўлишини таъминлаш керак.

Уникал майдон-бу қийматлари такрорланмайдиган майдондир.

Сатрлар

Сатр — мантикий боғланган *реквизитларга* мос келувчи майдонлар йиғиндисидир. Сатрнинг тузилиши ўз таркибига мос ҳар бир оддий маълумотга эга майдонлар таркиби ва кетма-кетлиги билан белгиланади.

Ҳар бир сатр алоҳида битта шахс ёки нарсга ҳақида бўлади. Ҳар бир сатрнинг формати ва атрибути бир хил. Сатрлар кортежлар ёки ёзув(қатор)лар ҳам деб аталади.

Қийматлар

Ҳар бир сатр устунга боғлиқ ҳолда бир қанча қийматларни ўз ичига олади. Ҳар бир қийматнинг типи устунга берилган типга мос бўлиши керак.

Талабалар

stud_id	surname	name	stipend	kurs	city
1	Islomov	Ikrom	150	1	Andijon
3	Fozilov	Fozil	200	3	Qashqadaryo
6 ...	Salimov	Vohid	150	4	Namangan

Атрибут

муносабат схемаси

Атрибут, майдон қиймати

Кортеж, ёзув, сатр

домен, устун

сатрли

сонли

13.1 - расм. Жадвал структураси

Калитлар

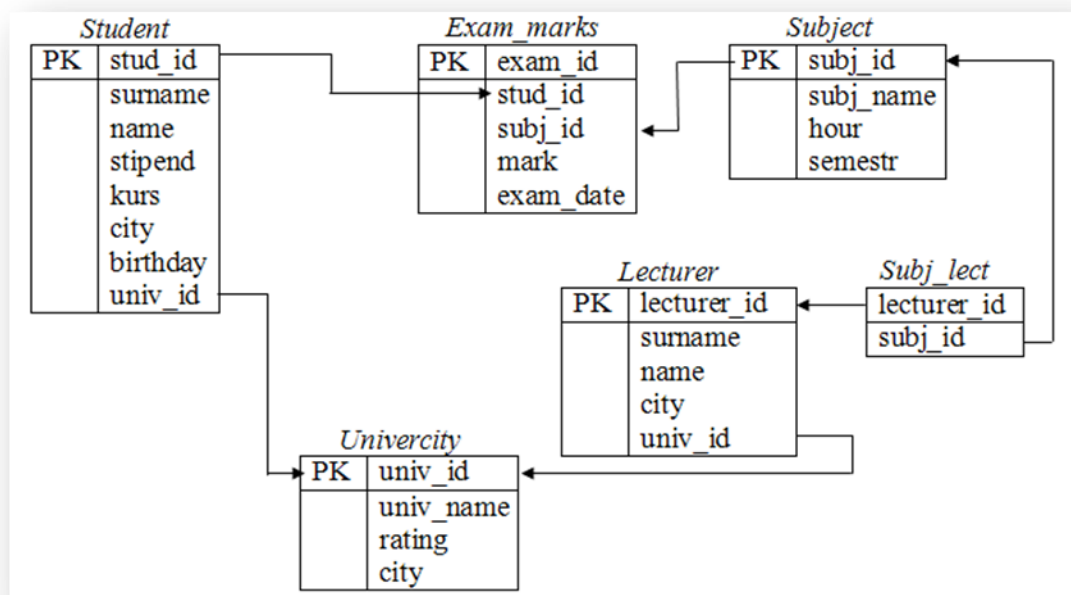
Баъзан жадваллардаги ҳар бир сатрни бир-биридан ажратиб олишга тўғри келади. Айнан шунинг учун калитдан фойдаланилади. Масалан, талабалар ҳақида маълумотлар киритилган бир қанча сатрлардан иборат *Талаба* жадвалидан ТДПУ да таҳсил оловчи Ахмедов Садриддин фамилия ва исмли талаба ҳақида маълумот олмоқчи бўлсак, бу жадвалда бошқа яна бир худди шундай исм ва фамилияли талаба ҳақида маълумот бўлиши мумкин. Уларни бир - биридан талаба кодини киритиш орқали фарқлай оламиз. Буни амалга ошириш учун яъни майдондаги қийматларни бир-биридан фарқлаб олиш учун шу майдонга(талаба коди майдонига) калит қўямиз. Реляцион маълумотлар базасида бундай калитни *бирламчи калит ёки бош калит*(Primary key) деб аталади. Бош калит қўйиладиган майдон(устун) иккита хоссага эга бўлиши керак:

1. Устундаги ёзувнинг ягоналиги. Калитдаги қиймат ўзгармаслиги лозим.
2. Кўп маъноликка йўл қўймаслик. Устундаги бирорта белгини ўзгартириш мумкин эмас, акс ҳолда бош калит ўзгаради.

Бундан ташқари бир нечта жадвалларни ҳам калит ёрдамида боғлаш мумкин. Масалан талаба ҳақидаги маълумотлар(шахсий, имтиҳон баҳолари, ўқув фанлари, ўқитувчилари, университети) ни жадвалга жойлаштириш керак бўлсин. Агар бу маълумотлар битта жадвалга жойлаштирилса, талабанинг имтиҳон баҳосини билиш учун ҳам барча маълумотларга мурожаат этишга тўғри келади. Бу эса ишлаш тезлигини пасайтиради. Шунинг учун талаба ҳақидаги маълумотлар категориялари бўйича алоҳида - алоҳида жадвалларда сақланади ва уларни талаба коди ёрдамида боғланади. Битта маълумотлар базасида бир нечта жадвал ташкил қилиш мумкин, калит улар орасидаги боғланишни яратиб беради. Реляцион жадваллар атамалари бўйича бундай муносабат *ташқи калит* деб номланади.

Масалан, бизда олти жадвал мавжуд: талабанинг шахсий маълумотлари жойлашган жадвал (Student), талабанинг имтиҳон баҳолари

жойлашган жадвал(Exam_marks), талабанинг ўқув фанлари жойлашган жадвал(Subject), ўқитувчилар маълумотлари жойлашган жадвал(Lecturer), университетларнинг маълумотлари жойлашган жадвал(University) ва ўқитувчилар ўқитадиган фанлари жойлашган жадвал(Subj_lect). Ушбу барча жадвалларда бирламчи калит мос равишда stud_id (Student жадвалида), exam_id (Exam_marks жадвалида), subj_id (Subject жадвалида), univ_id (University жадвалида), lecturer_id (Lecturer жадвалида)(13.2 - расм). Exam_marks жадвалида Student жадвалидаги stud_id майдонидаги қийматга тенг идентификатор ёзиладиган stud_id майдони мавжуд, ушбу майдон Exam_marks жадвали учун ташқи калит, Student жадвали учун эса бирламчи калит ҳисобланади. Худди шундай subj_id Exam_marks жадвали учун ташқи калит, Subject жадвали учун эса бирламчи калитдир. Subj_lect жадвалида эса бирламчи калит иштирок этмаган, бу жадвалда фақатгина ташқи калитлар мавжуд. Ташқи калитлар маълумотлар базасидаги жадваллар ўртасида алоқа ташкил этиш учун ишлатилади.



13.2 - расм. Жадваллар ўртасида алоқа ўрнатиш учун калитларни ишлатиш.

Индекслаш

Маълумотлар базаси билан ишлашда юзага келадиган асосий масалалардан бири бу – қидириш масаласи. Шунга кўра, агар маълумотлар базасида маълумотлар кўп бўлса, дастурчи олдида оддий қидириш масаласи эмас, балки самарали қидириш, яъни қидирув унча вақтни олмаслиги ва етарли даражада аниқ бўлиши керак. Бунинг учун (сўровларни ишлаб чиқишни оптималлаштириш учун) жадвалнинг баъзи майдонларини индекслаш керак бўлади. Битта устундаги сатрни кўрсатилган қиймат билан тез қидириш учун индексни ишлатиш фойдали. Индекс берилмаганда, жадвални ўқиш барча жадвалнинг биринчи ёзувидан бошлаб токи мос сатр

учрагунча амалга оширилади. Жадваллар қанча кўп бўлса, шунча кўшимча вақт керак бўлади. Агар жадвалнинг қаралаётган устунида индекс мавжуд бўлса, у ҳолда маълумотлар базаси қидириш учун барча маълумотларни кўриб чиқмасдан, маълумотлар файли оралиғи ўрнини тезда аниқлайди. Маълумотлар базаси қийматларни тезроқ қидиришни амалга ошириши учун, индексланган майдонларни хотирага яқин жойга жойлаштиради. 1000 та сатрдан таркиб топган жадваллар учун бу барча ёзувларни кетма-кет саралаш билан таққослаганда камида 100 маротаба тезроқ амалга ошади. Бироқ баъзи ҳолларда барча 1000 та сатр учун рухсат зарур бўлганда, кетма-кет ўқиш тезроқ амалга ошади, шунинг учун дискдан қидириш амали талаб қилинмайди. Шунга кўра баъзан индекслар фақатгина тўсиқ бўлади. Масалан, агар катта ҳажмдаги маълумотлар жадвалга кўчириляётган бўлса, у ҳолда ҳеч қандай индексга эга бўлмагани маъқул. Бироқ баъзи ҳолларда бир вақтда бир қанча индексни ишлатиш талаб этилади. (масалан, тез-тез ишлатиладиган жадвалларда сўровларни қайта ишлаш учун).

MySQL да индексни урта кўриниши мавжуд: *PRIMARY*, *UNIQUE* ва *INDEX*. Калит (KEY) сўзи эса индекс (*INDEX*) сўзига синоним сифатида ишлатилади. Барча индекслар хотирада *дарахтсимон* кўринишида сақланади.

PRIMARY – уникал индексли (калитли) майдонлар бўш қийматни қабул қилмайди (яъни NOT NULL ни). Жадвал фақатгина битта биринчи даражали индексга эга бўлади, лекин бир нечта майдонлардан иборат бўлиши мумкин.

UNIQUE – индекс (калит), берилган майдон фақатгина уникал қийматни қабул қила олади.

INDEX – оддий индекс.

MySQL да бундан ташқари сатрли майдонларни сатр бошланишидан бошлаб берилган сондаги белгиларни индекслаш мумкин.

Схемалар.

Маълумотлар базаси жадваллари лойиҳаларининг тўлиқ тўплами маълумотлар базасининг *схемаси* (*тархи*) дейилади. Схема қай бир маънода чизма (чертёж) га ўхшашдир. Схема жадвалларни устунлари билан бирга, устунларнинг маълумот турларини, ҳар бир жадвал учун бирламчи калитни ва барча ташқи калитларни кўрсатиб туриши керак. Схема ўз ичига, масалан, қайси нарса қай мақсадда кераклигини кўрсатувчи маълум бир намунавий маълумотларни олмайди. Схемалар биз фойдаланадиган “мавжудлик-муносабатлар” сингари диаграммалар кўринишида ёки матн кўринишида тасвирланиши мумкин. Масалан, ўқув маълумотлар базасининг схемасини тузамиз:

student(*stud_id*, surname, name, stipend, kurs, city, birthday, univ_id)

exam_marks(*exam_id*, stud_id, subj_id, mark, exam_date)

subject(*subj_id*, subj_name, hour, semestr)

lecturer(*lecturer_id*, surname, name, city, univ_id)

subj_lect(lecturer_id, subj_id)

university(*univ_id*, univ_name, rating, city)

Схемадаги белгилаб қўйилган элементлар, бошқа белгиланганлар билан муносабатларнинг бирламчи калити ҳисобланади. Чизиқ билан белгиланган

элементлар – бошқа жадвалдаги белгиланган муносабатлар учун ташқи калитлардир.

Муносабатлар

Реляцион маълумотлар базасида уч турли асосий муносабатлар мавжуд. Улар муносабатнинг ҳар қайси томонида қатнашувчи элементлар сонига қараб таснифланади. Демак, «**бирга-бир**», «**битта-кўпга**» ва «**кўп-кўпга**» муносабатлари:

«**Бирга-бир**» муносабати бўйича, муносабатда ҳар бир элементдан биттадан қатнашади. **Бирга-бир** муносабатининг маъноси шуки, битта жадвалнинг битта сатри бошқа жадвалдаги фақат битта сатрга тўғри келади (мос келади). Мисол учун, агар иккита жадвални кўрадиган бўлсак, улардан бирида талабалар тўғрисида маълумотлар, иккинчисида эса таълим муассасалари тўғрисидаги маълумотлар бўлса, у ҳолда бу жадваллар орасида **бирга-бир** муносабат мавжуд, чунки битта жадвалда маълумоти бўлган талаба учун таълим муассасалари тўғрисидаги маълумотлар бўлган иккинчи жадвалдан фақат битта сатр тўғри келиши мумкин.

Ҳаммадан кўп учрайдиган маълумотлар базасидаги муносабатлар тури бу **битта-кўпга** муносабати. «**Битта-кўпга**» муносабатида биринчи жадвалнинг битта сатри иккинчи жадвалнинг бир нечта сатрига боғланиши мумкин. Берилган муносабатлар турини кўрсатиш учун, ўқитувчилар ва улар дарс берадиган фанлар тўғрисида маълумотлар бўлган жадвалларга мурожаат этишимиз мумкин. Бошқа мисоллар сифатида корхона ва унда ишлайдиган ишчилар орасидаги муносабатлар кўрилиши мумкин. Ҳудди шундай муносабатлар компьютер ва унга кирувчи компонентлар орасида мавжуд ва х. к.

Агар «**кўп-кўпга**» муносабати мавжуд бўлса, демак, бир жадвалнинг бир нечта сатри иккинчи жадвалнинг бир нечта сатрига боғланган бўлади. Иккита жадваллар орасида **кўп-кўпга** муносабати ҳосил бўлади, қачонки: биринчи жадвалдаги битта сатр иккинчи жадвалдаги биттадан ортиқроқ сатр билан боғланиши мумкин бўлса; иккинчи жадвалдаги битта сатр биринчи жадвалдаги биттадан ортиқроқ сатр билан боғланиши мумкин бўлса.

Мисол учун талабалар порталининг ўқув маълумотлар базасига мурожаат этамиз. Икки гуруҳ объектларни кўриб чиқамиз: талабалар рўйхати ва фанлар бўйича топширилган имтиҳонлар рўйхати. Бу маълумотларни ўз ичига олган жадваллар орасида кўп-кўпга муносабати мавжуд, чунки ҳар қайси талаба биттадан ортиқ фандан имтиҳон топширган. Ҳудди шундай, ҳар қайси фандан биттадан талаба имтиҳон топширган бўлиши мумкин.

Жадвалларнинг турлари

Маълумотлар базаси қоида бўйича қуйидаги икки турли жадваллардан ташкил топади:

- Реал объектлар ҳақида маълумот сақловчи оддий жадваллар, улар бошқа содда объектлар билан «бирга-бир» ва «битта-кўпга» муносабатларига кириша оловчи калитларга эга бўлади.

- Иккита ҳақиқий объект орасидаги «кўп-кўпга» муносабатларини акс эттирувчи *боғловчи жадваллар*. Бундай жадваллар қисман баъзи бир ҳақиқий ўтказишлар ҳисобига ташкиллаштирилади.

Кўп миқдордаги бўш атрибутлар (катаклар) дан воз кечиш.

Маълумотларда кўп миқдордаги бўш қийматларнинг мавжудлиги – амалий жиҳатдан ёмон ҳолдир. Бу қаттиқ дискдаги мавжуд жойдан мақсадли фойдаланишда ва устундаги барча қийматлар йиғиндиси ва шу каби ифодаларни ҳисоблашда бир қатор муаммоларга олиб келади. Фойдаланувчи жадвалда бўш қийматни кўрар экан, у буни мос бўлмаган қийматми ёки маълумотлар базасидаги хатоликми, ёки бўлмаса, шунчаки маълумот киритилмаганми, қайси бири эканлигини билмайди. Маълумотлар базасини лойиҳалашда энг мақбул йўлни тутиб, бўш қийматлар билан боғлиқ кўпгина муаммолардан қутилиш керак.

13.2. SQL тили тарихи ва вазифалари

Барча реляцион турдаги маълумотлар базасига маълум талаблар асосида мурожаат қилиб, улардан сўров талабларига жавоб берадиган натижаларни олиш ва база билан ишлаш учун SQL (Structured Query Language — структуралаштирилган сўров тили) тили ишлаб чиқилган. У фойдаланувчиларга маълумотлар базаси билан ўзаро алоқа (маълумотларни кўриш, қидириш, кўшиш ва бошқариш) учун шароит яратади.

SQL (Structured Query Language) — Структуралашган сўровлар тили — реляцион МБ билан ишларда қўлланиладиган сўровлар тили.

Маълумки, реляцион моделнинг тарихи (ва билвосита SQL тарихи ҳам) 1970 йил Е.Ф.Коддни (бу пайтда у IBM корпорациясининг Сан Хоседаги тадқиқот марказида ишлаган) мақоласи чиққан даврдан бошланди. 1974 йил шу лабораторияда ишловчи Д. Чемберлен SEQUEL («Structured English Query Language») деб номланган тилни эълон қилди. 1976 йил бу тилнинг қайта ишланган SEQUEL/2 версияси яратилди ва у расмий равишда SQL деб аталган. Ҳозирги кунда SQL қисқартмасини баъзилар «сиквэл» деб талаффуз этади. Бироқ расмий равишда у «эс-кю-эл» деб ўқилиши керак.

SQL тили реляцион алгебра пайдо бўлгандан кейин яратилди ва унинг биринчи тимсоли(прототипи) IBM Research компанияси томонидан 70-йиллар охирида ишлаб чиқилган. Бу тил биринчи IBM System R номли МББТ таркибига киритилган. Кейинчалик бу тил кўпгина тижорат МББТ таркибида қўлланилган ва кенг тарқалганлиги сабабли вақт ўтиши билан реляцион МББТ ларда маълумотлар устида амаллар бажарувчи тилларнинг норасмий стандарти бўлиб қолди. SQL тилининг биринчи расмий стандарти 1989 йил қабул қилинган. Кўпгина МББТ лари ушбу стандартни қўллайди. Бироқ маълумотлар базаси билан боғлиқ ахборот технологияларининг ривожланиши ва баъзи талабларнинг пайдо бўлиши биринчи SQL стандартини қайта ишлаш ва кенгайтиришни тақозо этди.

SQL тили фойдаланувчи реляцион маълумотлар базаси билан мулоқат қилиши учун мўлжалланган бўлиб, қуйидаги 3 та қисмдан иборат:

- DDL (Data Definition Language) – маълумотларни аниқлаш тили. Маълумотлар базасини (жадвалларини, индексларини ва х.к.) яратиш ва унинг схемасини тахрирлаш учун мўлжалланган.

- DCL (Data Control Language) – маълумотларни бошқариш тили. Фойдаланувчиларнинг маълумотлар базаси объектларига мурожаатини чегаралаш операторларидан иборат.

- DML (Data Manipulation Language) – маълумотларни қайта ишлаш тили. Маълумотлар базаси жадвалларига ўзгартиришлар киритиш учун мўлжалланган.

Маълумотлар базаси билан ишловчи ихтиёрий тил фойдаланувчига қуйидаги имкониятларни яратиши лозим:

- структурасини тўла тавсифлаган ҳолда маълумотлар базасини ва жадвалларини яратиш;

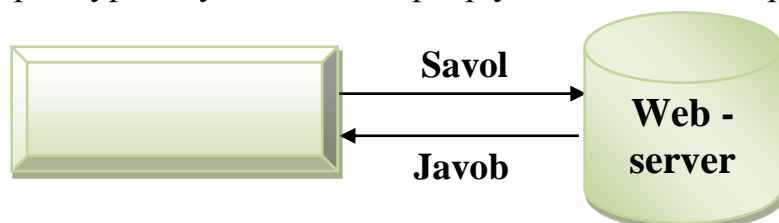
- маълумотлар устида манипуляция амалларини бажариш, масалан, жадвалларга маълумотларни киритиш, таҳрирлаш ва ўчириш;
- оддий ва мураккаб сўровларни бажариш.

Бундан ташқари, маълумотлар базаси билан ишловчи тил юқоридаги амалларни бажариш учун фойдаланувчилардан кам уринишларни талаб қилиши, ҳамда буйруқларининг синтаксиси ва тузилишини ўрганиш учун осон ва тушунарли бўлиши керак. Ниҳоят бу тил универсал бўлиши керак. Бу бир МББТ дан бошқасига ўтганда буйруқларни бир хил структураси ва синтаксисидан фойдаланишни таъминлайди. MySQL тили бу талабларни барчасини қаноатлантиради.

13.3. WEB –маълумотлар базасини ташкил этиш

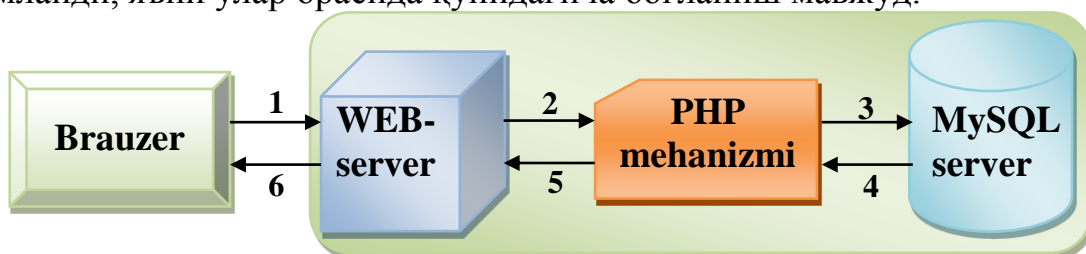
WEB –маълумотлар базасининг ички архитектураси ва ишлаш принципини кўриб ўтаемиз.

Бу тизим иккита объектдан иборат бўлиб, улар WEB –браузер ва WEB –сервер ҳисобланади. Уларни боғлаб турувчи канал мавжуд. WEB-браузер серверга сўров жўнатади ва сервер унга жавоб қайтаради.



13.3 - расм. WEB –сервер ва WEB –браузер орасидаги боғланиш.

WEB-маълумотлар базасини базали архитектураси ўзида WEB-браузер, WEB-сервер, ишловчи механизм ва маълумотлар базасини серверини жамлайди, яъни улар орасида қуйидагича боғланиш мавжуд.



13.4 - расм. WEB –маълумотлар базасининг базали архитектураси ўзида WEB –браузер, WEB –сервер, механизм саҳифалари ва маълумотлар базасини жамлайди.

1. WEB-саҳифа орқали аниқланган HTTP –сўровларни WEB-браузер орқали жўнатади.

2. WEB-сервер сўровни қабул қилиб, PHP механизмига қайта ишлаш учун жўнатади.

3. PHP механизми саҳифани синтактик анализ қилишни бошлайди.

4. MySQL сервери маълумотлар базасига келаётган сўровларни қабул қилади, уларни қайта ишлайди, ундан кейин натижаларни жўнатади. Ушбу ҳолда маълумотларни PHP механизмига узатади.

5. PHP механизми сўров натижаларини HTML кўринишига ўтказиб, ишни охирига етказади, шундан сўнг натижаларни HTML шаклида WEB серверга ўтказади.

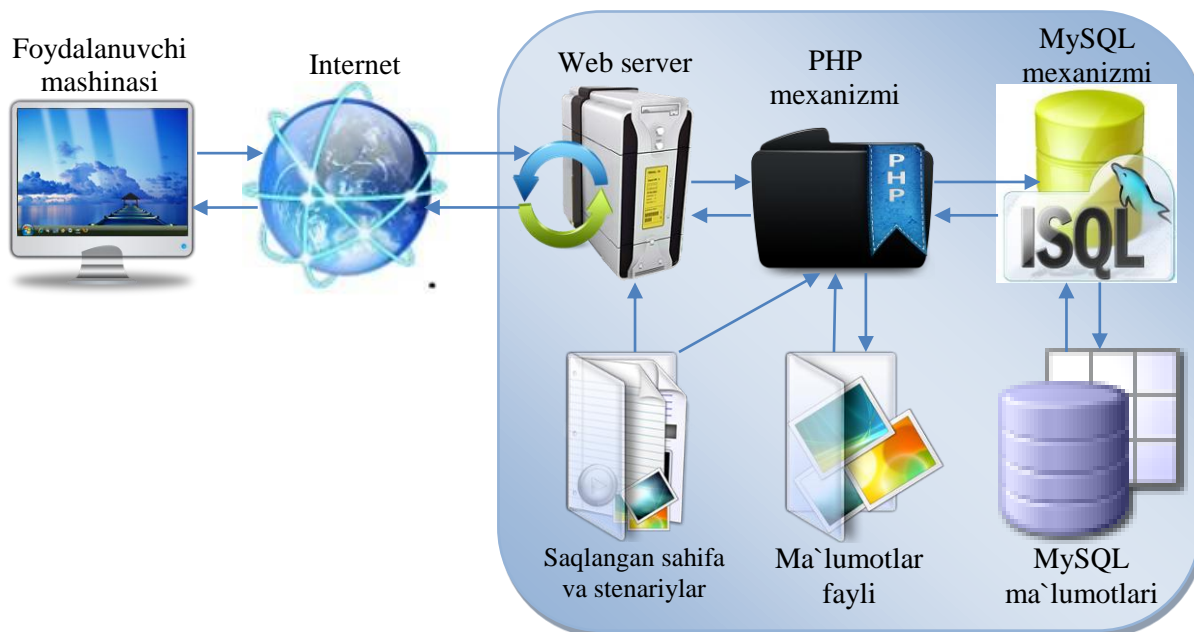
6. WEB сервер керакли маълумотларни фойдаланувчи кўра оладиган қилиб, HTML ни браузерга чиқаради.

WEB маълумотлар базасини бошқариш тизимида юзага келадиган ҳар бир мижоз сервер алоқаси тизимнинг ички структураси ва фойдаланувчи маълумотлари, амалларига боғлиқ ҳолда амалга оширилади.

Бундай тизим асосий учта қисмдан ташкил топади:

1. Фойдаланувчи машинаси
2. Internet
3. Қайта ишловчи тизим

Охирги кўрсаткичлар шундан далолат берадики, фойдаланувчиларининг он-лайн харидларини амалга ошириши, "Банк-мижоз" хизматидан фойдаланиши, электрон хужжат алмашиш учун дастурий воситалардан фойдаланишлари, соғлиқни сақлаш ва тиббиёт соҳаси ташкилотларининг электрон каталогининг яратилиш, хўжалик юритувчи субъектлар ва ташкилотларнинг Web-сайтларининг яратилиши ва янгиланиб бориши кониқарли даражада эмас. Бу эса, тегишлича, мазкур йўналишдаги ишлар такомиллаштирилишини тақозо этади.



Мижоз фойдаланувчи машинасидаги WEB саҳифа ёрдамида фойдаланувчи браузеридан WEB серверга Internet орқали сўров юборади. Агар саҳифада PHP сценарийлар мавжуд бўлса, WEB сервер бошқарувни саҳифани қайта ишлаш учун PHP механизмига узатади.

PHP сценарийлар дискдан маълумотларни ўқиш ва ёзиш, PHP ёки HTML файлларини include() ва require() операторларини ишлатиб юклаб олиш, ҳамда SQL - сўровларини MySQL демонида жўнатиб, ундан сўров жавобларини қабул қилиш амалларини бажаради. MySQL механизми дискдаги МББТ ида жойлашган маълумотларни ўқиш ва ёзиш учун жавоб беради.

13.4. MySQL маълумотлар базасини бошқариш тизими (МББТ)



13.5 – расм. *MySQL маълумотлар базаси кўриниши.*

MySQL – бу реляцион маълумотлар базасини бошқариш тизими. **MySQL** – очик кодли тизим. Бундан ташқари, тезкор, ишончли ва асосийси МББТ ишлатиш учун содда бўлган тизим. Бу интерфейс MySQL да маълумотлар базаси билан ишлашни анчагина соддалаштиради.

Матнли режимда маълумотлар базаси билан ишлаш худди буйруқлар сатрида буйруқларни киритиш каби бўлади (13.6 - расм), унинг натижаси эса алоҳида жадвал шаклида чиқазилади. (13.7 - расм).

13.1- мисол.

```
mysql> show databases;
```

13.6 - расм. *Буйруқлар сатрида MySQL билан ишлаш.*

`show databases` буйруғи— барча мавжуд маълумотлар базасини чиқазади

```
mysql> show databases;
+-----+
| Database |
+-----+
| book     |
| mysql    |
| test     |
+-----+
3 rows in set (0.01 sec)

mysql>
```

13.7 – расм. *show databases буйруғининг натижаси.*

MySQL ни ўрнатиш ва уни ишга тайёрлаш олдинги бўлимларда кўриб ўтилган. **MySQL** ни ўрнатгандан сўнг `mysql.exe`, `mysqld.exe` ва х. файллари жойлашган каталогга (Windows XP да `c:\mysql\bin`) кирилади. `mysqld.exe` файли **MySQL**-тизимини ишга туширади. **MySQL**-тизими бир қанча системаларда сервис сифатида ишга тушади. Тизим ишга тушгандан сўнг, `mysql.exe` дастури юкланади. Бу ерда хатто парол ҳам сўрамайди. Шундан сўнг, буйруқлар сатрида

13.2 – мисол.

```
shell> mysql.exe -u root
ёки
shell>mysql -u root mysql
```

буйруқларини ёзиб, `mysql` тизимининг барча администратор ҳуқуқларини олиш мумкин.

биринчидан, баъзи-бир камчиликларни тўғрилаймиз. (администратор паролнинг йўқлиги, бошқа фойдаланувчиларни киришига имкон яратади, шунинг учун унга парол қўямиз):

13.3 – мисол.

```
shell> mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('new_password')
WHERE user='root';
mysql> DELETE FROM user WHERE user='';
mysql> FLUSH PRIVILEGES;
```

MySQL фойдаланувчилари ҳақидаги барча маълумотлар махсус маълумотлар базаси *mysql* нинг *user* жадвалида сақланади, унга кириш ҳуқуқига фақат тизим администратори эга. Шунинг учун, бирор-бир паролни ўзгартириш учун ушбу жадвални ўзгартирилади. Парол сифатида киритилган маълумотлар шифрланиши учун *PASSWORD* функцияси ишлатилади. Администратор паролни ўзгартиришдан ташқари, барча фойдаланувчи логинларини (*DELETE* буйруғи) ўчириш керак. *Flush Privileges* буйруғи маълумотлар базасини бошқариш тизимини(*mysql*) янгилаш учун имкон яратади.

Энди лойиҳада қўрилган талабалар порталининг ўқув маълумотлар базасини яратамиз. (биз ҳали ҳам тизим администратори сифатида ишляпмиз):

13.4 – мисол.

```
mysql> create database talaba;
```

Unix тизимида маълумотлар базасининг номи регистрга сезгир, шунинг учун бу тизимда маълумотлар базасига фақатгина *talaba* номи билан мурожаат этиш мумкин, *Talaba*, *TALABA* ёки яна бошқа номлар билан эмас. Бу жадваллар номи учун ҳам тегишли. *Windows* тизимида эса бу чегара қўлланилмайди. Унда *talaba~Talaba~TALABA* номлар тенг кучли.

Маълумотлар базасини яратиш, ундан фойдаланиш учун рухсат бермайди. *talaba* маълумотлар базасидан фойдаланиш учун актив ҳолатга ўтказиш керак, бунинг учун ушбу буйруқни берамиз:

13.5 – мисол.

```
mysql> USE talaba;
Database changed
```

mysql сеанси бошлангандан сўнг, маълумотлар базаси фақатгина бир маротаба яратилади, лекин ушбу базадан бир неча маротаба фойдаланиш мумкин.

MySQL нинг барча буйруқлар сатри нуқта вергул билан тугатилади. Агар нуқта вергул ишлатилмаган бўлса, у қўйилмагунча берилган буйруқ сатри буйруқлари бажарилмайди.

13.6 – мисол.

```
mysql> show tables
->
->;
```

Энди оддий фойдаланувчи яратамиз ва унга яратилган маълумотлар базасида ишлаш учун ваколат берамиз. Бунинг учун *GRANT* буйруғидан фойдаланамиз.

GRANT – фойдаланувчига маълумотлар базасида ишлаш учун ваколатларни (имтиёзлар, привилегия) беради.

MySQL да 4 та даражадаги ваколатлар мавжуд:

1. Глобал
2. Маълумотлар базаси
3. Жадваллар
4. Устунлар

Синтаксиси:

GRANT (*ALL PRIVILEGES* | (*SELECT, INSERT, UPDATE, DELETE, REFERENCES (column list), USAGE*)) *ON ITEM TO user_name*[*identified by 'password'*] [*WITH GRANT OPTION*]

PRIVILEGES (ваколатлар) бир-биридан вергул орқали ажратиб ёзилади.

column list (устунлар рўйхати) ваколатлар берилаётган устунни аниқлаш учун ишлатилади. Устунлар сони биттадан кўп бўлса, вергул орқали ажратилади.

ITEM(элемент) ўрнида янги ваколат берилаётган маълумотлар базаси ёки жадвал берилади. Агар *ITEM* ўрнига (*.*) ишлатилса, ваколатлар барча маълумотлар базаси учун ўрнатилади. Бундай амал, яъни ваколатларни барча маълумотлар базаси учун ўрнатиш *глобал ваколатлар* деб аталади. (*.*) ўрнида (*) амалини ҳам ишлатиш мумкин.

Агар *ITEM dbname.*(маълумотлар_базаси_номи.*)* каби берилса, яъни ваколатлар берилган маълумотлар базасининг барча жадваллари учун ўрнатилса, бу *маълумотлар базаси ваколатлари* дейилади.

Агар *ITEM dbname.table_name (маълумотлар_базаси_номи.жадвал_номи)* каби берилса, демак ваколатлар кўрсатилган жадвал учун берилади ва бундай ваколатлар *жадваллар ваколати* дейилади.

Агар *ITEM dbname.table_name* ва *column list* да берилган зарур устунлар рўйхати берилса, демак ваколатлар кўрсатилган устунлар учун берилади ва бундай ваколатлар *устунлар ваколати* дейилади.

user_name ўрнида фойдаланувчининг MySQL га кириш учун керак бўлган номи берилади. Фақатгина бу ном тизимда мавжуд ном билан бир хил бўлмаслиги лозим. Фойдаланувчиларни ажратишга қулай бўлишлиги учун унинг номига хост номи қўшиб ёзилса ҳам бўлади, масалан *lola* (*lola@localhost* каби) ва *lola@somewhere.com*. Кўпгина фойдаланувчилар турли доменларда бир хил номга эга бўлишади. Фойдаланувчи номига хост номини қўшиб берилиши айнан шу ҳолатларда қўл келади.

password параметрида тизимга кириш учун парол берилади.

Агар *with grant options* опцияси кўрсатилган бўлса, бу фойдаланувчига ўз ваколатларини бошқа фойдаланувчига узатиш имконини беради.

Ваколатлар *mysql* номли маълумотлар базасининг тўртта тизимли жадвалларида сақланади. Бу жадваллар: *mysql.user*, *mysql.db*, *mysql.tables_priv*, *mysql.columns_priv* каби номланади ва улар юқорида кўрилган тўртта даражадаги ваколатларга тегишли.

```
mysql> grant all
```



```
on *
to lola@localhost
identified by '135'
-> width grant options;
```

lola фойдаланувчиси 135 пароли билан барча маълумотлар базасига ваколат ва шу ваколатларни узатиш ҳуқуқини олади.

Mysql да учта асосий ваколатлар тури мавжуд. Улар:

1. Оддий фойдаланувчига берилиши мумкин бўлган ваколатлар.
2. Фақат администраторларга тегишли бўлган ваколатлар.
3. Махсус ваколатлар тўплами.

1. Оддий фойдаланувчига берилиши мумкин бўлган ваколатлар:

select (жадвал ва устунлар учун) - сатрларни танлашда ёрдам беради;

insert (жадвал ва устун учун) - жадвалга янги сатрлар(ёзувлар) қўшиш учун ишлатилади;

update (жадвал ва устун учун) - сатрлардаги маълумотларни ўзгартириш учун ишлатилади;

delete (жадвал учун) - сатрларни ўчиради;

index (жадвал учун) - индекслар яратиш ва ўчириш учун ишлатилади;

alter (жадвал учун) - жадвалга ўзгартириш киритиш учун ишлатилади;

create (маълумотлар базаси ва жадваллар учун) - янги маълумотлар базаси ёки жадваллар яратиш учун ишлатилади;

drop (маълумотлар ва жадвал учун) - маълумотлар ёки жадвалларни ўчиришда ишлатилади.

2. Администратор учун ваколатлар:

Reload – ваколатлар жадвалларини ишга туширишда ва ваколат журналли файллар, жадвалларга қўшимча киритишда фойдаланилади;

Shutdown – Server MySQL ни тўхтатиш учун ишлатилади;

PROCESS – тизимдаги жараёнларни бошқариш ва уларни ўчириш учун фойдаланилади;

File – жадваллардаги маълумотларни файлга ва аксинча файлдаги маълумотларни жадвалга жойлаштиради.

3. Махсус ваколатлар тўплами:

ALL - юқоридаги барча ваколатларни(оддий фойдаланувчи ва администраторга берилган ваколатларни) ўз ичига олади. ALLPRIVILEGES ва ALL га тенг кучли.

USAGE – фойдаланувчига системага кириши учун имкон беради. Лекин рухсатсиз ҳеч қандай амал бажариб бўлмайди.

Энди янги фойдаланувчи яратамиз.

13.7 – мисол.

```
mysql> grant all privileges
on talaba.*
to ilyosbey@localhost
identified by '123';
```

Ilyosbey фойдаланувчиси 123 пароли билан talaba маълумотлар базасида ишлаш учун рухсат ҳуқуқини олади.

REVOKE – grant га қарама қарши амалларни бажаради. У фойдаланувчидан ваколатларни тортиб олади.

Синтаксиси:***REVOKE** privileges [(column)] on item from user_name;**ёки**Mysql > **REVOKE** grant options on item from user_name;*

фойдаланувчидан бошқа фойдаланувчига бериладиган ваколатларни тортиб олади.

13.8 – мисол.

```
Mysql > REVOKE all on * from ilyosbey;
```

ilyosbey фойдаланувчисидан барча ваколатларни тортиб олади.

13.9 - мисол.

```
Mysql > REVOKE alter, create, drop on talaba, * from ilyosbey;
```

ilyosbey фойдаланувчисига ваколатларни чеклаб қўяди, лекин тортиб олмайди.

Энди lola номли фойдаланувчи сифатида мос парол билан тизимга киравиз:

13.10- мисол.

```
shell>mysql -u lola -p
Enter password: ***
Welcome to the MySQL monitor!...
mysql>
```

13.5. MySQL МББТ нинг асосий операторлари

Маълумотлар базаси билан қуйидаги амалларни бажариш мумкин:

1. Маълумотлар базасини(МБ) яратиш, ўчириш ва ўзгартириш;
2. Фойдаланувчиларга ваколат бериш, ўчириш, ўзгартириш, қўшиш;
3. Маълумотлар базаси(жадвал ва ёзувлар)га маълумотларни киритиш, ўчириш ва ўзгартириш;
4. МБ маълумотларини саралаш.

Биринчи иккита амални бажариш ҳуқуқи фақатгина МББТ администраторлари ва махсус ҳуқуқи бор фойдаланувчиларга берилган.

Маълумотлар билан ишлашда маълумотлар базасини яратиш, лозим бўлганда бу маълумотлар сақланадиган жадваллар структурасини ўзгартириш ва ўчириш керак бўлади. Бунинг учун *SQL* нинг қуйидаги операторлари мавжуд *CREATE TABLE*, *ALTER TABLE* ва *DROP TABLE*.

Маълумотлар базасини яратиш: *CREATE TABLE* оператори

CREATE TABLE оператори жорий маълумотлар базасида берилган ном билан жадвал яратади. Агар маълумотлар базаси активлашмаган бўлса ёки берилган жадвал номи мавжуд бўлса, буйруқни бажаришда хатолик ҳақида огоҳлантиради.

Синтаксиси:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS]

жадвал_номи [(устунларни_аниқлаш,...)]

[жадвал_опцияси] [select_ифода]

MySQL 3.22 ва ундан кейинги версияларида жадвал номини *malumotlar_bazasi_nomi.jadval_nomi* каби кўрсатади. Ёзувнинг бу кўриниши кўрсатилган маълумотлар базаси жорий ёки жориймаслигига боғлиқ эмас.

MySQL 3.23 версиясида вақтинчалик жадвал яратиш учун *TEMPORARY* калит сўзи ишлатилади. Вақтинчалик жадвал боғланиш якунланганда автоматик тарзда ўчирилади, унинг номи эса берилган боғланиш давомида ҳақиқий ҳисобланади. Бу шуни англатадики, иккита турли боғланишда мавжуд жадвал худди шундай номли жадвал билан ёки бир хил номли вақтинчалик жадваллар бир-бири билан конфликтсиз ишлатилиши мумкин (ёки мавжуд жадвал вақтинчалик жадвал ўчирилгунча яширин бўлиб туради). *MySQL* 4.0.2 версиясида вақтинчалик жадвалларни яратиш учун *CREATE TEMPORARY TABLES* ваколатига эга бўлиш зарур.

MySQL 3.23 ва ундан кейинги версияларида кўрсатилган жадвал мавжуд бўлса, хато бермаслиги учун *IF NOT EXISTS* калит сўзи ишлатилади. Бу ҳолда жадвал структураларининг бир хиллиги текширилмайди.

Ҳар бир жадвал маълумотлар базаси каталогида аниқланган файллар тўплами сингари тасвирланади.

устунларни_аниқлаш параметри жадвалда қандай устунлар бўлиши кераклигини белгилайди. Жадвалнинг ҳар бир устуни бўш бўлиши (*NULL*), одатга кўра қийматга эга бўлиши, калитли ёки автоинкремент(*AUTO_INCREMENT*) бўлиши мумкин. Бундан ташқари, ҳар бир устун учун унда сақланадиган маълумотлар типини кўрсатиш керак.

Агар NULL ҳам, NOT NULL ҳам кўрсатилмаса, у ҳолда устун NULL кўрсатилгандек интерпретация қилинади. Агар майдон автоинкремент сифатида белгиланса, у ҳолда жадвалга маълумот киритилганда ва бу майдонга бўш қиймат (NULL, ёки ҳеч нарса ёзилмаганда) ёки 0 киритилганда унинг қиймати ҳар гал 1 та қийматга оширилади. Жадвалда автоинкремент майдон фақатгина битта бўлиши мумкин, шунинг учун уни индекслаш шарт эмас. Автоинкремент кетма-кетлиги 1 дан бошланади.

Қоидага кўра устунлар қуйидагича берилади (*устунларни_аниқлаш*):

устун_номи тип [NOT NULL / NULL]

[DEFAULT одатга кўра қиймати]

[AUTO_INCREMENT][PRIMARY KEY]

[reference_definition]

Устун *тип* (*устунларни_аниқлаш* параметрининг *устун_номи тип*) қуйидагилардан бирортасини қабул қилиши мумкин:

- **бутун:**

Тип	Оралиғи	Хотира (байт)	Тавсифи
TINYINT[(M)]	-127...128 ёки 0...255	1	Энг кичик бутун сон
SMALLINT[(M)]	-32768...32767 ёки 0...65535	2	Кичик бутун сон
MEDIUMINT[(M)]	-8388608...8388607 ёки 0...16777215	3	Ўртача катталиқдаги бутун сон
INT[(M)]	$-2^{31} \dots 2^{31}-1$ ёки $0 \dots 2^{32}-1$	4	Оддий бутун сон
INTEGER[(M)]			INT нинг синоними
BIGINT[(M)]	$-2^{63} \dots 2^{63}-1$ ёки $0 \dots 2^{64}-1$	8	Катта бутун сон

- **хақиқий:** REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]

Тип	Оралиғи	Хотира (байт)	Тавсифи
FLOAT [(precision)]	precision(аниқлик) га боғлиқ ҳолда	турли	Бир ва икки карра аниқликдаги каср сон
FLOAT [(M,D)]	$\pm 1.175494351\text{E}-38$ $\pm 3.402823466\text{E}+38$	4	Бир аниқликдаги каср сон. FLOAT(4) га эквивалент.
DOUBLE [(M, D)]	± 1.79769313486231 $57\text{E}+308$ ± 2.22507385850720 $14\text{E}-308$	8	Икки карра аниқликдаги каср сон. FLOAT(8) га эквивалент.
DOUBLE PRECISION [(M, D)]			DOUBLE [(M, D)] га ўхшаш

REAL [(M, D)]			DOUBLE [(M, D)] га ўхшаш
DECIMAL[(M[,D])]	турли	M+2	Char сифатида сақланган каср сон. Оралиғи M га боғлиқ.
NUMERIC[(M,D)]			DECIMAL га ўхшаш

- **белгили:** CHAR(length) [BINARY] и VARCHAR(length) [BINARY]

Тип	Оралиғи	Тавсифи
CHAR(M)	1 дан 255 гача белги	Сатрнинг (устуннинг)ўзгармас катталиги (узуслиги) ($1 \leq M \leq 255$). Бу узуслик 1 дан 255 гача бўлган ихтиёрий қийматни қабул қилиши мумкин. char типдаги катталикларни сақлаганда бўш қолган жойларни ўнг томондан пробеллар билан тўлатади ва аксинча сақланаётган катталикларни чиқариш пайтида тўлатилган пробеллар ўчириб ташланади. Char устундаги берилган катталик устуннинг максимал берилган узуслигидан катта бўлса у ҳолда катталик мос равишда қирқилади.
VARCHAR(M)	1 дан 255 гача белги	Сатрнинг (устуннинг)ўзгармас катталиги (узуслиги) ($1 \leq M \leq 255$). VARCHAR устундаги катталиклар эса ўзгарувчиларнинг узуслигидан иборат. Бу ҳам Char турига ўхшаб 1..255 гача бўлган ихтиёрий қийматни қабул қилиши мумкин. VARCHAR туридаги катталикларни сақлаш пайтида фақат керакли бўлган белгилар сонини сақлайди ва 1 байт ёзувнинг узуслиги учун жой ажратади. Сақланаётган катталикларнинг бўш жойлари пробел билан тўлатилмайди ва аксинча ўчириб ташланади. VARCHAR устундаги берилган катталик устуннинг максимал берилган узуслигидан катта бўлса, у ҳолда катталик мос равишда қирқилади.

- **катта объектлар билан ишлаш учун:** BLOB

TINYBLOB	$2^8-1(255)$	Кичик объект майдони.
BLOB	$2^{16}-1(65535)$	Объект майдони. Бу маълумотлар типи ўзида иккиламчи катта ўлчовдаги объектни,

		шунингдек ўзгарувчан маълумотлар миқдорини ўзида мужассамлаштиради. Бу типнинг 4 та модификацияси мавжуд. Улар фақат сақланадиган катталикларнинг максимал узунликлари билан фарқланади.
MEDIUMBLOB	$2^{24}-1$ (16777216)	Ўртача катталиқдаги объект майдони.
LOB	$2^{32}-1$ (4294967295)	Катта объект майдони.

- **матнли: TEXT**

TINYTEXT	2^8-1 (255)	Кичик матн майдони.
TEXT	$2^{16}-1$ (65535)	Маълумотлар тури ҳам BLOB каби 4 та модификация турига эга. TEXT ва BLOB турларидаги асосий фарқ шундаки BLOB катталиклари учун маълумотларни саралаш ва таққослаш регистрга боғлиқ. TEXT катталиклари учун регистрнинг аҳамияти йўқ. Бошқача қилиб айтганда TEXT бу регистрдан мустақил бўлган блок. Агар BLOB ёки TEXT устундаги берилган қиймат устуннинг максимал берилган узунлигидан катта бўлса у ҳолда қиймат мос равишда қирқилади.
MEDIUMTEXT	$2^{24}-1$ (16777215)	Ўртача катталиқдаги матн майдони.
LONGTEXT	$2^{32}-1$ (4294967295)	Катта матн майдони.

- **сана ва вақт: DATE и TIME**

Тип	Оралиғи	Тавсифи
DATE	1000-01-01 дан 9999-12-31	Маълумотни фақатгина сана сифатида сақлайди. "YYYY-MM-DD" (йил – ой – кун)форматда ишлатилади. MySQL да DATE типи қуйидагича кўринишларда бўлади. YYYY-MM-DD YY-MM-DD YYMMDD YYMM DATE типи узунлиги 4 байт.
TIME	-838:59:59	Фақат вақт ҳақидаги маълумотни

	838:59:59	<p>сақлайди. "HH:MM:SS" форматида ишлатилади. Сатр ёки сон сифатида ишлатилиши мумкин. MySQL нинг TIME типи қуйидаги синтаксисларни қабул қилади.</p> <p>HH:MM:DD HHMMDD HHMM HH</p> <p>TIME типи маълумотлари узунлиги 3 байт.</p>
DATETIME	<p>сатрли контекстларни ишлатишда: '1000-01-01 00:00:00' – '9999-12-31 23:59:59'</p> <p>сонли контекстларни ишлатишда '0000000000000000' - '99991231235959'.</p>	<p>Бу DATE ва TIME типларининг бирлаштирилгани. DATETIME типи қуйидаги тарафлари билан TIMESTAMP типига ўхшаш:</p> <p>Таркибида DATETIME типидagi устун мавжуд бўлган жадвалга маълумот қўшилаётган вақтда, DATETIME устуни ўзгармайди. DATETIME типининг узунлиги 8 байт.</p>
TIMESTAMP(M)	1970-01-01 00:00:00	<p>киритиш/янгилашлар автоматик ўзгартирилади. YYMMDDHHMMSS ёки YYYYMMDDHHMMSS форматларини қабул қилади. INSERT ни бажаришда TIMESTAMP майдонларини ўзгартириш мумкин. бу ёзувларни қўшишда эркин сана/вақт ни ўрнатиш учун фойдали. INSERT ёки UPDATE амалларини бажаришда жорий вақт ва саналарни автоматик ёзиб олиш учун берилган маълумотлар таъминлаб беради.</p> <p>TIMESTAMP(M) TIMESTAMP(14) YYYY-MM-DD HH:MM:SS TIMESTAMP(12) YY-MM-DD HH:MM:SS TIMESTAMP(10) YY-MM-DD HH:MM TIMESTAMP(8) YYYY-MM-DD TIMESTAMP(6) YY-MM-DD</p>
YEAR[(2 4)]	70-69 (1970-2069)	<p>Йил. Икки ёки тўртта рақамли кўринишда тасвирлаш мумкин.</p>

	1901-2155	
--	-----------	--

Майдон(устун) *узунлиги* киритилаётган қийматда жами нечта рақам(белги) қатнашиши кераклигини аниқлайди. дес қатнашган майдонларда эса бу ўнлик касрларда нуқтадан сўнг, яъни каср қисми нечта рақамдан иборат бўлиши кераклигини билдиради. Бу сонлар фақатгина устуннинг максимал узунлигини ҳисоблаш учун ишлатилади.

*устунларни_аниқлаш*да устунлар сони ва типи билан бирга чегаралаш ва текшириш учун калитли ва индексли устунлар рўйхатини ҳам бериш мумкин:

PRIMARY KEY (*индексланган_устун_номи, ...*)

ёки

KEY [*индекс_номи*] (*индексланган_устун_номи,...*)

ёки

INDEX [*индекс_номи*] (*индексланган_устун_номи,...*)

ёки

UNIQUE [INDEX] [*индекс_номи*] (*индексланган_устун_номи,...*)

ёки

FULLTEXT [INDEX] [*индекс_номи*] (*индексланган_устун_номи,...*)

ёки

[*CONSTRAINT symbol*]

FOREIGN KEY [*индекс_номи*] (*индексланган_устун_номи,...*)

[*reference_definition*]

ёки

CHECK (*expr*)

FOREIGN KEY, *CHECK* ва *REFERENCES* лар айна вақтда *MySQL*да ҳеч қандай вазифани бажармайди. Улар бошқа *SQL*-серверлар билан ҳамкорлик қилиш учун қўшилган. Шунинг учун уларга тўхталмаймиз.

Кўп ҳолларда устунга киритилган қийматлар бир-биридан фарқ қилиши мумкин. Агар устун учун *UNIQUE* чекланиш ўрнатилса, бу устунга мавжуд қийматни киритишга уриниш рад этилади. Бу чекланиш бўш эмас (*NOT NULL*) деб эълон қилинган устунларга қўлланиши мумкин.

Масалан,

```
CREATE TABLE Student
  (id INT NOT NULL UNIQUE,
   surname VARCHAR(100),
   name VARCHAR(50),
   stipend INT,
   kurs INT,
   city VARCHAR(100),
   birthday DATA,
   univ_id INT,
  );
```

Уникаллиги талаб қилинадиган устунлардаги калитлар (бирламчи калитлардан ташқари) кандидат калитлар ёки калитлар дейилади.

Жадвал чекланиши *UNIQUE* устунлар гуруҳига ўрнатилиши мумкин. Бу бир неча устунлар қийматлари комбинацияси уникаллигини таъминлайди.

SQL бирламчи калитларни тўғридан-тўғри бирламчи калит (PRIMARY KEY) чекланиши орқали таърифлайди. PRIMARY KEY жадвални ёки устунларни чеклаши мумкин. Бу чекланиш UNIQUE чекланиши каби ишлайди, жадвал учун фақат битта бирламчи калит (ихтиёрий сондаги устунлар учун) аниқланиши мумкин. Бирламчи калитлар NULL қийматига эга бўлиши мумкин эмас.

Масалан:

```
CREATE TABLE Student
(id INT NOT NULL PRIMARY KEY,
surname VARCHAR(100),
name VARCHAR(50),
stipend INT,
kurs INT,
city VARCHAR(100),
birthday DATA,
univ_id INT,
);
```

Жадвал яратишда, уни баъзи хусусиятлари(*жадвал_опцияси*)ни ҳам кўрсатиш керак, масалан:

- жадвал типи: *TYPE* = {*BDB* / *HEAP* / *ISAM* / *InnoDB* / *MERGE* / *MRG_MYISAM* / *MYISAM* }

- автоинкремент ҳисоблагичининг бошланғич қиймати: *AUTO_INCREMENT* = *сон*

- жадвал сатрининг ўртача узунлиги: *AVG_ROW_LENGTH* = *сон*

- жадвал учун изоҳ (60 белгидан иборат сатр): *COMMENT* = «*сатр*»

- сатрнинг тахминий энг катта ва энг кичик қиймати: *MAX_ROWS* = *сон* ва *MIN_ROWS* = *сон*

NOT NULL жадвал устундаги барча ячейкалар қийматга эга бўлиши зарурлигини билдиради, агар NOT NULL кўрсатилмаса, устун қийматлари бўш (NULL) бўлиши мумкин.

AUTO_INCREMENT —сонли устунлар учун ишлатиладиган MySQL нинг махсус имконияти. Агар жадвалга сатрларни қўйишда ушбу майдонни бўш қолдирилса, MySQL автоматик тарзда идентификаторнинг уникал қийматини қўяди. Бу қиймат майдондаги мавжуд максимал қийматдан биттага катта бўлади. Ҳарбир жадвалда бундай майдон фақатгина битта бўлади. AUTO_INCREMENT устунлари индексланган бўлиши зарур.

PRIMARY KEY калит сўзи устун номидан сўнг аниқланиб, ушбу устун жадвал учун бирламчи калит эканлигини англатади. Ушбу устундаги маълумотлар уникал бўлиши лозим. MySQL ушбу устунни автоматик тарзда индекслайди.

UNSIGNED чегаралаши қўйилган устун қиймати ёки бутун сон ёки нол бўлиши мумкин.

13.11 – мисол. 13.2 – расмда структураси берилган Талабалар порталининг ўқув жараёни жадвалларини яратамиз.

```
mysql>CREATE TABLE Student
(stud_id INT UNSIGNED NOT NULL PRIMARY KEY,
surname VARCHAR(100),
name VARCHAR(100),
stipend INT,
```

```

kurs INT,
city VARCHAR(200),
birthday DATE,
univ_id INT UNSIGNED NOT NULL          );
mysql>CREATE TABLE exam_marks
(exam_id INT UNSIGNED NOT NULL PRIMARY KEY,
stud_id INT UNSIGNED NOT NULL,
subj_id INT UNSIGNED NOT NULL,
mark INT,
exam_date DATE          );
mysql>CREATE TABLE subject
(subj_id INT UNSIGNED NOT NULL PRIMARY KEY,
subj_name VARCHAR(100) NOT NULL,
hour INT,
semestr INT          );
mysql>CREATE TABLE lecturer
(lecturer_id INT UNSIGNED NOT NULL PRIMARY KEY,
surname VARCHAR(100) NOT NULL,
name VARCHAR(100) NOT NULL,
city VARCHAR(200),
univ_id INT UNSIGNED NOT NULL          );
mysql>CREATE TABLE subj_lect
(lecturer_id INT UNSIGNED NOT NULL,
subj_id INT UNSIGNED NOT NULL );
mysql>CREATE TABLE univercity
(univ_id INT UNSIGNED NOT NULL PRIMARY KEY,
univ_name VARCHAR(100) NOT NULL,
rating INT,
city VARCHAR(200));

```

Жадвал яратишда CREATE буйруғининг яна бир элементи *SELECT* ифодаси (*select_ифодаси*) ҳам қатнашиши мумкин.

[IGNORE / REPLACE] SELECT ...

(*SELECT* нинг ихтиёрий тўғри ифодаси)

Агар CREATE буйруғи ёрдамида жадвал яратишда *select_ифодаси* кўрсатилса, у ҳолда танлашда олинган барча майдонлар яратилган жадвалга кўшилади.

MySQL нинг махсус SHOW буйруғи ёрдамида *маълумотлар базаси* таркиби (*маълумотлар базаси* жадваллари ва жадвал майдонлари) ни кўриш мумкин.

13.12-мисол. Барча *маълумотлар базаларини кўрсатиш*:

```
mysql>SHOW databases;
```

13.13-мисол. talaba *маълумотлар базасини жорий этиш*(активлаштириш):

```
mysql>use talaba;
```

13.14-мисол. Жорий (актив) базадаги барча *жадвалларини кўрсатиш*:

```
mysql>show tables;
```

MySQL маълумотлар базасидаги жадваллар рўйхатини чиқазади:

Tables in talaba

student
exam_marks
subject
lecturer
subj_lect

```
univercity
```

6 rows in set (0.07 sec)

DESCRIBE буйруғи жадвал ҳақида қўшимча маълумот олиш имкониятини беради.

13.15-мисол. univercity жадвалидаги майдонлар ҳақида маълумот олиш.

```
mysql> describe univercity;
```

MySQL фойдаланувчига жадвал ҳақидаги маълумотни беради:

Field	Type	Null	Key	Default	Extra
univ_id	int		PRI		
univ_name	varchar(100)	YES		NOT NULL	
rating	int	YES		NULL	
city	varchar(100)	YES		NULL	

4 rows in set (0.05 sec)

13.16 – мисол. student жадвалидаги барча устунларни кўрсатиш:

```
mysql> show columns from student;
```

Маълумотлар базасига маълумотларни киритиш: INSERT оператори

INSERT оператори мавжуд жадвалга янги сатр киритади. Бу операторнинг бир нечта формаси мавжуд бўлиб, *жадвал_номи* параметри ушбу барча формаларда сатрлар киритилиши лозим бўлган жадвал номини билдиради. Қиймат киритиладиган устунлар устун номлари(*устун_номи*) рўйхатида ёки SET қисмида кўрсатилади.

Синтаксиси(1-форма):

INSERT [*LOW_PRIORITY* | *DELAYED*] [*IGNORE*]

[*INTO*] *жадвал_номи* [(*устун_номи*,...)]

VALUES (*қиймат*,...),(...),...

INSERT буйруғининг бу формаси сатрни буйруқда кўрсатилган қийматга мос равишда киритади. Жадвал номидан сўнг берилган қавс ичида устунлар, *VALUES* калит сўзидан сўнг эса уларнинг қийматлари киритилади.

13.17 - мисол. Student жадвалига фамилияси(*surname*) ва туғилган санаси(*birthday*) қийматлари мос равишда «Akromov» ва «1992» га тенг бўлган сатрларни киритинг.

```
mysql> INSERT INTO Student
(surname, birthday) VALUES
('Akromov', '1992');
```

Синтаксиси(2-форма):

INSERT [*LOW_PRIORITY* | *DELAYED*] [*IGNORE*]

[*INTO*] *жадвал_номи* [(*устун_номи*,...)]

SELECT ...

INSERT буйруғининг бу формасида бошқа жадвал ёки жадваллардан олинган сатрларни киритади.

13.18 - мисол. subj_lect жадвалининг ўқитувчининг идентификатори (*lecturer_id*) майдонига lecturer жадвалидан шарт бўйича(фамилияси

'Qobulov', университет идентификатори 10 га тенг) танлаб олинган сатрни кўйиш.

```
mysql> INSERT INTO subj_lect (lecturer_id)
      SELECT lecturer_id FROM lecturer
      WHERE surname='Qobulov'
      AND univ_id='10';
```

Синтаксиси(3-форма):

INSERT [*LOW_PRIORITY* / *DELAYED*] [*IGNORE*]

[*INTO*] *жадвал_номи*

SET *устун_номи* = *қиймат*,

устун_номи = *қиймат*, ...

13.19 - мисол. Student жадвалига surname майдони 'Zuparov', name майдони 'Odil' қийматига тенг сатрни киритинг.

```
mysql> INSERT INTO Student
      SET surname='Zuparov',
      name='Odil';
```

INSERT ... VALUES бир нечта қийматдан иборат рўйхатли формаси *MySQL* 3.22.5 ва ундан кейинги версияларида қўлланилади. *устун_номи* = *қиймат* ифодалари синтаксиси *MySQL* 3.22.10 ва ундан кейинги версияларида қўлланилади.

Жадвалга маълумотларни киритишда қуйидагиларга амал қилинади:

- Агар **INSERT ... VALUES** ёки **INSERT ... SELECT** лар учун устунлар рўйхати кўрсатилмаган бўлса, у ҳолда барча устунлар учун қийматлар рўйхати(**VALUES()**) ёки **SELECT** ишининг натижаси аниқланган бўлиши керак. Агар жадвалда устунлар тартиби аниқланмаган бўлса, уни олиш учун **DESCRIBE жадвал_номи** буйруғини ишлатилади.

- Қиймати аниқ кўрсатилмаган ихтиёрий устун учун қиймат одатга кўра ўрнатилади. Масалан, агар берилган устунлар рўйхатида жадвалнинг барча устунлари кўрсатилмаган бўлса, у ҳолда тушириб қолдирилган устунларга одатга кўра қиймат ўрнатилади.

- *expression* ифодаси аввалроқ қийматлар рўйхатига киритилган ихтиёрий устунларга тегишли бўлиши мумкин. Масалан, уни қуйидагича кўрсатиш мумкин:

```
mysql> INSERT INTO жадвал_номи (col1,col2)
      VALUES (15,col1*2);
```

қуйидагича кўрсатиш мумкин эмас:

```
mysql> INSERT INTO жадвал_номи (col1,col2)
      VALUES (col2*2,15);
```

Юқоридаги учта буйруқ формасида иштирок этувчи зарур бўлмаган учта параметр ҳақида тўхталамиз: **LOW_PRIORITY**, **DELAYED** ва **IGNORE**.

LOW_PRIORITY ва **DELAYED** параметрларидан кўп миқдордаги фойдаланувчилар жадвал билан ишлаётганда фойдаланишади. Улар берилган амал бошқа фойдаланувчилар амалларидан муҳимлигини (биринчи бажарилиши кераклигини) ўрнатишга кўрсатма беради. Агар **LOW_PRIORITY** калит сўзи кўрсатилса, берилган **INSERT** буйруғини бошқа мижозлар бу жадвални ўқишни тугатгунча тўхтатиб турилади. Бу ҳолатда мижоз берилган киритиш буйруғи ўз ишини тугатгунча кутиб туриши шарт. Чунки бу пайтда шиддат билан ишлаётган жадвал анчагина вақт талаб

килиши мумкин. *INSERT* нинг бу буйруғининг акси *DELAYED* эса мижоз учун бошқа фойдаланувчиларга боғлиқ бўлмаган ҳолда(мустақил равишда) амални ҳозирок давом эттиришга имкон яратади.

Агар *INSERT* буйруғида *IGNORE* калит сўзи кўрсатилса, бу жадвалдаги *PRIMARY* или *UNIQUE* такрорий калитларга эга бўлган барча сатрлар бекор қилинади ва жадвалга киритилмайди. Агар *IGNORE* кўрсатилмаган бўлса, мавжуд калит қийматлари такрорланган сатрлар топилганда бажарилаётган киритиш амали тўхтатилади.

13.20 – мисол. Аввалги маърузаларда яратилган талабалар порталининг жавалларига маълумотларни киритамиз:

```
mysql> INSERT INTO Student VALUES
(1, 'Islomov', 'Ikrom', 150, 1, 'Andijon', '3-12-1992', 10),
(3, 'Fozilov', 'Fozil', 200, 3, 'Qashqadaryo', '1-12-1990', 10),
('6, 'Salimov', 'Vohid', 150, 4, 'Namangan', '7-06-1989', 22),
(10, 'Karimov', 'Botir', 0, 2, 'Jizzax', '8-12-1991', 10);
mysql> INSERT INTO lecturer VALUES
(24, 'Qobulov', 'Botir', 'Toshkent', 10),
(46, 'Nasriddinov', 'Islom', 'Toshkent', 10),
(74, 'Hasanov', 'Farhod', 'Namangan', 22),
(108, 'Sobirov', 'Nizom', 'Namangan', 22);
mysql> INSERT INTO subject VALUES
(10, 'Informatika', 56, 1),
(22, 'Fizika', 34, 1),
(43, 'Matematika', 56, 2);
mysql> INSERT INTO univercity VALUES
(22, 'NamDU', '34', 'Namangan'),
(10, 'TDPU', '29', 'Toshkent'),
(11, 'AnDU', '60', 'Andijon');
mysql> INSERT INTO exam_marks VALUES
(10, 12, 10, 5, '12-01-2010'),
(34, 32, 10, 4, '23-01-2010'),
(75, 55, 10, 5, '05-01-2010'),
(238, 12, 22, 3, '17-06-2009');
mysql> INSERT INTO subj_lect VALUES
(24, 24),
(46, 46),
(74, 74),
(108, 108);
```

Маълумотлар базасидан маълумотларни олиш: *SELECT* оператори
SELECT оператори танланган бир ёки бир нечта жадвалдан сатрлар(ёзувлар)ни олиш учун ишлатилади. Унинг ёрдамида саралаш амалга ошириладиган *table_references* жадвалидан олиниши керак бўлган ифода (*select_ифода*) ёки устунлар, ҳамда ушбу устундаги маълумотларга мос келувчи (*where_definition*) шартлар, бу маълумотларни чиқариш тартиби - *group_type* ва натижавий маълумотларни қанчаси барузерга чиқазилиш керак - *limit_criteria* да кўрсатилади.

Синтаксиси:

```
SELECT select_ифода1, select_ифода2,
...
[FROM table_references
[WHERE where_definition]
[ GROUP BY group_type ]
```


[*HAVING where_definition*]

[*ORDER BY {сон / устун_номи / формула} [ASC / DESC], ...*]

[*LIMIT limit_criteria*] ;

[] тўртбурчак қавс, унинг ичидаги параметрларни бериш шарт эмаслигини билдиради, | тўғри чизик эса бўлиши мумкин бўлган вариантлар. **ORDER BY** калит сўздан сўнг *устун_номи*, *сон* (белгисиз бутун) ёки *формула* ва тартиблаш усули (ўсиш тартиби бўйича – **ASC** ёки камайиш тартиби бўйича – **DESC**) кўрсатилади. Одатга кўра тартиблаш усули берилмаганда тартиблаш ўсиш тартибида амалга оширилади.

select_ифода ўрнида «*» ёзилса, бу барча устунларни чиқазиш кераклигини англатади. *select_ифода* да «*» дан ташқари *max*, *min* ва *avg* каби функциялари ҳам ишлатилади.

Бундан ташқари **SELECT** оператори ёрдамида бирор-бир жадвални ишлатмасдан, бирор-бир сатрни олмасдан ҳисоблаш амалларини ҳам бажариш мумкин.

масалан, 2*2 нечага тенглигини ҳисоблаш учун, қуйидагича ёзиш керак:

```
mysql> SELECT 2*2;
4
```

WHERE конструкцияси ишлатилганда *where_definition* параметрида маълумотларни МБ дан олиш учун шартлар берилади. Ушбу шартлар турли ифодадар ёки амаллар бўлиши мумкин.

13.7. - жадвал. **WHERE** конструкцияси билан ишловчи операторлар

Амал	Номи	Мисол	Вазифаси
=	тенг	stipend=150	Иккала қиймат тенглигини текширади
>	катта	stipend>100	Битта қиймат иккинчисидан катталигини текширади
<	кичик	stipend<150	Битта қиймат иккинчисидан кичиклигини текширади
>=	Катта ёки тенг	stipend>=100	Битта қиймат иккинчисидан катта ёки тенг муносабатини текширади
<=	Кичик ёки тенг	stipend<=150	Битта қиймат иккинчисидан кичик ёки тенг муносабатини текширади
!= ёки о	Тенг эмас	hour!=0	Иккала қиймат тенг эмаслигини текширади

13.21 - мисол. Student жадвалидан 'Vohid' исмли талабаларни барча маълумотларини чиқазинг:

```
mysql> SELECT * FROM Student
WHERE name='Vohid';
```

Натижа:

stud_id	surname	Name	stipend	kurs	city	birthday	univ_id
6	Salimov	Vohid	150	4	Namangan	7-06-1989	22

13.22 - мисол. Университет идентификатори 10 га тенг бўлган ўқитувчилар(lecturer жадвалидан)нинг исми ва шаҳрини(name,city) чиқазинг:

```
mysql> SELECT name, city
FROM lecturer WHERE univ_id=10;
```

Натижа:

name	city
Botir	Toshkent
Islom	Toshkent

13.23 - мисол. Student жадвалидан барча талабалар рўйхатини танлаш.

```
mysql> SELECT *
FROM student;
```

Натижа:

stud_id	surname	Name	stipend	kurs	city	birthday	univ_id
1	Islomov	Ikrom	150	1	Andijon	3-12-1992	10
3	Fozilov	Fozil	200	3	Qashqadaryo	1-12-1990	10
6	Salimov	Vohid	150	4	Namangan	7-06-1989	22
10	Karimov	Botir	0	2	Jizzax	8-12-1991	10

Агар ушбу сўровга **ORDER BY** surname ифодаси қўшилса, у холда рўйхат фамилия бўйича тартибланади. Одатга кўра тартиблаш, ўсиш тартиби бўйича бажарилади. Агар камайиши бўйича тартиблаш керак бўлса, у холда охириги ифодадаги атрибут номидан кейин **DESC** сўзи қўшилади.

SQL тилида шарт ифодаларини тузиш учун солиштириш ва мантикий операторлардан ташқари яна бир қатор *махсус операторлар* қўлланилади. Бу операторлар бошқа дастурлаш тилларида мавжуд эмас.

13.24 - жадвал. *WHERE* конструкцияси билан ишловчи махсус операторлар

Оператор	Номи	Мисол	Вазифаси
IS NOT NULL	Манзили нолга тенг эмас (бўш эмас)		Майдон қиймати мавжудлигини текширади
IS NULL	Манзили нолга тенг (бўш)		Майдон қиймати мавжуд эмаслигини текширади
BETWEEN	оралиғида	BETWEEN 2 AND 3	бирор қийматлар диапазониға тегишлилигини текшириш
IN	оралиғида	IN (2,3)	бирор қийматлар тўпламиға тегишлилигини текшириш
NOT IN	Оралиғида эмас	NOT IN (2,3)	бирор қийматлар тўпламиға тегишли эмаслигини текшириш
LIKE	мослик	name like('Anvar%')	намуна билан мослигини текшириш

NOT LIKE	мослик	name not like('Anvar%')	намуна билан мос эмаслигини текшириш
REGEXP	Доимий ифода	name regexp	Қиймат доимий ифодага мослигини текширади

13.25 - мисол. Фамилияси К ҳарфи билан бошланувчи талабалар рўйхатини танлаб олиш.

Бундай ҳолатда **LIKE** операторидан фойдаланиш қулай ҳисобланади. **LIKE** оператори фақат белгили майдонлар учун қўлланилади ва майдон қиймати операторда кўрсатилган намунага мослигини текшириш имконини яратади. Намуна қуйидаги махсус белгилардан ташкил топади:

_ (тагига чизиш белгиси) – битта ихтиёрий белгини билдиради;

% (фоиз белгиси) – ихтиёрий миқдордаги белгилар кетма – кетлигини билдиради.

```
mysql> SELECT surname, kurs
FROM student
WHERE surname LIKE 'K%';
```

Натижа:

surname	kurs
Karimov	2
Komilov	5

Кўпинча устунлардаги минимал, максимал ёки ўртача қийматларни ҳисоблашга тўғри келади. Масалан, келтирилган имтиҳон натижалари жадвалида ўртача баҳони ҳисоблаш мумкин. Бундай ҳисоблашни бажариш учун SQL тилида махсус **агрегат(мураккаб амалларни бажарувчи) функциялари** мавжуд:

MIN() – устундаги минимал қиймат;

MAX() – устундаги максимал қиймат;

SUM() – устундаги қийматлар суммаси;

AVG() – устундаги қийматларни ўртачаси;

COUNT() – устундаги NULL дан фарқли қийматлар миқдори.

13.26 - мисол. Талабалар олган баҳоларнинг ўртачасини ҳисобланг.

```
mysql> SELECT AVG(mark)
FROM exam_marks;
```

Натижа:

AVG(mark)

4.25

Агрегат функцияларини ҳам **WHERE** сўзи билан биргаликда қўллаш мумкин.

13.27 - мисол. Идентификатор номери 55 га тенг бўлган талабанинг имтиҳонларда олган баҳоларининг ўртачасини ҳисобланг:

```
mysql> SELECT AVG(mark)
FROM exam_marks
WHERE stud_id = 75;
```

Натижа:

AVG(mark)

5.00

13.28 - мисол. Идентификатор номери 10 га тенг бўлган фан бўйича талабаларнинг олган баҳоларини ўртачасини ҳисобланг:

```
mysql> SELECT AVG(mark)
FROM exam_marks
WHERE subj_id = 10;
```

Натижа:

AVG(mark)

4,666666666666667

Бирор хулосага келишдан олдин SQL тилининг барча имкониятларини кўриб чиқиш керак. Масалан, ихтиёрий матнни сўров таркибига киритиш мумкин. 13.28 – мисолни қуйидагича ёзамиз:

```
mysql> SELECT 'Ўртача балл =', AVG(mark)
FROM exam_marks
WHERE subj_id = 10
```

Натижа:

Ўртача балл =**4,666666666666667**

Агрегат функцияларини жадвал устунларининг гуруҳланган қийматлари учун ҳам қўллаш мумкин. Бу махсус **GROUP BY** конструкцияси бўлиб, унда кўрсатилган устун қийматлари бўйича гуруҳлаш амалга оширилади.

13.29 - мисол. Фанлар бўйича талабаларнинг олган баҳоларини ўртачасини ҳисобланг ва фанларни ўсиш тартиби бўйича гуруҳлаб чиқазинг:

```
mysql> SELECT subj_id, AVG(mark)
FROM exam_marks
GROUP BY subj_id;
```

Натижа:

subj_id	mark
10	4,666666666666667
22	3

Бу имконият ҳам одатдагидек **WHERE** сўзи билан биргаликда қўлланилиши мумкин. Бу сўровни бажаришда МББТ аввал жадвалдан WHERE ифодасидаги шартни қаноатлантирувчи сатрларни танлаб олади, кейин танланган сатрларни гуруҳлаш ва агрегатлаш амалини бажаради.

13.30 - мисол. Фанлар бўйича 4 баҳодан юқори баҳо олган талабаларни идентификатор номери бўйича гуруҳлаб чиқазинг:

```
mysql> SELECT stud_id, mark
FROM exam_marks
WHERE mark >=4
GROUP BY stud_id;
```

Натижа:

stud_id	mark
12	5
32	4
55	5

Гуруҳлаш биттадан кўп майдонлар бўйича ҳам бажарилиши мумкин. GROUP BY секциясига эга бўлган сўровлар учун қуйидаги муҳим чекланиш мавжуд: бундай сўровлар натижаси таркибига гуруҳлаш бажарилган устунлар ва агрегатлаш натижасига эга бўлган устунлар кирази.

Агрегат ва гуруҳлаш функцияларини ишлатишда эътибор бериш талаб этилади:

ANSI SQL да агар агрегат функциялари ёки GROUP BY конструкцияси ишлатилса, **SELECT** конструкцияси фақатгина агрегат функцияларини ва **GROUP BY** конструкциясида кўрсатилган устунларни қабул қилади. Агар **GROUP BY** конструкциясида устунни ишлатиш талаб этилса, у албатта **SELECT** конструкциясида кўрсатилиши керак.

MySQL эса амаллардан чексиз фойдаланишни таъминлайди ва **SELECT** конструкциясида керак бўлмаган элементларни олиб ташлаш имкониятини берувчи кенгайтирилган систаксисни қўллайди.

Маълумотларни гуруҳлаш ва мураккаб амалларни бажаришга қўшимча равишда **HAVING** конструкцияси ёрдамида мураккаб натижаларни текширишнинг барча имкониятлари мавжуд. У **GROUP BY** конструкциясидан сўнг ишлатилади ва **WHERE** га ўхшаш бўлиб, фақатгина гуруҳларга қўлланилади.

Юқоридаги мисолни кенгайтирамиз, фанлар бўйича ўртача ўзлаштириш кўрсаткичи 4 дан юқори бўлган фанларни чиқазамиз:

```
mysql> SELECT subj_id, AVG(mark)
FROM exam_marks
GROUP BY subj_id
HAVING AVG(mark)>4;
```

Натижа:

Эътибор беринг, **HAVING** конструкцияси гуруҳга мурожаат этипти. Ушбу сўров қуйидагича натижани беради:

subj_id	mark
10	4,666666666666667

Ушбу сўров натижасида фойдаланувчи фақат оддий сонларни эмас, балки матнни ҳам кўради.

SELECT операторининг Web-хужжатларда фойдаланиш мумкин бўлган **LIMIT** конструкцияси билан танишамиз. Бу конструкцияда экранга натижанинг нечта сатри чиқазилиши кўрсатилади. Ушбу конструкцияда иккита параметр кўрсатилиши зарур: сатр номери(чиқариладиган сатрларнинг бошланғич номери) ва сатрлар сони. Сатрларни номерлаш нолдан бошланади.

13.31 - мисол. Талабалар жадвалидан барча талабаларнинг исмларини олиб, натижанинг иккинчи сатридан бошлаб иккита сатр чиқазинг.

```
mysql> SELECT name
FROM student
LIMIT 1,2;
```

Натижа:

stud_id	surname	Name	stipend	kurs	city	birthday	univ_id
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	10
6	Salimov	Vohid	150	4	Namangan	7/06/1989	22

Масалан, МБ дан олинган талабалар рўйхати кўпгина сатрлардан иборат уларни алоҳида саҳифаларда тасвирлаш учун натижани саҳифага бўлиб – бўлиб чиқазинг лозим. Ушбу ҳолларда **LIMIT** конструкцияси фойдаланиш учун жуда қулай.

Маълумотни бир нечта жадваллардан олиш*Иккита жадвални боғлаш*

13.32 - мисол. Андижонда туғилган талабаларни исми, фамилияси ва таълим муассасаси номини чиқазинг.

```
mysql> SELECT student.surname, student.name, univercity.univ_name
FROM student, univercity
Where student.city= 'Andijon'
and student.univ_id=univercity.univ_id;
```

Натижа:

surname	Name	
Islomov	Ikrom	TDPU

13.33 - мисол. Имтиҳонларда 4 ёки 5 баҳо олган талабаларни рўйхатини чиқазинг.

Бунинг учун WHERE сўзидан кейин солиштириш амалларини (<, >, =, <> ва х.к.) ва мантикий операторлар қатнашган ифодаларни жойлаштириш мумкин.

```
mysql> SELECT student.surname, student.name, subject.subj_name,
exam_marks.mark
FROM student, subject, exam_marks
WHERE student.stud_id=exam_marks.stud_id
AND subject.subj_id= exam_marks.subj_id
AND (( mark >= 4 ) AND ( mark<= 5 ));
```

Натижа:

surname	name	subj_name	mark
Zoitova	Ozoda	Informatika	5
Komilov	Farhod	Informatika	4
Boltayeva	Vasila	Informatika	5

13.33 - мисолни IN операторидан фойдаланиб ёзамиз. (имтиҳонларда камида битта 4 ёки 5 баҳо олган талабаларни фамилияси, исми, олган баҳосини танлаб олиш)

```
mysql> SELECT student.surname, student.name, subject.subj_name,
exam_marks.mark
FROM student, subject, exam_marks
WHERE student.stud_id=exam_marks.stud_id
AND subject.subj_id= exam_marks.subj_id
AND mark IN (2,3);
```

Натижа:

surname	name	subj_name	mark
Zoitova	Ozoda	Informatika	5
Komilov	Farhod	Informatika	4
Boltayeva	Vasila	Informatika	5

Ушбу натижани BETWEEN операторидан фойдаланиб ҳам олиш мумкин:

```
mysql> SELECT student.surname, student.name, subject.subj_name,
exam_marks.mark
FROM student, subject, exam_marks
WHERE student.stud_id=exam_marks.stud_id
AND subject.subj_id= exam_marks.subj_id
AND mark BETWEEN 2 AND 3;
```

Натижа:

surname	name	subj_name	mark
Zoitova	Ozoda	Informatika	5
Komilov	Farhod	Informatika	4
Boltayeva	Vasila	Informatika	5

Маълумотлар базасидаги сатрларни(ёзувларни) янгилаш: UPDATE оператори

UPDATE оператори жадвалда мавжуд устун қийматини мос киритилган қийматга ўзгартиради.

Синтаксиси:

UPDATE [*LOW_PRIORITY*] [*IGNORE*] *жадвал_номи*

SET *устун_номи1=ифода1*

[*, устун_номи2=ифода2, ...*]

[*WHERE where_definition*]

[*LIMIT limit_criteria*]

Бу ерда, SET ифодасида айнан қайси устун ўзгартирилиши ва қандай қиймат унга ўрнатилиши кераклиги кўрсатилади. WHERE дан сўнг, янгиланиши керак бўлган сатрлар кўрсатилади, агар ҳеч қандай сатр кўрсатилмаса, барча сатрлар янгиланади. Агар ORDER BY ифодаси берилган бўлса, у ҳолда сатрлар унда кўрсатилган тартибда янгиланади.

Агар LOW_PRIORITY калит сўзи кўрсатилса, берилган UPDATE буйруғини бажариш бошқа мижозлар бу жадвални ўқишни тугатгунча тўхтатиб турилади.

Агар IGNORE калит сўзи кўрсатилса, у ҳолда янгилаш буйруғи тўхтатилмайди, хатто агар калитлар такрорланса ҳам хато чиқазмайди. Конфликт ҳолатлар олиб келадиган сатрлар янгиланмайди.

Агар устунга янги қиймат берилаётган ифодада ушбу майдон(устун) номи ишлатилса, у ҳолда UPDATE ушбу устун учун унинг жорий қийматини ишлатади.

13.34 - мисол. hour устунига унинг жорий қийматидан битта катта қийматни ўрнатинг:

```
mysql> UPDATE subject
SET hour=hour+1;
```

MySQL 3.23 версиясида LIMIT # параметрини фақатгина # да берилган сондаги сатрлар ўзгартирилганлигига ишонч ҳосил қилиш учун ишлатилади.

13.35 - мисол. Жадвалдаги биринчи сатрни кўрсатилган тартибда ўзгартиринг.

```
mysql> UPDATE subject
SET hour=55 Limit 1;
```

Жадвал яратилгандан сўнг ўзгартириш: ALTER TABLE оператори

ALTER TABLE оператори мавжуд жадвал структурасини ўзгартириш учун имкон яратади. Масалан, жадвалга устун қўшиш ёки олиб ташлаш, индекс яратиш ёки ўчириш, устун номини ўзгартириш каби амалларни

бажаради. Хаттоки жадвал учун изохни ва уни типини ҳам ўзгартириш мумкин.

Синтаксиси:

ALTER [IGNORE] TABLE *жадвал_номи*

alter_specification

[, *alter_specification* ...]

Жадвални ўзгартириш учун қуйидагилар бажарилади (қуйидагиларнинг барчаси *alter_specification* ўрнида ёзилади):

•майдон қўйиш:

ADD [COLUMN] *устунларни_аниқлаш* [**FIRST** | **AFTER** *устун_номи*]

ёки

ADD [COLUMN] (*устунларни_аниқлаш1*, *устунларни_аниқлаш2*,...)

бу ва бундан кейин *устунларни_аниқлаш* жадвал яратишда қандай ёзилган бўлса, худди шундай ёзилади.

•индексларни қўйиш:

ADD INDEX [*индекс_номи*] (*индексланган_устун_номи*,...) ёки

ADD PRIMARY KEY (*индексланган_устун_номи*,...) ёки

ADD UNIQUE [*индекс_номи*] (*индексланган_устун_номи*,...) ёки

ADD FULLTEXT [*индекс_номи*] (*индексланган_устун_номи*,...)

•майдонни ўзгартириш:

ALTER [COLUMN] *устун_номи*

{**SET DEFAULT** *literal* | **DROP DEFAULT**}

ёки

CHANGE [COLUMN] *устуннинг_эски_номи* *устунларни_аниқлаш*

ёки

MODIFY [COLUMN] *устунларни_аниқлаш*

•калит, индекс, майдонни ўчириш:

DROP PRIMARY KEY

DROP INDEX *индекс_номи*

DROP [COLUMN] *устун_номи*

•жадвални номини ўзгартириш:

RENAME [TO] *жадвалнинг_янги_номи*

•жадвал майдонларини қайта тартиблаш:

ORDER BY *майдон*

ёки

жадвал_опцияси

ALTER TABLE оператори ёрдамида устун типини ўзгартиргандан сўнг бу буйруқ бажарилганлигини текшириш учун **DESCRIBE** *жадвал_номи* буйруғи ишлатилади. Агар устун типи ўзгартирилмаган бўлса, демак *MySQL* маълумотлар шаклини ўзгартиришни битта сабабга кўра инкор этган. Масалан, устун типини **VARCHAR** дан **CHAR** га ўзгартиришда, берилган жадвал ўзгарувчан узунликка эга устунлардан таркиб топган бўлса, *MySQL* яна **VARCHAR** ни ишлатишда давом этади.

ALTER TABLE ишлаш жараёнида мавжуд жадвалнинг вақтинчалик нусхасини яратади. Талаб қилинаётган ўзгартиришлар нусха устида

бажарилади, сўнг эски жадвал ўчирилади ва уни ўрнига янги яратилган жадвал номи қайта номланади. Бу барча ўзгартиришлар (муваффақиятсизларидан ташқари) янги жадвалга автоматик тарзда тушиши учун бажарилади. ALTER TABLE ишлаётган вақтда, эски жадвал бошқа мижозларни ишлаши учун очик бўлади. Жадвалга маълумотларни киритиш ва ўзгартириш вақтинча тўхтатиб турилади ва қачонки янги жадвал тайёр бўлгандан сўнг, бу амаллар бажарилади.

Агар IGNORE аниқланган бўлса, у ҳолда нусха кўчириш бекор қилинади. ALTER TABLE да RENAME дан бошқа ихтиёрий опция бажарилганда, MySQL ҳар доим вақтинчалик жадвал нусхасини яратади.

13.36 - мисол. Student жадвалига изох ёзиш учун янги майдон қўшамиз:

```
mysql> ALTER TABLE Student
ADD comment varchar(255) AFTER univ id;
```

Демак, жадваллар билан ишлашни: уларни яратиш, ўчириш ва ўзгартиришни ўргандик. Энди бу жадвалларда сақланаётган маълумотлар билан ишлашни: маълумотларни киритиш, ўчириш ва ўзгартиришни ўрганамиз.

Маълумотлар базасидан сатрларни(ёзувларни)ўчириш: DELETE оператори

DELETE оператори *жадвал_номи* жадвалидан *where_definition* шартини қаноатлантирувчи сатрни ўчиради ва ўчирилган ёзувлар сонини чиқазади.

Агар **DELETE** оператори **WHERE** сиз ишлатилса, барча сатрлар ўчирилади.

Синтаксиси:

DELETE [**LOW_PRIORITY**] **FROM** *жадвал_номи*

[**WHERE** *where_definition*]

[**LIMIT** *rows*]

13.37 - мисол. subj_lect жадвалидан ўқув фани идентификатори майдони бўш бўлган барча сатрларни ўчириш.

```
mysql> DELETE FROM subj_lect WHERE subj id='' '';
```

13.38 - мисол. Жадвалдаги барча ёзувларни ўчириш учун қуйидаги буйруқдан ҳам фойдаланиш мумкин.

```
mysql> DELETE FROM Student WHERE 1>0;
```

Қуйидаги кўриниши ушбу буйруқни шартсиз ишлатишдан кўра секинроқ ишлайди:

```
mysql> DELETE FROM student;
```

MySQL нинг махсус **DELETE** буйруғи учун **LIMIT** опцияси мижозга бошқарувни қайтаргунча ўчирилган сатрларни максимал сонини тизимга кўрсатади.

Жадвални ўчириш: DROP TABLE оператори

DROP TABLE оператори бир ёки бир нечта жадвалларни ўчиради. Бу буйруқ ишлатилганда жадвалнинг барча маълумотлари ва структураси ўчирилади, шунинг учун бу буйруқ билан ишлаганда эҳтиёт бўлиш керак.

Синтаксиси:

DROP TABLE [IF EXISTS] жадвал_номи

[, жадвал_номи,...]

[RESTRICT / CASCADE]

MySQL 3.22 ва ундан кейинги версияларида агар кўрсатилган жадвал мавжуд бўлмаса, хато ҳақида огоҳлантириш учун IF EXISTS калит сўзи ишлатилади.

RESTRICT ва CASCADE опцияси дастурни бошқа МББТ га ўтказишни соддалаштириш учун имкон яратади.

13.39 - мисол. DROP TABLE операторининг ишлатилиши

```
mysql> DROP TABLE IF EXISTS Student, lecturer, test;
```

Маълумотлар базасини ўчириш: DROP DATABASE оператори

DROP DATABASE оператори бутун маълумотлар базасини ўчириш учун ишлатилади. Бу буйруқ билан ишлаганда ҳам эҳтиёт бўлиш талаб этилади.

Синтаксиси:

DROP DATABASE база_номи;

Натижада барча сатрлар, жадваллар, индекслар(калитлар) ва маълумотлар базасининг ўзи ўчирилади.

Демак, буйруқлар сатри ёрдамида MySQL МББТ билан ишлашни кўриб чиқдик. Энди Mysql МББТ билан ишлаш учун phpmyadmin дан фойдаланамиз.

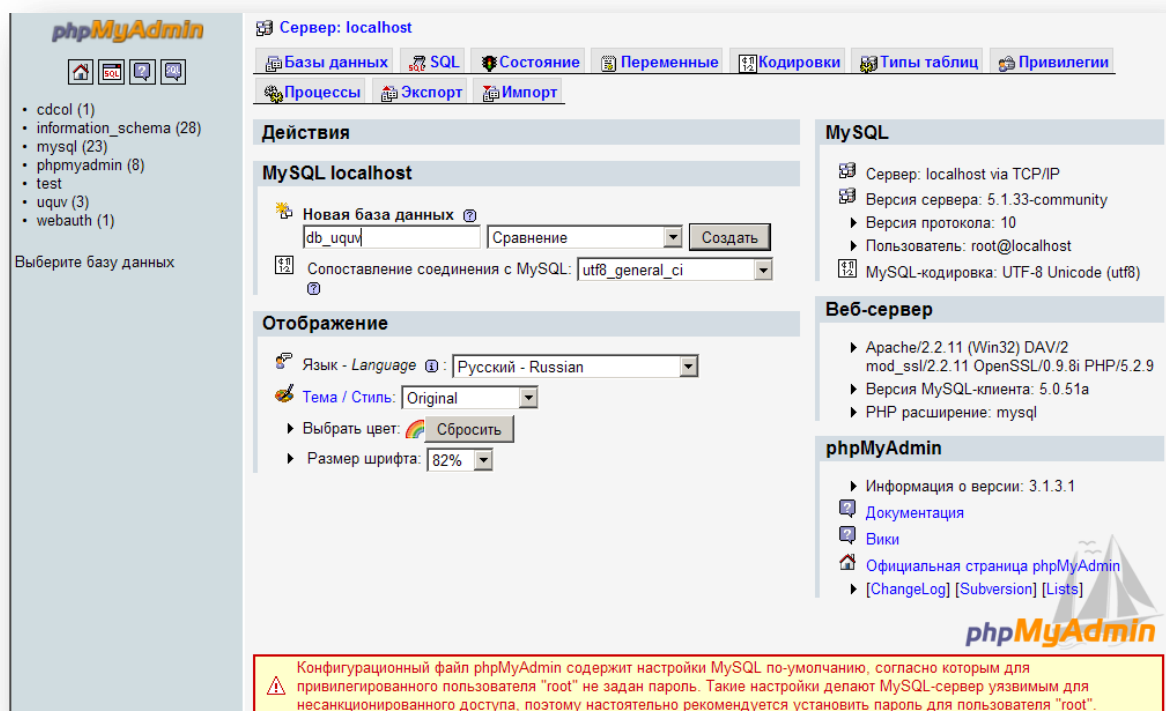
XIV. PHP VA MYSQL НИНГ ЎЗАРО БОҒЛАНИШИ

Бу бобда PHP ва MySQL МББТ нинг ўзаро боғланиши усуллари билан танишасиз. Асосий эътибор маълумотлар базаси билан алоқа ўрнатиш, сўровларни жўнатиш функцияси ва жавобларни қайта ишлашга қаратилади(`mysql_connect`, `mysql_query`, `mysql_result`, `mysql_num_rows`, `mysql_close`). MySQL маълумотлар базаси билан боғланиш учун PhpMyAdmin ни ишлатиш ҳам кўриб чиқилади.

14.1. PhpMyAdmin ёрдамида MySQL маълумотлар базаси билан ишлаш

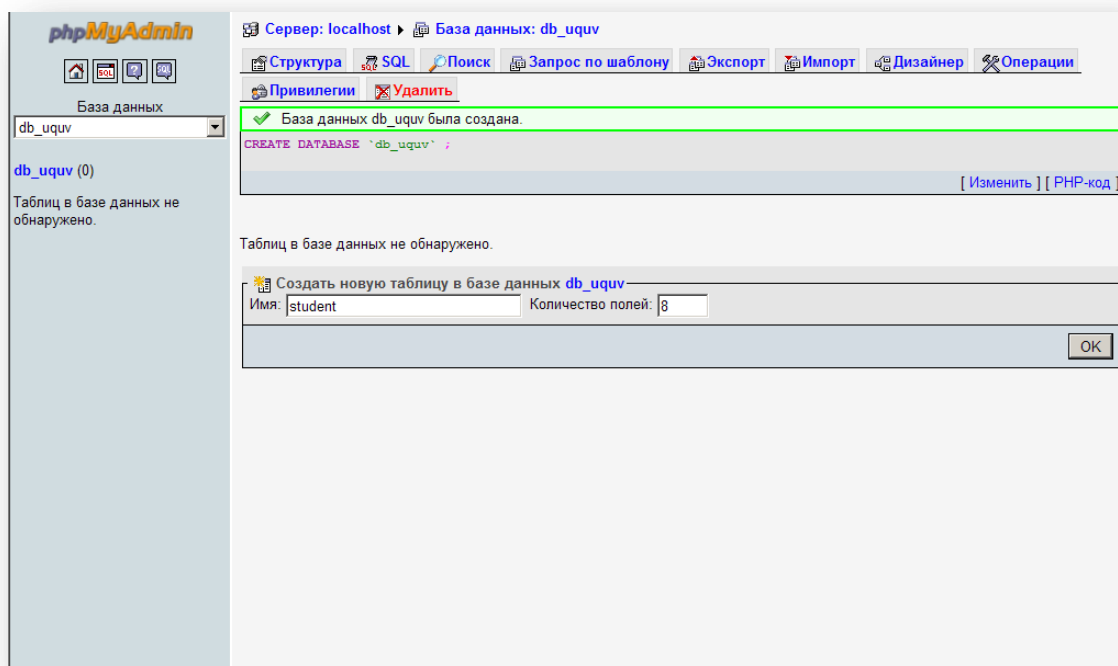
PhpMyAdmin браузернинг барча имкониятларидан тўлиқ фойдаланиш имкониятини беради, жумладан, экранга жойлашмай қолган расмларни суриб кўриш ва б. PhpMyAdminда SQL нинг маълумотлар билан ишловчи кўпгина базавий функциялари Web саҳифалардаги каби гипермуружаатга ўхшаш, интуитив тушунарли ҳаракат ва интерфейслар кўринишига келтирилган. Шунга қарамай, MySQL нинг функцияларни яхшироқ ўзлаштириб олиш учун матнли режимида ҳам ишлаб кўриш тавсия этилади.

Аввал буйруқлар сатрида бажарилган амалларни phpmyadmin да бажариш учун браузернинг манзил қаторида <http://localhost/phpmyadmin/> киритилади. phpmyadmin ёрдамида MySQL МББТ га кириб унда маълумотлар базаси яратилади. Бунинг учун ойнадаги **Новая база данных** сатрида яратилиши керак бўлган маълумотлар базаси номи(`db_uquv`) киритилади(бу ном чап томондаги маълумотлар базалари номлари билан бир хил бўлмаслиги керак). Ва *создать* тугмаси босилади(14.1-рasm).



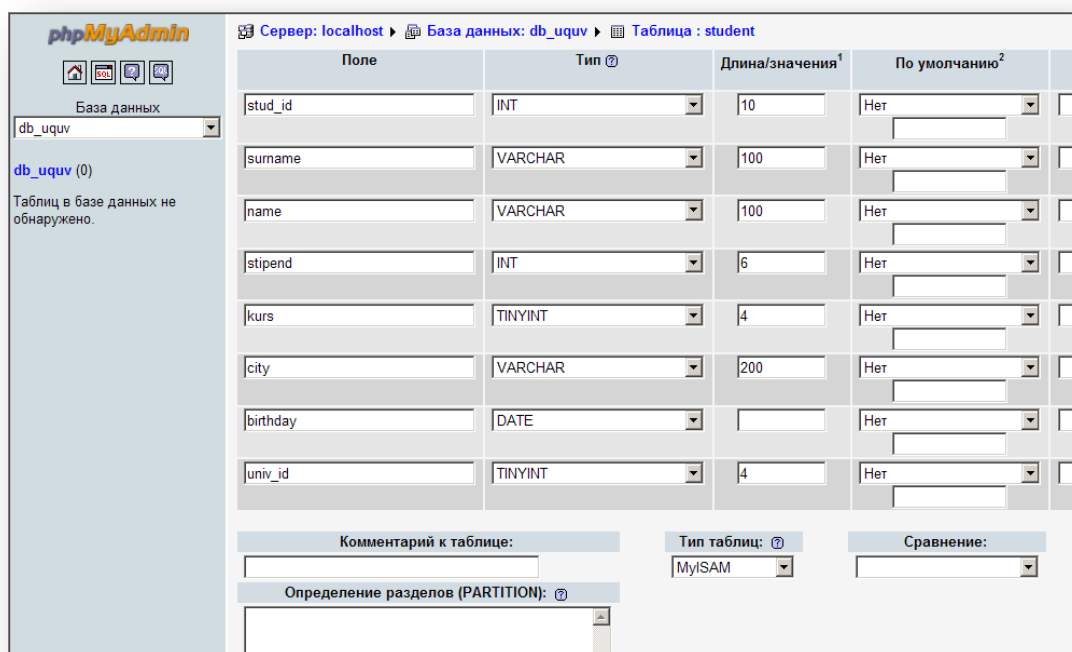
14.1- расм. Янги маълумотлар базасини яратиш.

Маълумотлар базаси яратилгандан сўнг уни таркибидаги жадваллар яратилади. Ушбу ойнада жадвал номи ва жадвалда бўлиши керак бўлган майдонлар сони кўрсатилади ва ОК тугмаси босилади(14.2 -рasm).



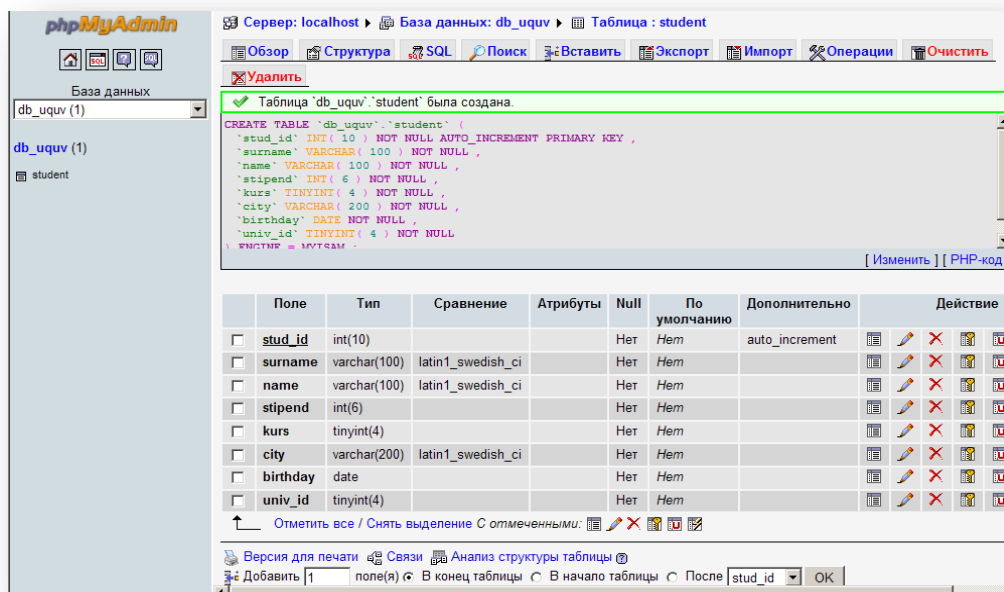
14.2 - расм. Маълумотлар базасида янги жадвал яратиши

Бу ойнада яратилган жадвалнинг майдонлари номи, типи, узунлиги кўрсатилади. Бунга қўшимча равишда майдонларга autoincrement кўрсаткичи ва калит(биринчи даражали ва индексли калит) қўйиш мумкин(14.3 -рasm).



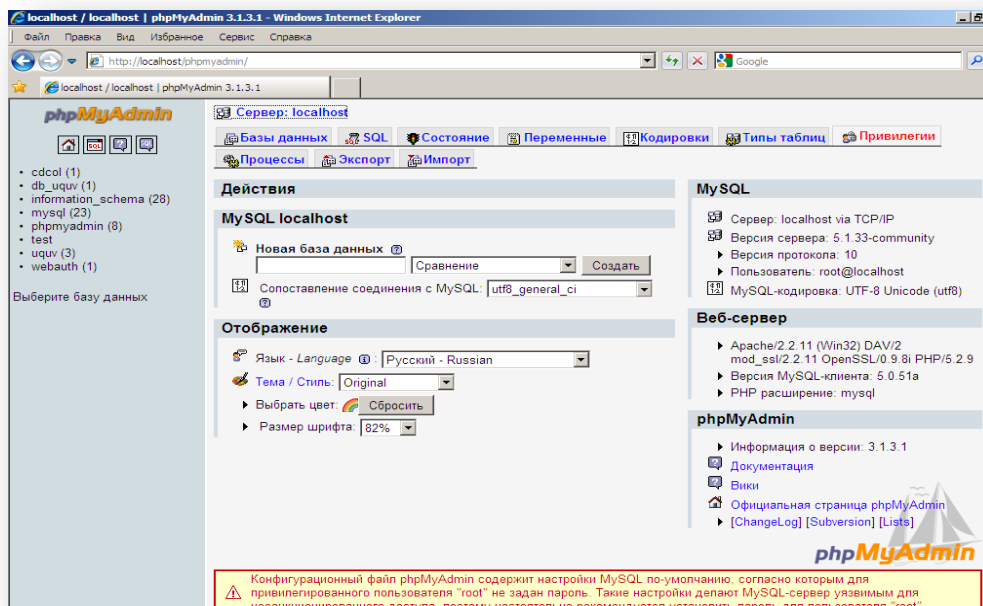
14.3 –расм. Жадвал майдонларини киритиши

Худди шундай бошқа жадваллар ҳам яратилади ва html форма ёки phpmyadmin ёрдамида бу жадвалларга маълумотлар киритилади.



14.4 - расм. Яратилган жадвал майдонлари

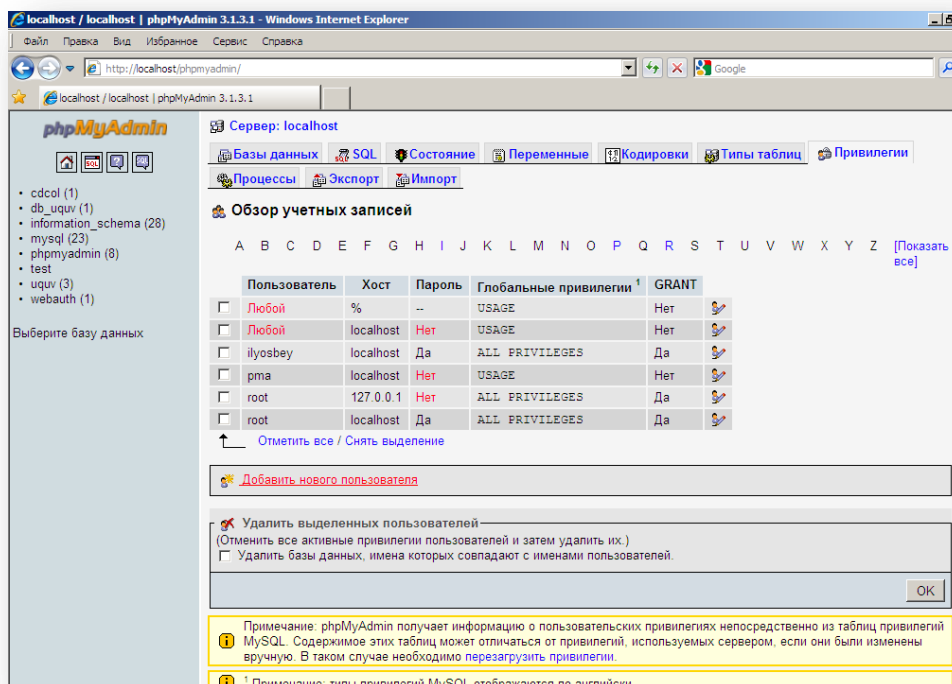
Демак маълумотлар базаси ва маълумотлар базасида жадваллар яратишни кўриб чиқдик. Энди ушбу MySQL МББТ га браузер оркали боғланиб, яратилган маълумотлар базаси билан ишлаш учун фойдаланувчи яратилади. Бунинг учун Сервер:localhost саҳифасида **Привилегии** бўлими танланади(14.5- расм).



14.5 - расм. Фойдаланувчига ваколат бериши.

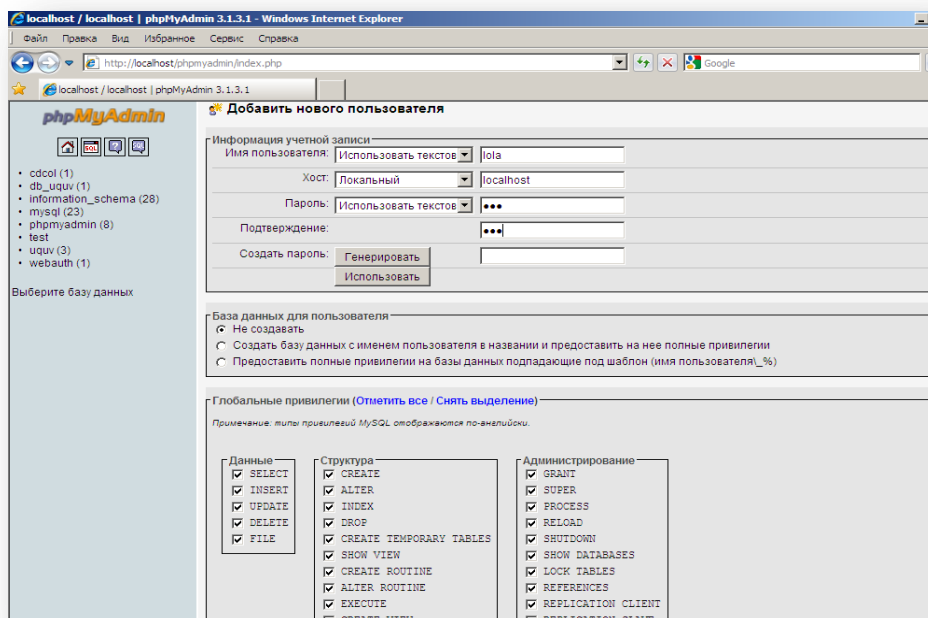
Бу бўлимда MYSQL МББТ билан ишловчи фойдаланувчилар рўйхати ва улар ҳақидаги маълумотлар(МБ билан ишлаш ваколатлари) жойлашган. Янги

фойдаланувчи яратиш учун фойдаланувчилар ойнасидан **Добавить нового пользователя** мурожаати(ссылка) босилади(14.6 - расм).



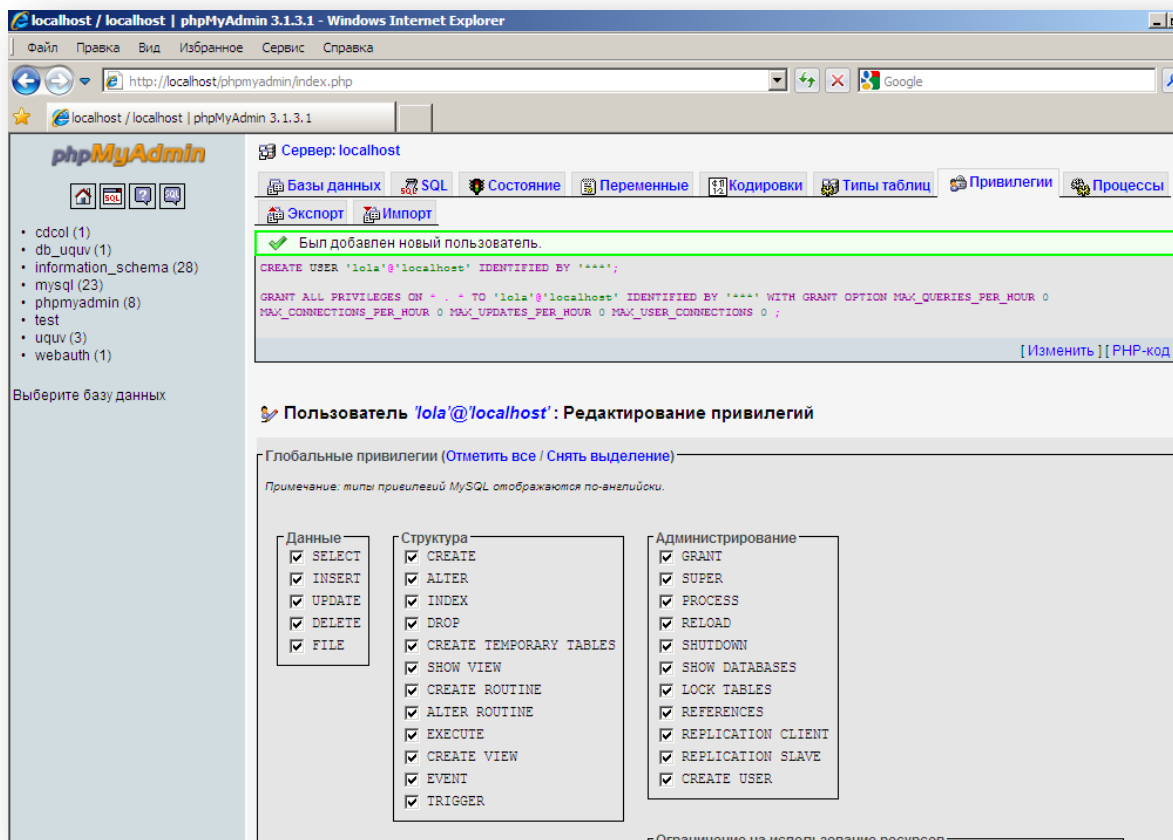
14.6 - расм. Фойдаланувчилар ойнаси

Бу саҳифада янги фойдаланувчининг номи, хост номи, пароли ва МБ билан ишлаш ваколатлари киритилади(14.7 - расм).



14.7 - расм. Янги фойдаланувчи яратиш

Демак бу саҳифада lola исмли фойдаланувчини 135 пароли билан localhost да ишлаши учун барча ваколатларни бердик. Энди бу фойдаланувчи MYSQL тизимига мурожаат эта олиши мумкин(14.8 - расм).



14.8 - расм. Фойдаланувчи ваколатлари билан берилган ойна

14.2. MySQL тизими билан боғланишни ўрнатиш ва тизимда ишлаш учун маълумотлар базасини танлаш.

MySQL маълумотлар базасига WEB орқали PHP технологияси ёрдамида сўровларни жўнатиш алгоритми:

1. Фойдаланувчи томонидан киритилган маълумотларни филтёрлаш ва текшириш.

2. MySQL тизими билан боғланишни ўрнатиш. Агар уриниш муваффақиятсиз якунланса, мос хабар чиқазилади ва жараён якунланади.

3. MySQL тизимида ишлаш учун маълумотлар базасини танлаш. Агар танлаш уриниши муваффақиятсиз якунланса, мос хабар чиқазилади ва жараён якунланади. Сўровларни қайта ишлаш учун бир вақтнинг ўзида бир нечта маълумотлар базасини очишга рухсат берилади.

4. Танланган базада сўровларни қайта ишлаш (маълумотларни қўшиш учун интерфейс яратиш, маълумотлар базасига маълумотларни ёзиш, маълумотлар базасида сақланаётган маълумотларни фойдаланувчига тақдим этиш).

5. Сўровларни қайта ишлаш тугатилгандан сўнг маълумотлар базаси тизими билан боғланишни ёпиш ёки тугатиш.

1. Фойдаланувчи томонидан киритилган маълумотларни филтёрлаш ва текшириш.

Аввал фойдаланувчи томонидан киритилган маълумот атрофидаги тасодифан ёзилган барча ортиқча бўш белгиларни(пробелларни) ўчириш зарур. Бунинг учун trim() функциясини қўлланади.

14.1 – мисол.

```
trim($city);
```

Кейинги босқич — фойдаланувчи барча майдонларни тўлдирганлигини текшириш. trim() функцияси ёрдамида ўзгарувчилар таркибидаги қийматлар тахрирлангандан сўнг ўзгарувчиларни қийматлари мавжуд ёки мавжуд эмаслиги текширилади:

14.2 – мисол.

```
if (!$stud_id || !$name || !$surname || !$kurs || !$univ_id)
{
    echo "Ko'rsatilgan maydonlar to'ldirilmagan. To'ldirilishi zarur
    bo'lgan maydonlarga ma'lumotlarni kiriting va qayta urinib ko'ring." ;
    exit;
}
```

Фойдаланувчи маълумотларини MySQL маълумотлар базасига ёзишдан аввал олдинги маърузаларда кўриб ўтилган addslashes() ва stripslashes() функциялари ёрдамида тахрирлаб олиш мумкин.

htmlspecialchars() функцияси маълумотларда учрайдиган ((&) амперсанд, (<) кичик, (>) катта, (") иккиталик қўштирнок) белгиларини HTML кодировкасига ўтказиш учун ишлатилади.

2. MySQL тизими билан боғланишни ўрнатиш.

mysql_pconnect() функцияси

mysql_connect() функцияси MySQL тизими билан боғланишни ўрнатади.

Синтаксиси:

int mysql_pconnect ([string хост [:порт][:сокетга йўл] [, string фойдаланувчи_номи] [,string парол])

ёки

*resource mysql_pconnect ([server campu [,username campu[,password campu
[, int new_link [, int client_flags]]]])*

хост параметрида MySQL тизимининг ваколатлар жадвалида кўрсатилган компьютер хости номи берилади. Бу ном MySQL ишлаётган Web-серверга сўровларни йўналтириш учун ишлатилади, бироқ MySQL тизимига масофадан туриб ҳам боғланиш мумкин. Хост номи берилганда шарт бўлмаган параметрлари - *порт* номери, *сокет*(локал хост учун) йўлини ҳам кўрсатиш мумкин. *фойдаланувчи_номи* ва *парол* параметрлари MySQL ваколатлари жадвалида берилган фойдаланувчи номи ва пароли билан мос тушиши керак. Агар *хост* параметри берилмаган бўлса, mysql_pconnect() локал хост билан алоқани ўрнатишга ҳаракат қилади.

14.3 – мисол. MySQL билан очиқ боғланиш:

```
@ $db = mysql_pconnect("localhost", "ilyosbey", "135");
```

Берилган мисолда localhost – компьютер хост номи, ilyosbey-фойдаланувчи номи, 135 эса – парол. mysql_pconnect() функцияси олдида ишлатиладиган @ белгиси боғланиш уриниши муваффақиятсиз якунланганда чиқадиган хатоларни яшириш учун ишлатилади. Бу белги die() функцияси чиқазадиган хатолар ҳақидаги хабарларни алмаштиради.

Агар функцияга бир хил аввалги параметрлар билан икки марта мурожаат этилса, янги боғланиш ўрнатилмайди, лекин мурожаат(ссылка)ни эски боғланишга қайтаради. Бу ҳолатга дуч келмаслик учун, *new_link* параметри ишлатилади. Бу параметр ихтиёрий ҳолатда яна бир боғланишни очишга ёрдам беради.

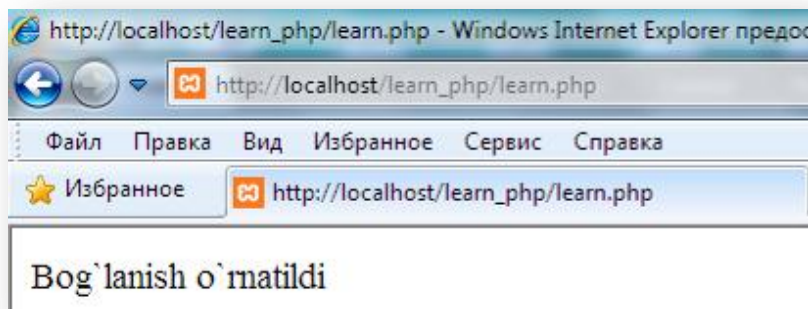
client_flags параметри - бу қуйидаги константаларни комбинацияси: *MYSQL_CLIENT_COMPRESS* (сиқиш протоколини ишлатади), *MYSQL_CLIENT_IGNORE_SPACE* (функция номидан сўнг пробелларни ўқиш учун имконият яратади), *MYSQL_CLIENT_INTERACTIVE* (боғланиш ёпилгунча- wait_timeout билан бирга - interactive_timeout секунд кутади).

new_link параметри PHP 4.2.0, версиясидан, *client_flags* параметри эса PHP 4.3.0 версиясидан бошлаб қўшилган.

Тизим билан боғланиш агар *mysql_close()* функцияси ёрдамида ёпилмаган бўлса, скрипт бажарилиши тугатилганда ёпилади.

14.4 – мисол. ilyosbey фойдаланувчиси учун «135» пароли билан локал сервердаги маълумотлар базасига боғланишни ўрнатамиз:

```
<?
$conn = mysql_pconnect("localhost", "ilyosbey", "135")
or die ("Bog`lanish o`rnatilmadi: ". mysql_error());
echo "Bog`lanish o`rnatildi";
?>
```

Натижа:**14.9 – расм.** Маълумотлар базаси билан муваффақиятли боғланиш

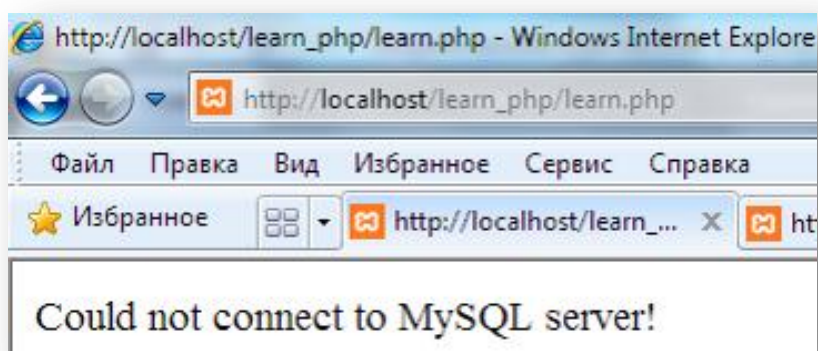
mysql_connect амали куйидаги буйрук сатрига тенг кучли

```
shell>mysql -u ilyosbey -p135
```

Агар дастур турли хостларда бир нечта MySQL тизимлари билан боғланишни ўрнатса, керакли MySQL тизимига *mysql_pconnect()* функцияси қайтарган боғланиш идентификатордан фойдаланиланиб сўровларни йўллаш мумкин.

14.5 – мисол.

```
<?
$link1 = @mysql_pconnect("localhost", "ilyosbey", "135")
or die("localhost serveriga bog`lana olmadi!");
$link2 = @mysql_pconnect("www.someotherhost.com", "usr", "secret")
or die("Could not connect to MySQL server!");
?>
```

Натижа**14.10 - расм.** Маълумотлар базаси билан муваффақиятли боғланиш

\$link1 ва *\$link2* идентификаторлари маълумотлар базасига сўровлар билан охирги мурожаатни узатади.

Кўп фойдаланувчилар муҳотида тизим ресурсларини тежаш учун *mysql_pconnect()* ни *mysql_connect()* билан биргаликда ишлатиш тавсия этилади. *mysql_pconnect()* функциясининг параметр типлари ва қайтарадиган

кийматлари `mysql_connect()` функцияси билан мос тушади, биттагина фарқи `mysql_connect()` функцияси маълумотлар базасига *доимий* боғланиб туриш учун ишлатилади.

MySQL тизимида ишлаш учун маълумотлар базасини танлаш.

MySQL тизимида муваффақиятли боғланиб бўлгандан сўнг, тизимда мавжуд сўровлар олиб бориладиган маълумотлар базаси танланади. Бизнинг маълумотлар `uquv` маълумотлар базасида сақланмоқда. MySQL маълумотлар базасидан фойдаланиш учун аввал `uquv` маълумотлар базаси танланади(активлаштирилади). МБ ни танлаш учун буйруқлар сатрида *use* буйруғидан фойдаланган эдик(`mysql>use uquv;`):

PHP да бунинг учун `mysql_select_db` функцияси ишлатилади.

mysql_select_db() функцияси

Синтаксиси:

`int mysql_select_db (`
 string *маълумотлар_базаси_номи*
 [, *int* *боғланиш_идентификатори*])

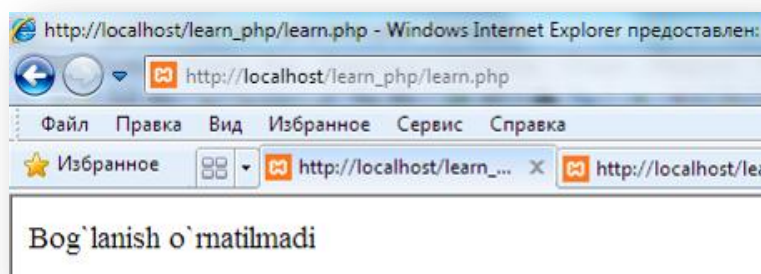
маълумотлар_базаси_номи параметрида танланиши лозим бўлган маълумотлар базаси кўрсатилади, боғланиш идентификатори параметрида эса бир нечта очик боғланишлардан *маълумотлар_базаси_номи* номли базанинг боғланиш идентификатори кўрсатилади.

Бу функция маълумотлар базасини муваффақиятли танлаганда TRUE ни, акс ҳолда FALSE ни қайтаради.

14.6 – мисол. `uquv` маълумотлар базасини танлаймиз:

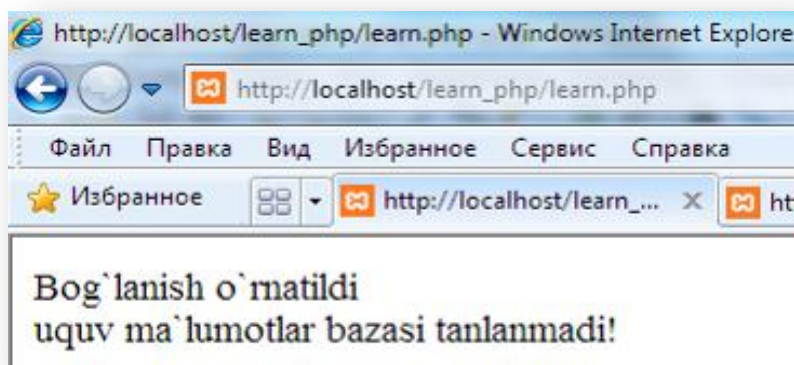
```
<?
@mysql_connect("localhost","ilyosbey","135")
or die ("Bog`lanish o`rnatilmadi");
echo "Bog`lanish o`rnatildi<br>";
@mysql_select_db("uquv","link1")
or die ("uquv ma`lumotlar bazasi tanlanmadi!");
?>
```

Натижа



14.11 - расм. *Серверга боғланиш*

Маълумотлар базасига боғлана олмаганда ёки маълумотлар базасини танламаганда браузерда қуйидагича хабарлар чиқазилади.



14.12 - расм. Маълумотлар базасига боғланиш

Агар дастурда фақатгина битта маълумотлар базаси танланаётган бўлса, унинг идентификаторини сақлаш шарт эмас. Лекин бир нечта маълумотлар базасини танлашда уларнинг қайтарадиган идентификаторлари сақланиши керак. Чунки, кейинчалик керакли базага сўровларни қайта ишлаш учун жўнатиш имконияти мавжуд бўлсин. Агар идентификатор кўрсатилмаган бўлса, охириги танланган маълумотлар базаси ишлатилади.

mysql_close() функцияси

mysql_close() функцияси шарт бўлмаган *боғланиш_идентификатори* параметри билан боғланишни ёпади. Агар параметр берилмаган бўлса, *mysql_close()* функцияси охириги очик боғланишни ёпади.

Синтаксиси:

int mysql_close ([int боғланиш_идентификатори])

mysql_pconnect() функцияси ёрдамида ўрнатилган очик боғланишни ёпиш шарт эмас.

14.3. Танланган базада сўровларни қайта ишлаш.

Танланган базада сўровларни қайта ишлаш қуйидаги тартибда амалга оширилади:

- а) маълумотларни қўшиш учун интерфейс яратиш;
- б) маълумотлар базасига маълумотларни ёзиш;
- с) маълумотлар базасида сақланаётган маълумотларни олиш.

а) Маълумотларни қўшиш учун интерфейс яратиш

Демак, бизда маълумотлар базаси ва унда қандайдир жадваллар мавжуд. Бу жадвалларга маълумотларни қўшиш интерфейсини яратиш учун уни структурасини (яъни майдонлар тўпламини) *html-формада* тасвирлаш керак.

Ушбу масалани қуйидагича қисм масалаларга бўлиб чиқамиз:

- *Жадвал майдонлари рўйхатини олиш;*
- *html-формада майдонларни тасвирлаш.*

Шундан сўнг, формага киритилган маълумотларни маълумотлар базасига ёзиш мумкин. Ушбу масалаларни тартиб билан кўриб чиқамиз.

• *Жадвал майдонлари рўйхатини олиш*

Жадвал майдонлари рўйхатини қандай олиш мумкин? Жуда оддий. PHP да бу ҳолат учун ўзининг буйруғи бор - `mysql_list_fields`.

mysql_list_fields() функцияси

Синтаксиси:

```
resource mysql_list_fields (
    string database_name,
    string table_name
    [,resource link_identifier])
```

Бу функция `database_name` маълумотлар базасида жойлашган `table_name` жадвалидаги майдонлар рўйхатини чиқазади. Бу функция ишининг натижаси - ресурс типигаги ўзгарувчи. Бу номи, типи ва байроғини ўзида жамлаган жадвал майдонлари ҳақидаги маълумотларни олиш учун ишлатиш мумкин бўлган мурожаат(ссылка).

mysql_field_name(), mysql_field_len(), mysql_field_type(), mysql_field_flags() функциялари

`mysql_field_name` функцияси сўров бажарилиши натижасида олинган майдон номини чиқазади. `mysql_field_len` функцияси майдон узунлигини чиқазади. `mysql_field_type` функцияси майдон типини, `mysql_field_flags` функцияси эса майдон байроқлари рўйхатини пробел ёрдамида ажратиб чиқазади. Майдон типи `int`, `real`, `string`, `blob` бўлиши мумкин. Байроқлар эса `not_null`, `primary_key`, `unique_key`, `blob`, `auto_increment` ва ҳ. бўлиши мумкин.

Бу буйруқларнинг барчасини **синтаксиси** бир хил:

```
string mysql_field_name (
    resource result, int field_offset)
string mysql_field_type (
    resource result, int field_offset)
string mysql_field_flags (
```

```
resource result, int field_offset)
string mysql_field_len (
resource result, int field_offset)
```

бу ерда *result* – сўров (масалан, *mysql_list_fields* ёки *mysql_query* (бу ҳақида кейинроқ тўхталамиз) функциялари ёрдамида жўнатилган сўров) натижаси идентификатори, *field_offset* эса – сўров натижасидаги устуннинг тартиб номери.

Умуман олганда *mysql_query* ёки *mysql_list_fields* типдаги функция натижаси сифатида жадвал, аниқроқ айтганда унга кўрсаткични чиқазади. Бу жадвалдан аниқ қийматларни олиш учун жадвални сатрма-сатр ўқий оладиган махсус функцияларни ишлатиш(задействоват) керак. Шундай функциялардан бири *mysql_field_name* ва б. Сўров бажарилиши натижасидаги жадвалдан барча сатрларни саралаш учун бу жадвалдаги сатрлар сонини билиш керак. *mysql_num_rows*(ресурс *result*) буйруғи - бир қанча *result* натижаларининг сатрлари сонини чиқазади.

Мисол сифатида ўқув маълумотлар базасини жадваллари билан ишлаймиз. Ўқув маълумотлар базасидаги *university* жадвалидаги устунлар рўйхатини олишни кўрамиз.

14.7 – листинг.

```
<?
$conn = mysql_connect("localhost","ilyosbey", "135")
or die("Bog`lanish o`rnatilmadi: ". mysql_error());
echo "Bog`lanish o`rnatildi";
mysql_select_db("uquv") or die("tanlanmadi");
$list_f = mysql_list_fields ("uquv","university",$conn);
$n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
    $type = mysql_field_type($list_f, $i);
    $name_f = mysql_field_name($list_f,$i);
    $len = mysql_field_len($list_f, $i);
    $flags_str = mysql_field_flags (
    $list_f, $i);
    echo "<br>Maydon nomi: ". $name_f;
    echo "<br>Maydon tipi: ". $type;
    echo "<br>Maydon uzunligi: ". $len;
    echo "<br>Maydon bayrog`i satri: ".
    $flags_str . "<br>";
}
?>
```

Натижа:

Bog'lanish o'rnatildi
Maydon nomi: univ_id
Maydon tipi: int
Maydon uzunligi: 6
Maydon bayrog'i satri: not_null primary_key unsigned
Maydon nomi: univ_name
Maydon tipi: string
Maydon uzunligi: 100
Maydon bayrog'i satri: not_null
Maydon nomi: rating
Maydon tipi: int
Maydon uzunligi: 6
Maydon bayrog'i satri: not_null
Maydon nomi: city
Maydon tipi: string
Maydon uzunligi: 200
Maydon bayrog'i satri: not_null

14.13 - расм. subject жадвалининг майдонлари

- **html-формада майдонлар рўйхатини акс эттириш**

Юқоридаги мисолга озгина ўзгартириш киритамиз. Майдон маълумотларини оддий ҳолда чиқазмаймиз, балки уларни *html-форманинг* мос элементларида акс эттирамиз. Бунинг учун, BLOB типдаги элементни `textarea` га ўтказамиз, сон ва сатрларни матнли сатрларни киритиш `<input type=text>` элементида акс эттирамиз, автоинкремент меткаси бор элементни эса чиқазмаймиз, чунки бу элемент қиймати автоматик тарзда қўйилади.

Ушбу *html-форма* ёрдамида фойдаланувчи автоматик тарзда `university`(университет) ва `subject`(ўқув фани) жадвалларига маълумотларни киритиши мумкин.

Байроқлар рўйхатидан `auto_increment` байроғини ажратиб олиш керак. Бунинг учун `explode` функциясидан фойдаланамиз.

Синтаксиси:

`array explode(string ажратувчи, string camp [, int limit])`

бу функция `camp` сатрини *ажратувчи* ажратувчиси ёрдамида қисмларга ажратади ва сатрдан ҳосил бўлган массивни чиқазади.

Бизнинг мисолда ажратувчи сифатида пробелни (« »), ажратиш учун сатр сифатида эса – майдон байроғи сатрини оламиз.

Демак, `university` жадвалига маълумотлар киритиш учун форма яратамиз:

14.8.1 – листинг. `new_univ_subject.php` файли. **uquv** маълумотлар базасининг `university` жадвалига маълумотлар киритиш формаси

```
<?
$conn=mysql_connect("localhost","ilyosbey","135");
// boglanish o'rnatamiz
```

```

echo "<form method=post action=new_univ_subject.php>";
$j_names=array ("university", "subject");
$s = "Jadvalni tanlang: <select name=jadval >";
// jadvallar ro'yhatni yaratamiz
foreach ($j_names as $k => $name) {
    $s .= "<option value=$name>$name";
    // jadvallar ro'yhatga jadvallarni qo'shamiz
}
$s .= "</select><br>";
echo $s;
    // jadvallarga nechta ma'lumot kiritilishi kerak o'sha sonni ko'rsata
miz
    echo "Jadvallarga kiritilishi lozim bo'lgan satrlar soni: <input type=inp
ut name='soni' size='5'>";
    echo "<input type=submit name='tan' value='Jadvalni tanlang'>";
    echo "</form>";

    if (isset($jadval) and isset($soni)) {
        $database = "uquv";
        $table_name = $jadval;
        mysql_select_db($database); // ishlash uchun MB ni tanlaymiz
        $list_f = mysql_list_fields($database,$table_name);
        // bazadagi maydonlar ro'yhatini olamiz
        $n = mysql_num_fields($list_f); // so'rov natijasidagi satrlar soni
        // ( jadvalida jami nechta maydon bor)
        echo "<form method=post action=insert_univ_subject.php>";
        // malumotlarni kiritish uchun forma yaratamiz
        echo "&nbsp;<TABLE BORDER=0 CELLSPACING=0 width=50% ><tr>
<TD BGCOLOR='#005533' align=center><font color='#FFFFFF'>
<b> $table_name ga yangi satr qo'shish</b></font></td></tr><tr><td><
/tr></TABLE>";
        for($k=0; $k<$soni; $k++){
            {
                echo $k+1 .' - universitet';
                echo "<table border=0 CELLSPACING=1 cellpadding=0 width=50% >";
                // har bir maydon uchun uning nomi, tipi, uzunligi va bayrogini olamiz
                for($i=0;$i<$n; $i++){
                    $type = mysql_field_type($list_f, $i);
                    $name_f = mysql_field_name ($list_f,$i);
                    $len = mysql_field_len($list_f, $i);
                    $flags_str = mysql_field_flags ($list_f, $i);
                    $flags = explode(" ", $flags_str);
                    foreach ($flags as $f){
                        if ($f == 'auto_increment') $key = $name_f;
                        // avtoinkrement nomini eslab qolamiz
                    }
                    if ($key <> $name_f){
                        echo "<tr><td align=right bgcolor='#C2E3B6'><font size=2>
<b>&nbsp;<b> $name_f .</b></font></td>";
                        $q_value=$name_f."[".$k."]";
                        switch ($type){
                            case "string":
                                $w = $len/5;
                                echo "<td> <input type=text name=\"\$q_value\" size = $w ></td>";
                                break;
                            case "int":
                                $w = $len/4;
                                echo "<td><input type=text name=\"\$q_value\" size = $w ></td>";
                                break;
                            case "blob":
                                echo "<td><textarea rows=6 cols=60 name=\"\$q_value\" ></textarea></td>";
                                break;
                            case "date":
                                $w = $len;

```

```

    echo "<td><input type=text name=\"\$q_value\" size = \$w >(YYYY-OO-
KK)</td>";
    break;
  }
}
echo "</tr>";
}
echo "</table>";
}
echo '<input type=hidden name="jadval" value ='.\$table_name.' >
<input type=hidden name="soni" value = '.\$soni.' >';
echo "<input type=submit name='add' value='Qo`shish'>";
echo "</form>";
}
?>

```

http://localhost/learn_php/new_univ_subject.php - Windows Internet Explorer предоставлен: Mail.Ru

http://localhost/learn_php/new_univ_subject.php

Файл Правка Вид Избранное Сервис Справка

Избранное localhost / localhost / uqu... http://localhost/learn_php... http://localhost/learn_...

Jadvalni tanlang: university

Jadvallarga kiritilishi lozim bo`lgan satrlar soni: Jadvalni tanlang

university ga yangi satr qo`shish

1 - universitet

univ_id

univ_name

rating

city

2 - universitet

univ_id

univ_name

rating

city

3 - universitet

univ_id

univ_name

rating

city

4 - universitet

univ_id

univ_name

rating

city

5 - universitet

univ_id

univ_name

rating

city

6 - universitet

univ_id

univ_name

rating

city

7 - universitet

univ_id

univ_name

rating

city

Qo`shish

14.14 - расм. *university* жадвалига маълумотлар киритиш формаси б) Маълумотлар базасига маълумотларни ёзиш

Маълумотларни маълумотлар базасига киритиш учун форма яратилди. Энди ушбу формадаги маълумотлар маълумотлар базасига жўнатилади. SQL

тилининг INSERT буйруғидан фойдаланиб, МБ даги жадвалга формада олинган маълумотларни ёзамиз. Масалан:

```
mysql> INSERT INTO Student
SET name='Islomov';
```

Савол туғилади, бундай буйруқни(ёки SQL нинг ихтиёрий бошқа буйруғини) PHP скриптида қандай қўллаш мумкин. PHP да сўровларни маълумотлар базасига жўнатиш учун *mysql_query()* функциясидан фойдаланилади.

***mysql_query()* функцияси**

Синтаксиси:

*resource mysql_query (string сўров
[,resource боғланиш_идентификатори])*

Бу ерда, *сўров* параметри SQL тилидаги сўров матнларидан ташкил топган. *mysql_query()* функцияси шарт бўлмаган параметр *боғланиш_идентификатори* ёрдамида кўрсатилган *MySQL* тизимининг актив маълумотлар базасига ёки бу параметр тушириб қолдирилган ҳолда, охириги очик боғланишга эга маълумотлар базасига SQL-сўровларни жўнатади. Агар очик боғланиш мавжуд бўлмаса, функция МББТ га боғланишга уриниб кўради. *mysql_query()* функцияси ишининг натижаси фақатгина битта ифодадан(TRUE ёки FALSE дан) иборат бўлади.

Изох: сўров сатри нуқта вергул билан тугалланмаслиги керак.

mysql_query() функциясига мурожаат этганда, сўров типига қараб функция турли натижаларни беради: Фақатгина SELECT, SHOW, EXPLAIN, DESCRIBE сўровлари учун, агар сўров бажарилмаса *mysql_query()* кўрсаткичи сўров натижаси сифатида FALSE ни қайтаради. Агар фойдаланувчининг сўровда кўрсатилган жадвал билан ишлаши учун ҳуқуқи етарли бўлмаса, *mysql_query()* бу ҳолни ҳам хато ҳисоблаб, FALSE ни қайтаради.

Бошқа ҳолларда агар сўров муваффақиятли бажарилса, *mysql_query()* TRUE ни қайтаради. FALSE га тенг бўлмаган қиймат сўров муваффақиятли бажарилганлигини билдиради. SELECT буйруғи муваффақиятли бажарилганда, функция натижа идентификаторини қайтаради ва ушбу идентификатор охириги таҳрирлаш ва сўров натижаларини тасвирлаш учун *mysql_result()* функциясига узатилади.

Демак, энди биз маълумотлар базасига сатрларни қўшиш учун қандай қилиб сўров жўнатишни биламиз. Эсга оламиз, олдинги мисолда форма элементларини жадвал майдонлари номлари билан атадик. Шунинг учун *insert_univ_subject.php* скриптида форма маълумотларини *\$_POST['майдон_номи']* ўзгарувчиси сифатида қайта ишлаш мумкин.

14.8.2 - листинг. *insert_univ_subject.php* файли. *uquv* маълумотлар базасининг *university* жадвалига янги талаба маълумотларини ёзиш.

```
<?
$conn=mysql_connect("localhost","ilyosbey", "135");// bog`lanishni o`rnatamiz
$dbase = "uquv";
$table_name = $jadval;
$soni= $soni;
```

```

mysql_select_db($database); // MB ni tanlaymiz
$list_f = mysql_list_fields($database,$table_name);
// bazadagi jadvalning maydonlar ro'yhatini olamiz
$n = mysql_num_fields($list_f); // oldingi so'rov natijasidagi so'rovlar soni
// jadvalning barcha maydonlari uchun bitta so'rov tuzamiz
for($k=0; $k<$soni; $k++)
{
    $sql[$k] = "INSERT INTO $table_name SET ";
    for($i=0;$i<$n; $i++){
        $name_f = mysql_field_name ($list_f,$i);
        // maydon nomini olamiz
        $flags_str = mysql_field_flags ($list_f, $i);
        $flags = explode(" ", $flags_str);
        foreach ($flags as $f){
            if ($f == 'auto_increment') $key = $name_f;
            // avtoinkrement nomini eslab qolamiz
        }
        if ($key <> $name_f){
            $z=$_POST[$name_f];
            $kl=0;
            //massivdagi qiymatlarni olib so'rov tuzamiz
            foreach ($z as $qiymatlar)
            {
                if ($kl==$k) $svalue=$qiymatlar;
                $kl++;
            }
            $j = $i + 1;
            $sql[$k] = $sql[$k] . ' ' . $name_f . "=" . $svalue . " ";
            // $sql satriga nomi-qiymati juftligini yozamiz
            if ($j <> $n) $sql[$k] = $sql[$k] . ", ";
            // agar maydon ro'yhatda oxirgi bo'lmasa vergul qo'yamiz
        }
    }
    $sql[$k]=$sql[$k];
    $result[$k] = mysql_query($sql[$k],$conn); // so'rovnı jo'natamiz
    // so'rov muvaffaqiyatli bajarilganligi haqida xabar chiqazamiz
    if (!$result[$k]) echo " ($table_name)qo'shilmadi <br> ";
    else echo "so'rov muvaffaqiyatli bajarildi!<br>";
};
?>

```

Худди шу каби, subject(ўқув фани) жадвалига ҳам маълумотларни киритамиз. univercity(университет) ва subject(ўқув фани) жадвалига маълумотлар киритилди. Энди student(талаба), lecturer(ўқитувчи), exam_marks(имтиҳон баҳоси), subj_lect(ўқитувчининг ўқув дисциплинаси) жадвалларига маълумотлар киритиш учун алоҳида-алоҳида форма яратамиз. Чунки бу жадвалларда бошқа жадваллардан олинadиган қийматлар мавжуд.

14.9.1 – листинг. new_talaba.php файли. uquv маълумотлар базасининг student жадвалига маълумотлар киритиш формаси

```

<?
$conn=mysql_connect("localhost","ilyosbey", "135");// bog'lanishni o'rnatamiz
$database = "uquv";
$table_name = $jadval;
$soni= $soni;
mysql_select_db($database); // MB ni tanlaymiz
?>
<html>
<head>
<title>Talabalar portalining O'quv MB siga Talabalar haqidagi ma'lumotlarni k
iritish</title>
</head>

```

```

<body>
<h3>O`quv MB siga Talabalar haqidagi ma`lumotlarni kiritish</h3>
<form action="insert_talaba.php" method="post">
<table border=0>
<tr>
<td bgcolor='#C2E3B6'>Talabaning identifikator nomeri:</td>
<td><input type="text" name="stud_id" maxlength=13 size=13></td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Familiyasi:</td>
<td><input type="text" name="surname" maxlength=30 size=30></td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Ismi:</td>
<td><input type="text" name="name" maxlength=30 size=30></td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Stipendiyasi:</td>
<td><input type="text" name="stipend" maxlength=30 size=30></td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Kursi:</td>
<td><select name="jadval">
<option value="1">Birinchi kurs</option>
<option value="2">Ikkinchi kurs</option>
<option value="3">Uchinchi kurs</option>
<option value="4">To`rtinchi kurs</option>
</select>
</td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Manzili:</td>
<td><input type="text" name="city" maxlength=30 size=30></td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Tug`ilgan sanasi:</td>
<td><input type="text" name="birthday" maxlength=30 size=30>(YYYY-OO-KK)</td>
</tr>
<tr>
<td bgcolor='#C2E3B6'>Университети:</td>
<td><?
echo "<select name=univ_id>";
// universitetlar ro`yhatni yaratamiz
$univ=mysql_query("select univ_id, univ_name from univcity");
$univ_cnt=mysql_num_rows($univ);
for($i=0;$i<$univ_cnt;$i++)
{
$univ_result=mysql_fetch_array($univ);
echo "<option value=\".$univ_result[\"univ_id\"]\">\".$univ_result[\"univ_name
\">";
}
echo "</select><br>";
?>
</td>
</tr>
<tr>
<td colspan=2><input type="submit" value="Kiriting"></td>
</tr>
</table>
</form>
</body>
</html>

```

14.15 - расм. *Student* жадвалига талабалар ҳақидаги маълумотларни киритиш формаси.

14.15 - расмдаги формада киритилган маълумотларни маълумотлар базасига ёзиш **14.9.2**-листингда кўрилади.

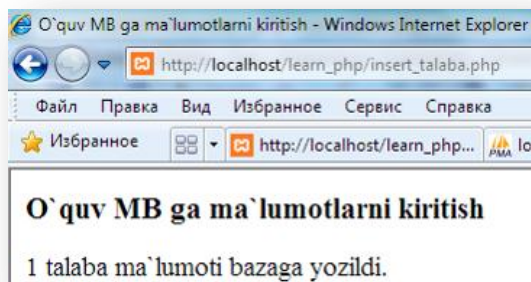
14.9.2 - листинг. **insert_talaba.php** файли. **uquv** маълумотлар базасининг **student** жадвалига янги талаба маълумотларини ёзиш

```
<html>
<head>
<title>O'quv MB ga ma'lumotlarni kiritish</title>
</head>
<body>
<h3>O'quv MB ga ma'lumotlarni kiritish</h3>
<?

if (!$stud_id || !$name || !$surname || !$kurs || !$univ_id)
{
echo "Ko'rsatilgan maydonlar to'ldirilmagan. To'ldirilishi zarur
bo'lgan maydonlarga ma'lumotlarni kiritish va qayta urinib ko'ring." ;
exit;
}
@ $db = mysql_pconnect("localhost", "ilyosbey", "135");
if (!$db)
{
echo "Ma'lumotlar bazasiga bog'lana olmadi . " ;
exit;
}
mysql_select_db( "uquv" ) ;
$query = "insert into student values
('".$stud_id."', '".$surname."', '".$name."', '".$stipend."', '".$kurs."', '".$sci
ty."', '".$birthday."', '".$univ_id."')";
$result = mysql_query($query);
if ($result)
echo mysql_affected_rows( ) . " talaba ma'lumoti bazaga yozildi.";
?>
```



```
</body>
</html>
```



14.16 – расм. *uquv* маълумотлар базасининг *student* жадвалига янги талаба маълумотларини ёзиш.

Худди шу каби, бошқа жадвалларга ҳам маълумотлар киритиш интерфейсини яратиш лозим.

Демак, маълумотларни киритиш масаласини Web-интерфейс ёрдамида ҳал этдик, лекин масаланинг бундай ҳал этилиши ҳамма вақт ҳам самара бермайди. Масалан, баъзи жадвалларда майдон қийматларини бошқа бир жадваллардан олишга тўғри келади ёки маълумотларни бу тарзда киритишда жадвалга бўш сатрлар ҳам кириши мумкин. Шунинг учун маълумотларни киритиш масаласини ечаётган вақтда масаланинг барча тарафларига эътибор қаратиш лозим. Масалани тўғри лойиҳалаш ва уни ҳал этиш йўллари топишни ўқувчига мустақил равишда қолдирамиз. Биз бошқа бир масалани-MySQL МББТ ида сақланаётган маълумотлар базасидаги маълумотларни чиқазишни кўриб ўтамиз.

с) *MySQL* да сақланаётган маълумотларни олиш

PHP технологияси ёрдамида маълумотларни браузерга чиқазиш учун, базадан олинган маълумотлар ўзгарувчига тенгланади. Сўнг ушбу ўзгарувчи браузерга чиқазилади.

MySQL да маълумотларни жадвалдан олишни амалга ошириш учун SQL тилининг `SELECT` буйруғидан фойдаланилади:

```
mysql> SELECT * FROM student;
```

Олдинги маърузада айтганимиздек, `mysql_query()` функцияси маълумотлар базасига сўровлар(маълумотларни жўнатиш, олиш, ўчириш, ўзгартириш каби) билан мурожаат этиш интерфейсини таъминлайди. Шунинг учун юқоридаги каби сўровни ҳам серверга `mysql_query()` функцияси ёрдамида жўнатиш мумкин. Энди сўровларда барча майдонларни танлаш, натижани эса браузерда чиқазиш талаб этилади. Бу сўров натижаси сифатида – жадвалнинг барча қийматлари, аниқроғи, бу жадвалга кўрсаткич чиқазилади.

Биз юқорида `mysql_field_name()` функцияси ёрдамида сўров натижасидан майдон номини олишни кўриб ўтган эдик. Энди бизга майдон номи эмас, балки майдон қийматини олиш керак бўлади. PHP нинг бундай

функциялари мавжуд, улардан энг кўп ишлатиладиганлари - *mysql_result()* ва *mysql_fetch_array()*.

***mysql_result()* функцияси**

mysql_result() функцияси *mysql_query()* функцияси билан биргаликда маълумотлар тўпламини олиш учун ишлатилади.

Синтаксиси:

mysql_result (*resource* натижа_идентификатори,
бу тун *camp*(ёки *ёзув*) [,майдон])

mysql_result() сўров натижаси сифатида битта ячейка қийматини чиқазади.

натижа_идентификатори параметрида *mysql_query()* функцияси қайтарган идентификатор - жадвалга кўрсаткич кўрсатилади. *camp*(ёки *ёзув*) параметрида *натижа_идентификатори* параметрида аниқланган маълумотлар тўпламидаги сатрларнинг қайси бири чиқазилиши лозимлиги кўрсатилади.

Шарт бўлмаган параметр *майдон* сифатида:

- жадвалдаги аралаш майдонлар;
- майдон номи;
- майдон номи *майдон_номи_жадвал_номи* форматида берилиши мумкин.

Агар сўровларни қайта ишлашда жуда кўп хотира сарфланаётган бўлса ва хотира етишмовчилиги юзага келса, у ҳолда PHP нинг стандарт функцияси *mysql_free_result* га мурожаат этилади. Унга мурожаат этишда *mysql_query()* қайтарган *натижа_идентификатори* узатилади. *mysql_free_result()* функцияси берилган сўров билан боғлиқ барча хотирани бўшатади.

Синтаксиси:

int mysql_free_result(int result);

Кўпгина ҳолларда SQL сўровида INSERT, UPDATE, REPLACE ёки DELETE буйруқлари иштирок этган вақтда сўров натижасидаги сатрлар сонини билиш талаб этилади. Бу масалани *mysql_affected_rows()* функцияси ҳал этади.

***mysql_affected_rows()* функцияси**

mysql_affected_rows() функцияси сўров натижасидаги сатрлар сонини аниқлаш учун ишлатилади. *mysql_affected_rows()* функцияси SELECT буйруғи иштирок этган сўровларда ишламайди. DELETE буйруғини WHERE сиз бажаришда *mysql_affected_rows()* функцияси ҳар доим 0 ни чиқазади.

Синтаксиси:

int mysql_affected_rows ([*int* боғланиш_идентификатори])

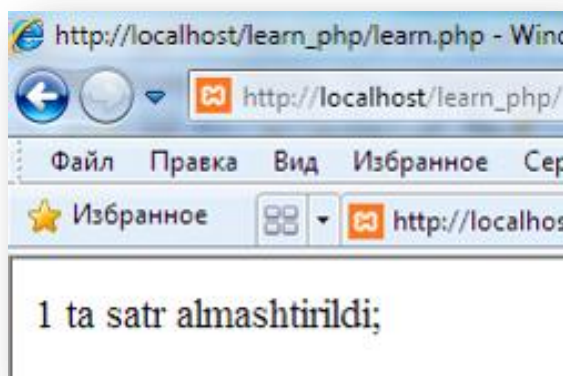
боғланиш_идентификатори параметри шарт бўлмаган параметр. Агар у кўрсатилмаса, *mysql_affected_rows()* охирги очиқ боғланишни ишлатишга уриниб кўради.

14.10 – мисол. *mysql_affected_rows()* функциясининг ишлатилиши

```
<?
// serverga bog`lanish va ma`lumotlar bazasini tanlash
```

```
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog`lan
ib bo`lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo`lmadi!");
// So`rov yaratish
$query = "UPDATE student SET surname = \"Fozilov\" WHERE stud_id = '3'";
$result = mysql_query($query);
print mysql_affected_rows(). " ta satr almashtirildi;";
mysql_close();
?>
```

Натижа



14.17 - расм. *mysql_affected_rows()* функциясининг ишлатилиши

mysql_num_rows() функцияси

SELECT буйруғи ишлатилганда чиқадиган сатрлар сонини аниқлаш учун *mysql_num_rows()* функцияси ишлатилади.

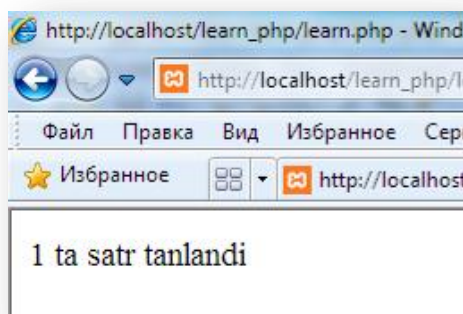
Синтаксиси:

int mysql_num_rows(int натижа)

14.11 - мисол. *mysql_num_rows()* функциясининг ишлатилиши

```
<?
// serverga bog`lanish va ma`lumotlar bazasini tanlash
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog`lan
ib bo`lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo`lmadi!");
// nomi 'p' bilan boshlanuvchi barcha fanlarni tanlash
$query = "SELECT name FROM student WHERE name LIKE \"%f%\"";
// so`rovni bajarish
$result = mysql_query($query);
print mysql_num_rows($result). " ta satr tanlandi";
mysql_close();
?>
```

Натижа



14.18 - расм. *mysql_num_rows()* функцияси ёрдамида МБ жадвалининг сатрлар сонини аниқлаш

14.12 - мисол. Subject жадвалидаги маълумотларни браузерда жадвал шаклида чиқазамиз:

14.12.1 - листинг.

```
<? // serverga bog'lanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog'lan
ib bo'lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");
// subject jadvalidagi barcha yozuvlarni olish
$query = "SELECT * FROM subject";
$result = mysql_query($query);
$x = 0;
print "<table>\n";
print "<tr>\n<th>Subj_id</th><th>Subj_name</th><th>Hour</th><th>Semestr</th>\n
</tr>\n";
while ($x < mysql_num_rows($result)) :
    $id = mysql_result($result, $x, 'subj_id');
    $name = mysql_result($result, $x, 'subj_name');
    $hou = mysql_result($result, $x, 'hour');
    $sem = mysql_result($result, $x, 'semestr');
    print "<tr>\n";
    print "<td>$id</td>\n<td>$name</td>\n<td>$hou</td>\n<td>$sem</td>\n ";
    print "</tr>\n";
    $x++;
endwhile;
print "</table>";
mysql_close();
?>
```

Натижа

Subj_id	Subj_name	Hour	Semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5

14.19 - расм. *mysql_result()* функцияси ёрдамида маълумотларни олиш

Кўпгина сўров натижалари билан ишлашда натижанинг бутун бир сатрини бир вақтнинг ўзида қайта ишлайдиган функцияларга амал қилишга тўғри келади (масалан, `mysql_fetch_row()`, `mysql_fetch_array()` ва х.).

Алоҳида устунларни олиш учун кўп маротаба `mysql_result()` функциясига мурожаат этгандан кўра барча устун ёзувларини индексли (0 дан бошланувчи индексли) массив элементи қийматларига тенглаштириш анчагина қулайликка олиб келади. Масала `mysql_fetch_row()` функция ёрдамида ҳал этилади.

mysql_fetch_row() функцияси

`mysql_fetch_row()` функцияси сўров натижасидаги маълумотлар тўпламини олиш учун ишлатилади.

Синтаксиси:

array ***mysql_fetch_row*** (*int* натижа_идентификатори)

натижа_идентификатори параметрида ***mysql_query()*** функцияси қайтарган қиймат берилади.

14.12.2 - листинг. `mysql_fetch_row()` функция ёрдамида маълумотларни олиш

```
<?
// serverga bog'lanish va ma'lumotlar bazasini tanlash
@mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog'lan
ib bo'lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo'lmadi!");
$query = "SELECT * FROM subject";
$result =mysql_query($query);
print "<table>\n";
print "<tr>\n<th>subj_id</th><th>subj_name</th><th>hour</th><th>semestr</th>\n</tr>\n";
while ($row = mysql_fetch_row($result)) :
print "<tr>\n";
print "<td>".$row["0"]."</td>\n<td>".$row["1"]."
</td>\n<td>".$row["2"]."</td>\n<td>".$row["3"]."</td>\n";
print "</tr>\n";
endwhile;
print "</table>";
mysql_close();
?>
```

Натижа

subj_id	subj_name	hour	semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5

14.20 - расм. `mysql_fetch_row()` функция ёрдамида маълумотларни олиш `mysql_fetch_array()` функцияси `mysql_fetch_row()` функциясига ўхшаш, бироқ битта фарқли тарафи одатга кўра майдон сатрлари қийматлари

ассоциатив массив сифатида сақланади. Индекслаш (ассоциатив, сонли ёки комбинацияланган) типини танлаш мумкин. Бу функция сўров натижалари сатрини қайта ишлайди ва қайта ишланган сўров натижалари сатрини массив(ассоциатив, сонли ёки иккаласи) сифатида чиқазади ёки агар сатрлар бўлмаса, FALSE ни қайтаради. *mysql_fetch_array()* – бу *mysql_fetch_row()* функциясининг кенгайтирилган версияси. *mysql_fetch_array()* функцияси *mysql_fetch_row()* га нисбатан секин ишламаслигини сезиш мумкин ва маълумотларга бирмунча қулай боғланиш имконини яратади.

Синтаксиси:

array mysql_fetch_array (int натижа_идентификатори
[, индекслаш_типи])

натижа_идентификатори параметрида *mysql_query()* функцияси қайтарган қиймат берилади. Шарт бўлмаган *индекслаш_типи* параметри ҳам константа бўлиши мумкин ва қуйидаги қийматлардан бирини қабул қилиши мумкин:

- **MYSQL_ASSOC** –*mysql_fetch_array()* функцияси ассоциатив массивни чиқазади. Агар параметр кўрсатилмаса, бу қиймат одатга кўра ишлатилади;
- **MYSQL_NUM** –*mysql_fetch_array()* функцияси сонли индексли массивни чиқазади;
- **MYSQL_BOTH** – олинадиган массив ҳам сонли индексли, ҳам ассоциатив индексли бўлиши мумкин.

Бу имконият PHP 3.0.7 версиясидан бошлаб қўшилган.

14.12.3 - листинг. *mysql_fetch_array()* функция ёрдамида маълумотларни олиш

```
<? @mysql_connect("localhost", "ilyosbey", "135") or die("MySQL serveriga bog`lanib bo`lmadi!");
@mysql_select_db("uquv") or die("uquv bazasini tanlab bo`lmadi!");
$query = "SELECT * FROM subject";
$result = mysql_query($query);
print "<table>\n";
print "<tr>\n<th>subj_id</th><th>subj_name</th>\n";
print "<th>hour</th><th>semester</th> \n</tr>\n";
while ($row = mysql_fetch_array($result)) {
print "<tr>\n";
print "<td>".$row["subj_id"]."</td>\n<td>".$row["subj_name"]."</td>\n";
print "<td>".$row["hour"]."</td>\n<td>".$row["semester"]."</td>\n";
print "</tr>\n";
}
print "</table>";
mysql_close();
?>
```

Натижа

subj_id	subj_name	hour	semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5

14.21 - расм. *mysql_fetch_array()* функция ёрдамида маълумотларни олиш

Изоҳ: бу функция ёрдамида чиқазиладиган майдонлар номи регистрга аҳамиятли.

14.12 – мисолни браузерда бошқа кўринишда тасвирлаш.

14.12.4 - листинг. 1 усул.

```
<?
$conn=mysql_connect("localhost","ilyosbey", "135");
$dbase = "uquv";
$table_name = "subject";
mysql_select_db($dbase);
$list_f = mysql_list_fields($dbase,$table_name);
$num = mysql_num_fields($list_f);
// $names massivida maydon nomlarini saqlaymiz
for($j=0;$j<$num; $j++){
    $names[] = mysql_field_name ($list_f,$j);
}
$sql = "SELECT * FROM $table_name"; // SQL so`rovini yaratamiz
$q = mysql_query($sql,$conn) or die(); // serverga so`rovni jo`natamiz
$num = mysql_num_rows($q); // natija satrlari sonini olamiz
// HTML-jadvallni yozamiz
echo "&nbsp;<TABLE BORDER=0 CELLSPACING=0 width=90%
align=center><tr><TD BGCOLOR='#005533' align=center>
<font color='#FFFFFF'><b>$table_name</b></font></td>
</tr></TABLE>";
echo "<table cellpadding=0 cellspacing=1 border=1
width=90% align=center>";
// omaydon nomlarini chiqazamiz
echo "<tr>";
foreach ($names as $val){
    echo "<th ALIGN=CENTER BGCOLOR='#C2E3B6'>
<font size=2>$val</font></th>";
}
// maydon qiymatlarini chiqazamiz
echo "</tr>";
for($i=0;$i<$num; $i++){ // tanlash sorovi natijasidagi barcha satrlarni navbat
    vilan olamiz
    echo "<tr>";
    foreach ($names as $k => $val) { // barcha maydon nomlarini olamiz
        $value = mysql_result($q,$i,$val); // maydon qiymatlarini olamiz
        echo "<td><font size=2>&nbsp;$value</font></td>";
        // maydon qiymatlarini chiqazamiz
    }
    echo "</tr>";
}
```



```
echo "</table>";
?>
```

Худди шуни *mysql_fetch_array()* ёрдамида амалга оширамиз:

14.12.5 - листинг. 2 усул.

```
<?
$conn=mysql_connect("localhost","ilyosbey", "135");
$dbase = "uquv";
$table_name = "subject";
mysql_select_db($dbase);
$list_f = mysql_list_fields($dbase,$table_name);
$n1 = mysql_num_fields($list_f);
// $names массивida maydon nomlarini saqlaymiz
for($j=0;$j<$n1; $j++){
    $names[] = mysql_field_name ($list_f,$j);
}
$sql = "SELECT * FROM $table_name"; // SQL so'rovini yaratamiz
$q = mysql_query($sql,$conn) or die(); // serverga so'rovni jo'natamiz
$n = mysql_num_rows($q); // natija satrlari sonini olamiz
// HTML-jadvallni yozamiz
echo "&nbsp;<TABLE BORDER=0 CELLSPACING=0 width=90%
align=center><tr><TD BGCOLOR='#005533' align=center>
<font color='#FFFFFF'><b>$table_name</b></font></td>
</tr></TABLE>";
echo "<table cellpadding=0 cellspacing=1 border=1
width=90% align=center>";
// maydon nomlarini chiqazamiz
echo "<tr>";
foreach ($names as $val){
    echo "<th ALIGN=CENTER BGCOLOR='#C2E3B6'>
<font size=2>$val</font></th>";
}
// maydon qiymatlarini chiqazamiz
echo "</tr>";
// yuqorigi qismi avvalgi misol bilan bir xil
for($i=0;$i<$n; $i++){
    // maydon qiymatlarini assotiativ massiv sifatida olamiz
    while($row = mysql_fetch_array($q, MYSQL_ASSOC)) {
        echo "<tr>";
        foreach ($names as $k => $val){
            echo "<td><font size=2>&nbsp;$row[$val]</font></td>";
            // maydon qiymatlarini chiqazamiz
        }
        echo "</tr>";
    }
}
echo "</table>";
?>
```

Натижа. Иккала усулда ҳам натижа бир хил чиқади

subject			
subj_id	subj_name	hour	semestr
10	Informatika	56	1
22	Fizika	34	1
43	Matematika	56	2
56	Tarix	34	4
94	Ingliz tili	56	3
73	Mehnat	34	5

14.22 - расм. Олинган маълумотларни браузерда тасвирлаш

Маълумотлар базасини яратиш ва ўчириш

PHP-саҳифасини ишлатиб MySQL да янги маълумотлар базасини яратиш учун **mysql_create_db()** функцияси ва маълумотлар базасини ўчириш учун— **mysql_drop_db()** функцияси қўлланилади.

Синтаксиси:

```
int mysql_create_db(string database, [int database_connection] );
```

```
int mysql_drop_db(string database, [int database_connection] ) ;
```

Бу иккала функцияда параметр сифатида маълумотлар базаси номи ва боғланиш идентификатори ишлатилади. Агар боғланиш идентификатори бўлмаса, охирги боғланилган маълумотлар базаси билан ишлайди. Бу функциялар маълумотлар базасини яратади ва ўчиради, функция иши муваффақиятли якунланса, true қийматини, акс ҳолда — false ни қайтаради.

3 ҚИСМ. ФОЙДАЛАНИШ УЧУН РУХСАТ

XV БОБ. Сессиялар билан ишлаш

Бу бобда тармоқдаги хавфсизликни таъминлаш ва бу учун сессия механизмларини ишлатилиши ҳақида сўз боради. Сессияни инициализациялаш, фойдаланувчи идентификаторини узатиш, сессия ўзгарувчиларини рўйхатдан ўтказиш, сессияни ўчириш кабилар кўриб ўтилади. Ҳамда сессияни `php.ini`, `httpd.conf`, `.htaccess` файлларида созлаш кўрилади. Боб сўнгида механизм ёрдамида фойдаланувчини рўйхатдан ўтказишга мисол келтирилади.

15.1. Фойдаланиш учун рухсат

"Фойдаланиш учун рухсат" нима? Бу тушунчани ҳаётий мисол ёрдамида тушунтиришга ҳаракат қиламиз. Тасаввур қилинг, кутубхонадан китоб олмоқчисиз. Лекин бу хизматдан фойдаланиш учун фақатгина кутубхона билети борларгагина рухсат берилган. Ушбу билет ёрдамида кутубхона ресурсларига "фойдаланиш учун рухсат" берилади дейиш мумкин. Кутубхоначи кутубхона билетини кўргандан сўнг, китобни ким олаётганлиги ҳақида билади ва зарур бўлган ҳолларда (масалан, китоб анчадан бери қайтарилмаган) бирор бир чора (китобхонга телефон қилади) кўради. Оддий ташриф буюрувчига нисбатан кутубхоначи кўпроқ ҳуқуққа эга: у аниқланган ташриф буюрувчига китобни бериши ёки бермаслиги мумкин, янги келган китобларни кўрсатиш учун қўйиши ва кам ўқиладиган китобларни архивга олиб қўйиши мумкин ва ҳ.

Ахборот технологияларида ҳам ҳамма мисоллар худди шундай. Тармоқда кўп сондаги ресурслар мавжуд, ҳатто "кутубхонадан" ҳам кўп. Уларнинг ҳар бирини ўзининг «кутубхоначи»си, ҳамда ресурс таркибига жавоб берувчи ва маълумотни фойдаланувчига тақдим этувчи инсон ва гуруҳлар мавжуд. Уларни администраторлар деб аталади. Администратор функцияси, қоидага кўра янги маълумотни қўшиш, мавжудларини таҳрирлаш ва ўчириш, фойдаланувчига маълумотларни тасвирлаш усуллари созлаш кабиларни ўз ичига олади. Фойдаланувчи (ресурсга оддий ташриф буюрувчи) функциясига эса маълумотларни кўриш ва қидириш қиради.

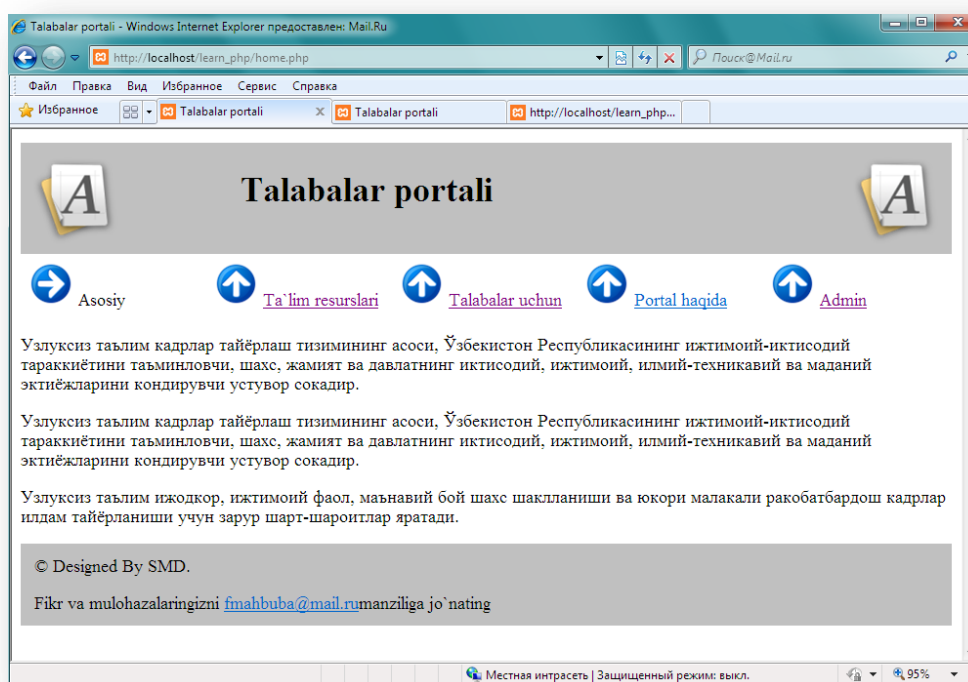
Фойдаланувчини администратордан қандай ажратиб олиш мумкин? Ҳақиқий кутубхонада бу равшан, лекин агар кутубхоначи ва фойдаланувчи ролини виртуал ҳақиқийликка кўчирилса, бу аниқлик йўқолади. Кутубхоначи фойдаланувчи каби кутубхона ресурсларига Internet орқали қира олади. HTTP протоколига мувофиқ барча мижозлар мутлақо тенг ҳуқуқли. Сайтга ким кирганлигини қандай билиш мумкин? Оддий фойдаланувчи (ташриф буюрувчи)ми ёки администратор (кутубхоначи)? Агар бу оддий фойдаланувчи бўлса, ташриф буюрувчига сайтнинг ёпиқ архивидан фойдаланишга рухсат бермасликни қандай таъминлаш мумкин? Шу ўринда савол туғилади, сўров жўнатган мижозни қандай идентификациялаш керак ва у сайтда ишлаётган вақтда у ҳақидаги маълумотларни қандай сақлаш мумкин?

Эсга биринчи келадиган, энг оддий варианты, - бу системадаги инсонни рўйхатдан ўтказиш ва унга кутубхона билетиға ўхшаш системанинг бошқарув(административ) қисмиға кириш учун логин ва пароль бериш керак. Бу маълумотлар компьютер серверида сақланади ва системаға киришдан олдин фойдаланувчи киритган логин ва паролни системада сақланаётгани билан мослиги текширилади. Ҳақиқатда, бу ерда ҳақиқий кутубхона билан таққослаганда вазият ўзгаради: кутубхона билети кутубхоначидан системанинг ёпиқ қисмиға кириш учун талаб этилади, китобхон эса сайтға бемалол кириши мумкин. Умуман олганда оддий ташриф буюрувчиларни ҳам рўйхатдан ўтказиш мумкин. У ҳолда барча рўйхатдан ўтган фойдаланувчиларни гуруҳларға уларни мос ҳуқуқларига кўра: кутубхоначилар (администраторлар) ва китобхонлар (оддий фойдаланувчилар) га ажратиш керак. Биз бу нозик масалани ичига кирмаймиз ва энг оддий сайтнинг баъзи бир саҳифаларига кириш учун талаб этилган логин ва паролни киритиш усулидан фойдаланамиз.

15.1 - мисол.

Бизда талабалар порталининг Admin бўлими мавжуд бўлиб, ушбу бўлим орқали порталға маълумотлар киритиш мумкин.

home.php файли. Талабалар порталининг Асосий саҳифаси



15.1 - расм. home.php файлининг браузердаги кўриниши

Администратор бўлимида фақатгина маълумотлар базасига маълумотларни киритиш ҳуқуқига эга бўлган шахсларгина ишлай олишади. Бу бўлимға кириш учун [Admin](#) менюсига мурожаат этилади.

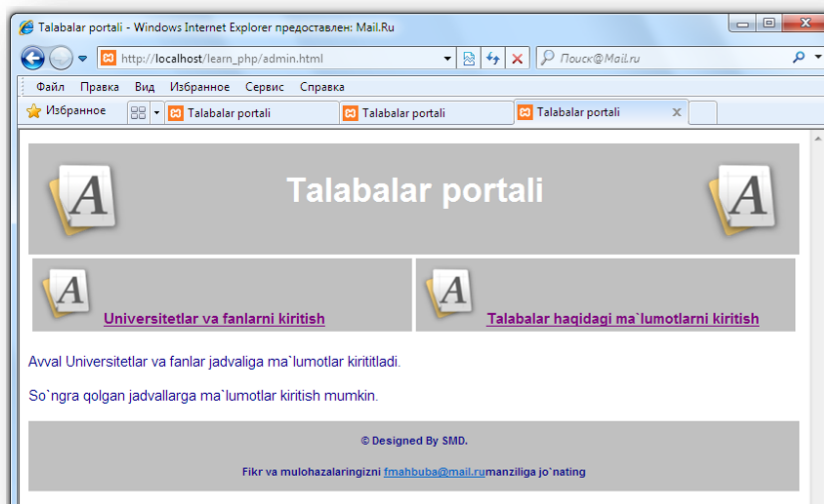
15.1a – листинг. admin.php файли. Талабалар порталининг Admin бўлими:

```
<html>
<head>
```

```

<title>Talabalar portali</title>
<style>
<!--
h1 {color: white; font-size :24p; text-align : center ; font-
family : arial , sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-
family : arial , sans-serif; font-weight : bold}
td {background: silver}
p {color:darkblue; font-size: 12pt; text-align: justify;font-
family: arial , sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-
family : arial , sans-serif ; font-weight : bold}
a: link , a : visited, a : active {color : white }
-->
</style>
</head>
<body>
<!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "logo.png "></td>
<td>
<h1>Talabalar portali</h1>
</td>
<td align = right><img src = "logo.png "></td>
</tr>
</table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
<tr >
<td width = "25%">
<img src = "s-
logo.png "> <span class=menu><a href="new_univ_subject.php">Universitetlar va
fanlarni kiritish</a></span></td>
<td width = "25%">
<img src = "s-
logo.png "> <span class=menu><a href="new_talaba.php">Talabalar haqidagi ma`l
umotlarni kiritish</a></span></td>
</tr>
</table>
<!-- page content -->
<p>Avval Universitetlar va fanlar jadvaliga ma`lumotlar kirititladi. </p>
<p>So`ngra qolgan jadvallarga ma`lumotlar kiritish mumkin.</p>
<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr>
<td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@mail.ru">f
mahbuba@mail.ru</a>manziliga jo`nating </p>
</td>
</tr>
</table>
</body>
</html>

```



15.2 - расм. *admin.php* файлининг браузердаги кўриниши

Агар бу файлларни шундайлигича қолдирилса, у ҳолда ихтиёрий фойдаланувчи "Admin" гипермуружаат(ссылка)ини босиб администратор саҳифасига кириши мумкин. Администратор саҳифасидан фақатгина администраторлар фойдалнишини таъминлаш учун бу саҳифага муружаат этилганда фойдаланувчи шахсини аниқловчи оралиқ скриптини кўшиш керак. Администратор саҳифасига муружаат этилганда *admin.html* файлига эмас, балки аввал *authorize.php* скриптига йўналтирилишини таъминлаш керак.

15.1b – листинг. *page.inc* файли

```
<?
class Page
{
//Page klassining atributlari
var $content ;
var $title = "Talabalar portali";
var $keywords = "Talabalar, o`quv jarayoni, maqolalar, darsliklar, baholar";
var $buttons = array( "Asosiy" => "home.php",
                    "Ta`lim resurslari" => "resurs.php",
                    "Talabalar uchun" => "talaba.php",
                    "Portal haqida" => "about.php",
                    "Admin" => "authorize.php "
                );
// Page klassining metodlari
... ..
```

Авторизация скрипти ўзида логин ва паролни киритувчи формани ва уларни тўғрилигини текширувчи кодни, агар текшириш муваффақиятли яқунланган бўлса, уни махфий саҳифага йўналтиришни, акс ҳолда хатолик ҳақидаги хабарни чиқазишни жамлаган бўлиши керак.

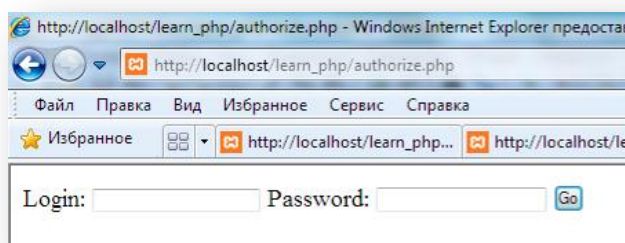
15.1c – листинг. *authorize.php*

```
<?
if (!isset($_GET['go'])) {
// foramadagi ma'lumotlar jo`natulganligini tekshiramiz
// avtorizatsiya uchun forma
```

```

// (login va parolni kiritish)
echo "<form>
Login: <input type=text name=login>
Password: <input type=password
name=passwd>
<input type=submit name=go value=Go>
</form>";
} else {
// agar forma to'ldirilgan bo'lsa,
// login parolni
// to'g'ri login va parol bilan solishtirish
if ($_GET['login']=="lola" &&
$_GET['passwd']=="123") {
Header("Location: admin.php");
// va maxfiy sahifaga yo'naltiramiz
} else echo "Noto'g'ri kiritildi,
qayta kiritib ko'ring<br>";
}
?>

```



15.4 - расм. *authorize.php* файлининг браузердаги кўриниши

Ҳаммаси жуда оддий. Лекин бизда махфий саҳифалар битта эмас, балки бир нечта. Ҳамда улар бир-бири билан кесишган жўнатма(ссылка)лар билан боғланган. У ҳолда сайтга ташриф буюрувчининг логин ва парolini ҳар доим эсда сақлаш зарурати туғилади. Бу масалани ҳал этиш учун, ҳар бир саҳифада логин ва парolni саҳифадан саҳифага узатувчи яширин параметрлар сифатида берилган формани ташкил этиш мумкин. Лекин бу усулда параметрларни олиш ва сохталаштириш мумкин, яъни бу усул хавфсиз эмас. Сайтга ташриф буюрувчиларнинг ишлаш жараёнида уларнинг маълумотларини сақлаш муаммоси ечими РНР да бир мунча қулай ва хавфсиз усул билан ҳал этилади – бу *сессия* механизми.

15.2. Сессия механизми

Сессия – ўзгарувчиларни яратиш ва ишлатиш имкониятини яратувчи механизм. Бу ўзгарувчилар фойдаланувчи сайт билан ишлаш вақти жараёнида ўз қийматларини сақлайдилар.

Бу ўзгарувчилар қийматлари ҳар бир фойдаланувчи учун турлича бўлиши мумкин ва ундан фойдаланувчи системадан чиқиб кетгунча сайтнинг ихтиёрий саҳифасида фойдаланиш мумкин. Шунинг учун, сайтга ҳар гал кирганда, фойдаланувчи ушбу сеанс давомида ёки сайт билан ишлаш сессиясида уни идентификациялаш имкониятини берувчи янги ўзгарувчи қийматини олади.

Фойдаланувчини идентификациялаш учун ҳар бир фойдаланувчига сессия идентификатори(SID, Session IDentifier) деб номланувчи уникал номер берилади. У PHP да фойдаланувчи сайтга кирган вақтнинг ўзида генерация қилинади, ҳамда ўзида 32 та белгидан иборат сатрни ифодалайди (масалан, ac4f4a45bdc893434c95dcaffb1c1811) ва фойдаланувчи сайтдан чиққан вақтда ўчирилади. Бу идентификатор серверга мижоз сўрови билан биргаликда узатилади ва сервер жавоби билан бирга ортга қайтарилади.

Сессия идентификаторини узатишни бир нечта йўллари мавжуд:

•***cookies ёрдамида.***

Cookies мижозни идентификациялашни бир қийматли методи сифатида махсус яратилган ва ўзида http протоколи кенгайтмасини ифодалайди. Бу ҳолда *сессия идентификатори* сўров жўнатаётган мижоз компьютерининг вақтинчалик файлида сақланади. Бу метод, шубҳасиз яхши, лекин кўпгина фойдаланувчилар хавфсизлик муаммоси туфайли ўзларини компьютерида *cookies* ёрдамини ўчириб қўйишади.

•***буйруқлар сатри параметри ёрдамида.***

Бу ҳолатда *сессия идентификатори* серверга узатилаётган барча сўровларни(URL) автоматик ўрнатади ва сервер томонида сақланади.

Масалан : <http://green.nsu.ru/test.php> манзили
<http://green.nsu.ru/test.php?PHPSESSID=ac4f4a45bdc893434c95dcaffb1c1811>
 манзилига айлантирилади

Идентификаторни узатишнинг бу усули агар сўров жўнатилаётган браузерда *cookies* ўчирилган бўлса автоматик тарзда ишлайди. У ҳар доим буйруқлар сатрида параметрларни узатишда етарли даражада ишончли бўлиши мумкин. Бошқа томондан эса *сессия* идентификаторини браузерда сақланган вариантидан фойдаланиш ёки сохталаштириш учун кўриш мумкин. Шундай бўлса ҳам, албатта бу муаммоларни ечиш мумкин. Масалан, ким 32 та турли белгидан иборат сатрни эслаб қолади? Агар сессиялар билан ишлашни тўғри ташкиллаштирилса(уларни ўчириш вақтида), у ҳолда хатто браузерда сақланаётган сессия номери ҳам ҳеч нарса бермаслиги мумкин.

15.3. Сессияни созлаш

Сессия билан ишлашдан аввал, уларни PHP интерпретатори ёрдамида қайта ишлашни, қандай қилиб тўғри созлашни аниқлаб олиш керак.

Одатда PHP га сессиялар билан ишлаш механизми махсус модуллар ёрдамида бириктирилган бўлади. Бу шуни англатадики, сессия билан ишни ташкил қилиш учун ҳеч қандай қўшимча элементларни ўрнатиш керак эмас. Лекин, ушбу модулнинг созловлари қандай эканлигини билиш сессия билан ишлашда вужудга келувчи хатоларни бартараф қилишда фойдалидир.

Сессиялар билан ишлаш учун PHP ни созлаш ҳам, `php.ini` файлига ёзиш орқали амалга оширилади. Бу файлга мурожаат этамиз.

Биз биламизки, *сессия идентификатори* сервер компьютерида ёки мижоз компьютерида ёки бошқа жойда сақланади.

`php.ini` даги *`session.save_path`* параметри, сессия маълумотлари сервернинг қаерида сақланиши кераклигини аниқлайди. У туфайли Windows-серверда бошқалардан кўра кўпроқ муаммолар пайдо бўлади, чунки одатда *`session.save_path`* қиймати `/tmp` га ўрнатилган бўлади. Ва агар сервернинг ўзак каталогида бундай папка бўлмаса, у ҳолда сессияни ишга туширишда хатолик юзага келади.

Сервер кўп микдордаги сессияларни бир вақтнинг ўзида қайта ишлаши мумкин ва уларнинг барча вақтинчалик файллари *`session.save_path`* параметрида берилган каталогда сақланади. Агар тизим катта ҳажмдаги папкалар билан яхши ишламаса, у ҳолда қисм каталогларни ишлатиш қулай ҳисобланади. Бунинг учун, папка номидан ташқари параметр қийматига, бу папкада(`N:/dir`) жойлашган қисм директорияларни ичма-ичлигини аниқловчи сон ҳам қўшилади. Модомики, нуқта вергул PHP ни созлаш файлида изоҳ белгиларидан бири экан, бу қийматни албатта қўштирноқ ичига олиш керак. Сессия маълумотларини сақлаш учун барча директория ва қисм директориялар мустақил равишда яратилади.

Масалан: `2;/Temp` шуни англатадики, *сессия* ўзгарувчилари `c:/Temp/0/a`, `c:/Temp/0/b` ва б. кўринишидаги папкада сақланади.

Мижоз томонида маълумотларни сақлаш *cookies* ёрдамида амалга оширилади. PHP ни *cookies* билан ишлашни *`session.use_cookies`*, *`session.cookie_lifetime`* параметрлари ёрдамида созланади.

`session.use_cookies` параметри сессия билан ишлашда *cookies* ишлатилаётганми ёки йўқлигини аниқлайди. Одатда бу опция уланган(ёқилган) бўлади (`"1"` қийматини қабул қилади).

`session.cookie_lifetime` параметрида *cookies* нинг ишлаш вақти давомийлиги секундларда берилади. Одатда бу `"0"`, ҳ. га тенг бўлади. *cookies* даги маълумотлар браузер ойнаси ёпилгунча сақланиб туради.

Бу параметрлардан ташқари, сессия номини аниқловчи *`session.name`*, сессияни автоматик тарзда ишга тушириш имкониятини берувчи *`session.auto_start`*, сессия маълумотлари кодировкаси усулларини берувчи *`session.serialize_handler`*, ва кэшда хужжатлар неча минутдан сўнг

истеъмолдан чиқишини аниқловчи *session.cache_expire* параметрларини кўрсатиш мумкин.

Одатда, сессия номи (*session.name*) PHPSESSID каби ўрнатилиб, *cookies* да сессия идентификаторида сақланаётган ўзгарувчи сифатида ишлатилади. Сессияни автоматик ишга тушириш ўчирилган бўлади, лекин *session.auto_start* ни қийматини "1" га тенглаб, уни юклаш мумкин. Сессия маълумотлари кодировкаси учун *php* ишлатилади. Кэшда сақланаётган маълумотларни истеъмолдан чиқиши, 180 минутдан сўнг амалга оширилади.

php.ini созлаш файлида амалга ошириш мумкин бўлган бир нечта созлашлар мавжуд. Лекин, РНР сессиялари билан ишлаш учун ушбу кўриб ўтган созлашларимиз етарли.

15.4. Сессиялар билан ишлаш

Сессия яратиш

Биринчидан, сессиялар билан ишлаш учун, яъни сессия механизмини ишга тушириш учун нима қилинади (агар улар сервер администратори томонидан созлаб бўлинган бўлса). Агар сервер созловида *session.auto_start* ўзгарувчисига "0" қиймати ўрнатилган бўлса (агар *session.auto_start=1*, у ҳолда сессия автоматик тарзда ишга тушади), у ҳолда сессия маълумоти ишлатилиши керак бўлган ихтиёрий скрипт қуйидаги буйруқ билан бошланади:

```
session_start();
```

Бундай буйруқни олгандан сўнг, сервер янги сессия яратади ёки сўров билан узатилган сессия идентификаторига асосланган жорий сессияни тиклайди. Бу қандай бажарилади? PHP интерпретатори сессия идентификатори(PHPSESSID)да сақланаётган ўзгарувчини аввал *cookies* дан, сўнг POST- ва GET-сўровлари ёрдамида узатилган ўзгарувчидан қидиради. Агар идентификатор топилса, у ҳолда фойдаланувчи идентификацияланган ҳисобланади, барча URL алмаштиришлар бажарилади ва *cookiesga* ёзиб қўйилади. Акс ҳолда фойдаланувчи янги ҳисобланади ва унинг учун янги уникал идентификатор генерация қилинади, сўнг URL алмаштиришлар бажарилади ва *cookies ga* ёзиб қўйилади.

Бирор бир маълумотни браузерга чиқазишда, сессия ўзгарувчиларини ишлатиш лозим бўлган барча скриптларда *session_start()* буйруғига мурожаат этади. Бу шу билан боғлиқки, *cookies* фақатгина маълумот экранга чиқазгунча ёзиб қўйилади.

Жорий сессия идентификаторини олиш учун *session_id()* функциясидан фойдаланилади.

Сессияни кўрсатиш учун *session_name([сессия_ному])* функцияси ёрдамида ном берилади. Буни сессияни инициализация қилгунча бажарилади. Жорий сессия номини шу функция ёрдамида, параметрсиз мурожаат этиб олиш мумкин: *session_name()*;

15.2 - мисол. Сессия яратиш.

home.php файлига ўзгартиришлар киритамиз. Ушбу саҳифада, сессия яратамиз ва у қандай идентификатор ва ном олишини кўрамиз.

```
<?
session_start();
// yangi sessiya yaratamiz yoki
// joriysini tiklaymiz
echo session_id();
// sessiya identifikatorini chiqazamiz
//
require("page.inc") ;
$homepage = new Page( );
$homepage -
> SetContent ("<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекист
он Республикасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-
техникавий ва маданий эктиёжларини кондирувчи устувор соқадир. </p>
```

```

<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Республикаси
нинг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-техникавий ва маданий эктиёжларини кондирувчи устувор соқадир.
</p>
<p>Узлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шаклланиши ва юк
ори малакали рақобатбардош кадрлар илдам тайёрланиши учун зарур шарт-
шароитлар яратади.</p>
");
$homepage -> Display ();
//
echo session_name();
// joriy sessiya nomini chiqazamiz
// Bu holda PHPSESSID
?>

```

Натижа:



15.5 - расм. *home.php* файлининг браузердаги кўриниши

Агар буларни ҳаммасини `authorize.php` файлида бажариб, администратор саҳифасига `home.php` орқали кирсак ва бу ойнани браузерда ёпмасак (ёпилса сессия идентификатори ўзгаради), ўзгарувчи(сессия `id` си ва унинг номи) чиқазадиган қийматлар бир хил бўлади.

Сессия ўзгарувчиларини рўйхатдан ўтказиш

Сессия жараёнида биз ўзимизнинг шахсий ўзгарувчиларимизни (масалан, логин ва паролни) узатишни ва сақлашни хоҳлаймиз. Бунга эришиш учун ўзимизнинг ўзгарувчиларни оддийгина рўйхатдан ўтказиш керак. Бу `session_register()` функцияси ёрдамида амалга оширилади.

Синтаксиси:

```

session_register(ўзгарувчи_номи1,
ўзгарувчи_номи2, ...);

```

Эътибор беринг, киймат эмас, балки *ўзгарувчи_номи* рўйхатга олинмокда. Ўзгарувчиларни сессия ишлатаётган ихтиёрий саҳифада бир марта рўйхатдан ўтказиш етарли. Ўзгарувчи номи *session_register()* функциясига \$ белгисисиз узатилади. Барча рўйхатдан ўтказилганлар шу ўринда ўзгарувчилар ҳам берилган сессия сайт билан ишлаш жараёнида глобал (ихтиёрий саҳифада фойдаланиш мумкин) сифатида шаклланади.

Ўзгарувчиларни уларни кийматларини оддийгина ассоциатив массив *\$_SESSION* га ёзиб ҳам рўйхатдан ўтказиш мумкин.

```
$_SESSION['ўзгарувчи_номи'] =  
'ўзгарувчи_қиймати';
```

Бу массивда барча рўйхатдан ўтган (глобал) сессия ўзгарувчилари сақланади.

Бундай ўзгарувчиларга доступ *\$_SESSION['ўзгарувчи_номи']* (ёки PHP 4.0.6 ва ундан аввалги версиялари учун *\$HTTP_SESSION_VARS['ўзгарувчи_номи']*) массиви ёрдамида амалга оширилади. Агар php ни созлашда *register_globals* опцияси ёқилган бўлса, у ҳолда сессия ўзгарувчиларига оддий ўзгарувчиларга мурожаат этган каби мурожаат этиш мумкин. **масалан:** *\$ўзгарувчи_номи*.

Агар *register_globals=off* (ўчирилган) бўлса, у ҳолда POST ёки GET методи билан узатилган ўзгарувчиларни рўйхатдан ўтказиш учун *session_register()* дан фойдаланиш мумкин эмас, чунки бу ишламайди. Умуман олганда, ўзгарувчиларни *\$_SESSION* ва *session_register()*лар ёрдамида рўйхатдан ўтказиш мумкин, лекин бу иккала методни бир вақтнинг ўзида ишлатиш тавсия этилмайди.

15.3 - мисол. Ўзгарувчиларни рўйхатдан ўтказиш.

Авторизация саҳифасида фойдаланувчи томонидан киритилган логин ва паролни рўйхатдан ўтказиш.

15.3а - листинг. authorize.php

```
<?  
session_start();  
// yangi sessiya yaratamiz yoki  
// joriysini tiklaymiz  
if (!isset($_GET['go'])) {  
    echo "<form>  
    Login: <input type=text name=login>  
    Password: <input type=password name=passwd>  
    <input type=submit name=go value=Go>  
    </form>";  
}  
else {  
    $_SESSION['login']=$_GET['login'];  
    // login o`zgaruvchisini ro`yhatdan o`tkazamiz  
    $_SESSION['passwd']=$_GET['passwd'];  
    // passwd o`zgaruvchisini ro`yhatdan o`tkazamiz  
    // endi login va parol - bu  
    //sessiya uchun global o`zgaruvchi  
    if ($_GET['login']=="lola" &&  
        $_GET['passwd']=="123") {  
        Header("Location: admin.php");  
        // admin.php  
        // sahifasiga yo`naltiramiz  
    }else echo "Noto`g`ri kiritildi,"
```



```

    Qayta kiriting!<br>";
}
print_r($_SESSION);
// sessiyaning barcha o'zgaruvchilarini chiqazish
?>

```

Энди биз `admin.php` саҳифасига ва сайтнинг ихтиёрий бошқа саҳифасига кира оламиз, `$_SESSION` массивида сақланаётган фойдаланувчи томонидан киритилган логин ва пароль билан ишлай оламиз. Агар администратор саҳифасини кодини куйидагича ўзгартирсак (эътибор беринг, биз уни `admin.php` да ўзгартирдик):

15.3b - листинг. `admin.php`

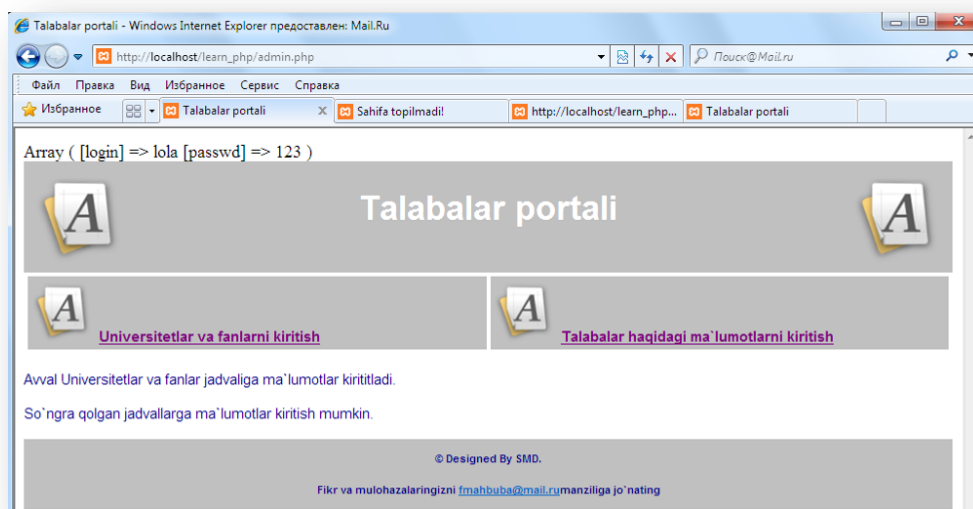
```

<?php
session_start();
// yangi sessiya yaratamiz yoki
// joriysini tiklaymiz
print_r($_SESSION);
// sessiyaning barcha o'zgaruvchilarini chiqazamiz
?>

<html>
<head>
<title>Talabalar portali</title>
<style>
<!--
Davomi bor ...

```

У ҳолда браузерда администратор саҳифасини ишга тушириб, куйидагини кўришимиз мумкин:



15.6 - расм. `admin.php` файлининг браузердаги кўриниши

Натижада `authorize.php` рўйхатдан ўтказилган ўзгарувчилар рўйхатини ва махфий саҳифани ўзини оламиз.

Бу бизга нима беради? Фараз қилинг, хакер администратор саҳифасига кириб маълумотлар базасидаги маълумотларни ўзгартирмоқчи. Ва у администратор саҳифаси (ёки саҳифалари)ни қандай номланишини билиб

олди. У ҳолда у браузер сатрида унинг манзилини киритишга ҳаракат қилиб кўради. Бу ҳолатларга учрамаслик учун, администратор саҳифасида қуйидагича код сатрларини ёзиш керак:

15.3с - листинг. admin.php нинг 2 - усули

```
<?php
session_start();
// yangi sessiya yaratamiz yoki
// joriysini tiklaymiz
print_r($_SESSION);
// sessiyaning barcha o'zgaruvchilarini chiqazamiz
if (!($_SESSION['login']=="lola" &&
$_SESSION['passwd']==123))
// parol va loginni
//to'g'riligini tekshiramiz
Header("Location: authorize.php");
// agar xato bo'lsa avtorizatsiya
//sahifasiga yo'naltiramiz
?>

<html>
<head>
<title>Talabalar portali</title>
<style>
<!--
Davomi bor ...
```

Сессия ўзгарувчиларини ўчириш

Сессия ўзгарувчиларини рўйхатга олиш(ишнинг барча сеанслари давомида уни глобал этиш) дан ташқари, бундай ўзгарувчиларни ва бутун сессияни ўчириш ҳам зарур.

session_unregister(ўзгарувчи_ному) функцияси жорий сессиядаги глобал ўзгарувчиларни ўчиради. Агар рўйхатга олиш *\$_SESSION* (PHP 4.0.6 ва ундан аввалги версиялари учун *\$HTTP_SESSION_VARS*) ёрдамида амалга оширилган бўлса, у ҳолда *unset()* тил конструкцияси ишлатилади. У кўрсатилган ўзгарувчини ўчиради, натижаси сифатида эса ҳеч қандай қийматни қайтармайди.

Бу каерларда қўл келади(ас қотади)? Масалан, махфий саҳифадан чиққандан сўнг ташриф буюрувчи ҳақидаги маълумотлар(логин ва парол)ни ўчириш учун ишлатилади. Агар тўғри логин ва парол сақланаётган бўлса ва сайтга ташрифдан сўнг браузер ойнаси ёпилмаган бўлса, у ҳолда бошқа ихтиёрий фойдаланувчи бу компьютерда ёпиқ ахборотни ўқиши мумкин.

15.4 - мисол. Сессия ўзгарувчиларини ўчириш.

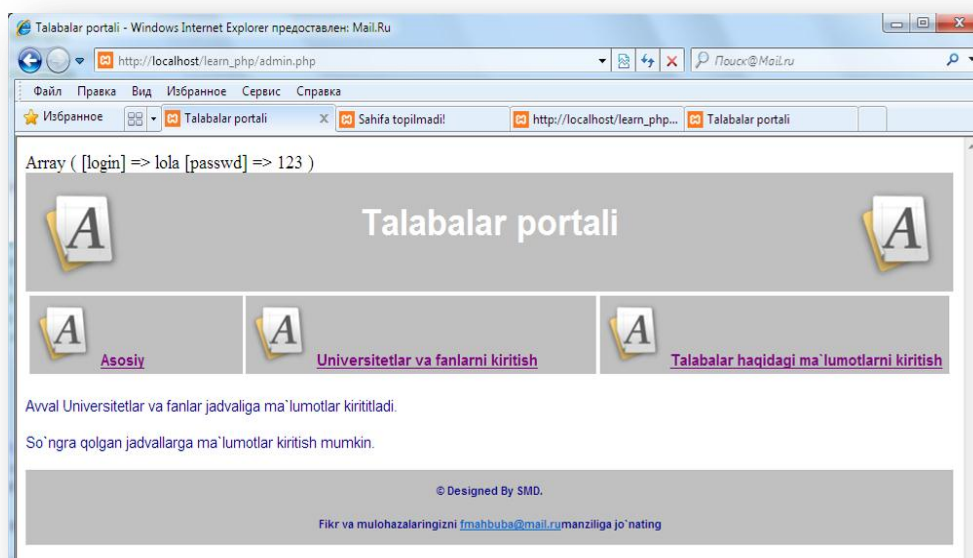
15.4а – листинг. admin.php файли. admin.php файлида асосий саҳифага қайтиш учун сатрлар ёзамиз:

```
<?php
session_start();
// yangi sessiya yaratamiz yoki
// joriysini tiklaymiz
print_r($_SESSION);
// sessiyaning barcha o'zgaruvchilarini chiqazamiz
if (!($_SESSION['login']=="lola" &&
$_SESSION['passwd']==123))
// parol va loginni
//to'g'riligini tekshiramiz
```

```

Header("Location: authorize.php");
// agar xato bo'lsa avtorizatsiya
//sahifasiga yo'naltiramiz
?>
<html>
<head>
<title>Talabalar portali</title>
<style>
<!--
h1 {color: white; font-size :24p; text-align : center ; font-
family : arial , sans-serif}
.menu {color: white; font-size: 12pt; text-align : center ; font-
family : arial , sans-serif; font-weight : bold}
td {background: silver}
p {color :darkblue; font-size: 12pt; text-align: justify;font-
family: arial , sans-serif}
p.foot {color: darkblue; font-size :9pt; text-align: center; font-
family : arial , sans-serif ; font-weight : bold}
a: link , a : visited, a : active {color : white }
-->
</style>
</head>
<body>
<!-- page header -->
<table width="100%" cellpadding = 12 cellspacing =0 border = 0>
<tr bgcolor = silver>
<td align = left><img src = "logo.png "></td>
<td>
<h1>Talabalar portali</h1>
</td>
<td align = right><img src = "logo.png "></td>
</tr>
</table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
<tr >
<td width = "25%">
<img src = "s-
logo.png "> <span class=menu><a href="home.php">Asosiy</a></span></td>
<td width = "25%">
<img src = "s-
logo.png "> <span class=menu><a href="new_univ_subject.php">Universitetlar va
fanlarni kiritish</a></span></td>
<td width = "25%">
<img src = "s-
logo.png "> <span class=menu><a href="new_talaba.php">Talabalar haqidagi ma'l
umotlarni kiritish</a></span></td>
</tr>
</table>
<!-- page content -->
<p>Avval Universitetlar va fanlar jadvaliga ma'lumotlar kiritiladi. </p>
<p>So'ngra qolgan jadvallarga ma'lumotlar kiritish mumkin.</p>
<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr>
<td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href ="mailto:fmahbuba@mail.ru">f
mahbuba@mail.ru</a>manziliga jo'nating </p>
</td>
</tr>
</table>
</body>
</html>

```



15.7 - расм. *admin.php* файлининг браузердаги кўриниши

15.4b – листинг. *home.php* файли. *home.php* да сессияда сақланиб турган, аввал киритилган логин ва паролни ўчириш учун скрипт ёзамиз:

```
<?
session_start();
session_unregister('passwd');
// parolni o'chiramiz
unset($_SESSION['login']);
// loginni o'chiramiz
print_r($_SESSION);
// sessiyaning global o'zgaruvchilarini chiqazamiz

require("page.inc") ;
$homepage = new Page( );
$homepage -
> SetContent ("<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекист
он Республикасининг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-
техникавий ва маданий эктиёжларини кондирувчи устувор соқадир. </p>
<p>Узлуксиз таълим кадрлар тайёрлаш тизимининг асоси, Ўзбекистон Республикаси
нинг ижтимоий-
иктисодий тараққиётини таъминловчи, шахс, жамият ва давлатнинг иктисодий, ижт
имоий, илмий-техникавий ва маданий эктиёжларини кондирувчи устувор соқадир.
</p>
<p>Узлуксиз таълим ижодкор, ижтимоий фаол, маънавий бой шахс шаклланиши ва юк
ори малакали рақобатбардош кадрлар илдам тайёрланиши учун зарур шарт-
шароитлар яратади.</p>
");
$homepage -> Display ( ) ;

?>
```

Асосий саҳифага қайтганда сессияда сақланиб турган логин ва парол ўчирилади. Энди махфий саҳифага кириш учун, яна қайтадан логин ва паролни киритилади.

Сессиянинг барча ўзгарувчиларини ташлаб юбориш учун *session_unset()* функциясидан фойдаланиш мумкин.

Жорий сессияни бутунлай ўчириш учун `session_destroy()` буйруғидан фойдаланилади. Бу функция сессиянинг глобал ўзгарувчиларининг қийматларини ташлаб юбормайди ва *cookies* ни ҳам ўчирмайди, бироқ жорий сессия билан боғлиқ барча маълумотларни ўчиради.

15.5 - мисол. Сессия ва глобал ўзгарувчиларни ўчириш

```
<?
session_start(); // sessiyanı initsiaziyalash
$test = "Sessiya o`zgaruvchisi";
$_SESSION['test'] = $test;
// $test o`zgaruvchisini ro`yhatdan o`tkazamiz.
// agar register_globals=on bo`lsa, u holda
// session_register('test');
// ni ishlatish mumkin
print_r($_SESSION);
// barcha global o`zgaruvchilarni chiqazamiz

echo session_id();
// sessiya identifikatorini chiqazamiz
echo "<hr>";
session_unset();
// sessiyaning barcha global
// o`zgaruvchilarini o`chiramiz
print_r($_SESSION);
echo session_id();
echo "<hr>";
session_destroy(); // sessiyanı o`chiramiz
print_r($_SESSION);
echo session_id();
?>
```

```
Array ( [test] => Sessiya o`zgaruvchisi ) e030d0d7708d3e09b6dbc28c235c6ddc
Array ( ) e030d0d7708d3e09b6dbc28c235c6ddc
Array ( )
```

Бу скрипт ишининг натижаси сифатида учта сатр чиқазилади: биринчиси - қиймати мавжуд `test` элементида эга массив, ҳамда *сессия идентификатори*, иккинчиси - бўш массив ва *сессия идентификатори*, учинчиси – бўш массив. Бундан кўриниб турибдики, *сессияни ўчиргандан сўнг унинг идентификатори* ҳам ўчирилади ва биз ўзгарувчиларни бошқа рўйхатдан ўтказа олмаймиз, ҳамда *сессия* устида ҳеч қандай ишни амалга ошириб бўлмайди.

15.5. Хавфсизлик

Сессия механизмини ишлатиш системанинг тўлиқ хавфсизлигига жавоб бермайди. Бунинг учун қўшимча чоралар кўрилиши керак. Сессия билан ишлаш жараёнида юзага келиши мумкин бўлган хавфсизлик муаммосига эътиборимизни қаратамиз, жумладан, биз ёзган дастурлаш мавзусига.

Биринчидан, паролни турли саҳифаларга узатиш хавфли, чунки уни билиб олишлари мумкин. Бундан ташқари, биз уни сессиянинг глобал ўзгарувчиси сифатида рўйхатдан ўтказдик, демак, у мижоз компьютерининг *cookies* ида сақланади. Бу эса ёмон. Умуман олганда, парол ва логинни яхшиси маълумотлар базасида сақлаш керак. Фойдаланувчи ҳақидаги маълумотлар "test" маълумотлар базасининг "users" жадвалида сақланса ҳам, биз унга `my_user` логини ва `my_passwd` паролни билан кира оламиз.

Иккинчидан, агар кимдир махфий саҳифа учун паролларни саралаш скриптини ёзса нима қилиш керак? Бу ҳолда авторизация саҳифасида бирорта бегона скрипт бир неча марта эшик қоқиб туради. Шунинг учун бизнинг сайтнинг авторизациясига сўров келганлигини текшириш керак, агар келмаса, уни давом эттиришга йўл қўймаслик керак. Сўров қабул қилган саҳифа манзилини `$_SERVER['HTTP_REFERER']` глобал ўзгарувчиси ёрдамида олиш мумкин. Шундай бўлса ҳам, албатта, агар сайтга кириш учун уни бузишга жиддий ёндошилган бўлинса, у ҳолда бу ўзгарувчи қийматини ўзгартириш керак. Шунга кўра унинг қийматини текширишни, сайтнинг хавфсизлигини таъминлашнинг асосий қадамларидан бири деб ҳисоблаш мумкин.

`ucqv` маълумотлар базасида `phpmyadmin` ёрдамида `users` жадвалини яратамиз. Бу жадвал учта майдондан иборат бўлиб, биринчи `id` майдони, `name` майдонида фойдаланувчилар номи ва `password` майдонида пароллари сақланади.

```
CREATE TABLE `users` (
  `id` INT( 6 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 20 ) NOT NULL ,
  `password` VARCHAR( 20 ) NOT NULL ,
  UNIQUE (
    `name`
  )
)
ENGINE=MYISAM;
```

Ва ушбу `users` жадвалига битта фойдаланувчи логини ва паролни киритамиз.

```
INSERT INTO `users` ( `id` , `login` , `password` )
VALUES (
  NULL , 'lola', '123'
);
```

15.6 - листинг. authorize.php

```
<?
session_start();
// yangi sessiya yaratamiz
```

```

// yoki joriysini tiklaymiz
$conn = mysql_connect("localhost", "ilyosbey", "135");
// MB ga bog'lanishni o'rnatamiz
mysql_select_db("uquv");
// ishchi MB sini tanlaymiz
$SERVER_ROOT = "http://localhost/learn php/";
// skriptlar qayerda joylashgan bo'lsa
/* ^$SERVER_ROOT doimiy ifoda va eregi funksiyasi yordamida jo'natilayotgan
skript
$_SERVER['HTTP_REFERER']) satri
$SERVER_ROOT satri bilan boshlanayaptimi tekshiramiz*/
if(eregi("^$SERVER_ROOT", $_SERVER['HTTP_REFERER']))
{
    // agar ha bo'lsa, u holda avval bajargan ishimizni bajaramiz
    // parolni ro'yhatdan o'tkazmaymiz
    if (!isset($_POST['go']))
    { echo "<form method=POST >
      Login: <input type=text name=login>
      Password: <input type=password name=passwd>
      <input type=submit name=go value=Go>
      </form>";
    }
    else {
        /* ma'lumotlar bazasiga so'rov:
        jadvaldan so'rovda jo'natilgan, foydalanuvchi tomonidan kiritilgan
        loginga mos users login ni, parolga mos parolni olamiz
        Agar jadvalda bunday login va parol mavjud bo'lmasa, login va parol n
        oto'g'ri kiritilgan deb hisoblaymiz*/
        $sql = "SELECT login FROM users
        WHERE login='" . $_POST['login'] . "' AND password='" . $_POST['passwd']
        . "'";
        $q = mysql_query($sql, $conn); // MB ga so'rovni jo'natamiz
        $n = mysql_num_rows($q); // so'rov javobidagi satrlar soni
        if (!$n==0){
            $_SESSION['user_login']=$_POST['login'];
            // login o'zgaruvchisini ro'yhatdan o'tkazamiz
            Header("Location: admin.php");
            // secret_info.php sahifasiga yo'naltiramiz
        }else echo "Kiritilgan login va parol noto'g'ri, qayta urunib ko'ring<br>";
        }
        print_r($_SESSION); // sessiyani barcha o'zgaruvchilarini chiqazamiz
    }
}
?>

```

Биринчи иккита муаммо ҳал этилди. Лекин биттаси қолди. Агар хакер сўров сатрига бирор – бир глобал ўзгарувчининг қийматини (масалан, логинни) ёзса, нима қилиш керак? Умуман олганда агар register_globals=On бўлсагина бу мумкин. Ҳар ҳолда биз глобал ўзгарувчилар билан ишлаш учун `$_SESSION` массивни ишлатамиз, унга эса бундай ҳийлалар ўтмайди. Шундай бўлса ҳам бу муаммони ечишга ҳаракат қиламиз. Бунинг учун параметр қийматини солиштиришдан олдин сўров сатрини тозалаш керак. У ҳолда аввал `$user_login` қийматини ташлаб юборамиз. Кейин ушбу ўзгарувчини яна рўйхатдан ўтказамиз, фақатгина янги эмас, балки аввал мавжуд ўзгарувчи сифатида. Бунинг учун рўйхатдан ўтказишда доллар белгиси тушириб қолдирилмайди.

15.7 - листинг. *admin.php*.

```
<?php
```

```
unset($user_login); // o`zgaruvchini o`chiramiz
session_start(); // yangi sessiya yaratamiz yoki
// joriysini tiklaymiz
session_register($user_login);
// avval mavjud kabi o`zgaruvchini
// ro`yhatdan o`tkazamiz
if (!$user_login=="lola") // loginni tekshiramiz
    Header("Location: authorize.php");
// agar xato bo`lsa,
// avtorizasiya sahifasiga yo`naltiramiz
?>

<html>
<head>
<title>Talabalar portali</title>
<style>
<!--
Davomi bor ...
```


АДАБИЁТЛАР РЎЙХАТИ

1. Арипов М., Мухаммадиев Ж. Информатика, информацион технологиялар. (Хуқуқшунослик мутахассисликлари учун дарслик) Т. 2004 й
2. Арипов М ва бошқалар «Информатика, информацион технологиялар» (Бакалавр учун ўқув қўлланма) Т., ТДТУ. 1-2 қисм. 2003 й 320-430 б.
3. Арипов М ва бошқалар «Информатика, информационные технологии» (Бакалавр учун ўқув қўлланма) Т., Рус тилида. 2003 й 300 б.
4. Арипов М. и др. «Основы Internet» Т.; Университет 2002 й. 194 б.
5. Арипов М. Internet ва электрон почта асослари. Т.; Университет 2000 й. 126 б
6. Арипов М. Англо-русско-узбекский словарь сокращенных слов по информатике. Т.; Университет 2001 й. 145 б.
7. Арипов М., Тиллаев А. Web саҳифалар яратиш технологиялари. Т. 2006 й. 170 б.
8. Бегимқулов У.Ш., Мамаражабов М.Э., Турсунов С . FLASH MX дастури ва ундан таълимда фойдаланиш имкониятлари Т. ТДПУ. 2006 й.
9. Орлов Л.В. «Web–сайт без секретов». М.: ЗАО “Новый издательский дом”, 2004г.
10. А. Матросов, А. Сергеев, М. Чаунин. HTML 4.0. «БХВ–Санкт – Петербург», 2003г.
11. Кришномурти Б., Рексфорд Дж. «Web–протоколы. теория и практика».–М.: ЗАО “Издательство БИНОМ”, 2002г.
12. Дротов В.А. «JAVA SCRIPT в Web–дизайне». СПб. «БХВ-Петербург», 2002г.
13. Голиш Л.В. «Технология обучения на лекциях и семинарах в экономическом ВУЗе». Учебное пособие, Ташкент, ТГЭУ, 2005г.
14. Максимова О.В., Невзорова В.И. «Информационные технологии для экономистов». Учебное пособия. Ростов на Дону: Феникс, 2004г.
15. Шафран Э. Создание Web-страниц: самоучитель-СПб:Питер,2001.
16. Шарипов М.С., Эржонов Х.Д. «HTML» Ўқув қўлланма, Тошкент, НУДУ, 2006 г.
17. И. Ф. Астахова, Т. В. Курченкова, Р. А. Дураков, И. С. Битюцких, Д. В. Комаров. «Web-технологии с базами данных» , Воронеж, ИПЦ ВГУ, 2008.
18. Томсон Л. Разработка Web-приложений на PHP и MySQL : пер. с англ. / Л. Томсон, Л. Веллинг. – 2-е изд., испр. – М. : DiaSoft, 2003.

19. Гилмор Г. PHP 4. Учебный курс / Г. Гилмор. – СПб. : Питер, 2001.
20. Гончаров А. Самоучитель HTML / А. Гончаров. – СПб. : Питер, 2002.
21. Дейт К. Дж. Введение в системы баз данных / К.Дж. Дейт. – 7-е изд. – М. : Вильямс, 2001.
22. Коннолли Т. Базы данных. Проектирование. Реализация. Сопровождение / Т. Коннолли, К. Бегг. – М. : Вильямс, 2003.
23. Мазуркевич А. PHP: настольная книга программиста / А. Мазуркевич, Д. Еловой. – Мн. : Новое знание, 2004.
24. Мелони Д. PHP 4 в действии : пер. с англ. / Д. Мелони, П. Вайнберг. – М. : Лучшие книги, 2002.
25. Профессиональное PHP программирование : пер. с англ. / Дж. Кастаньетто, Х. Рафат, С. Шуман, и др. – СПб. : Символ-Плюс, 2001.
26. Петюшкин А. В. HTML. Экспресс-курс / А. В. Петюшкин. – СПб. : БХБ-Петербург, 2003.
27. Харрис Э. PHP/MySQL : пер с англ. / Э. Харрис. – М. : КУДИЦ-ОБРАЗ, 2005.
28. Олищук А.В. Разработка WEB-приложений на PHP 5. Профессиональная работа / А. В. Олищук, А. Н. Чаплыгин. – М. : Вильямс, 2006.

Интернет Web-сайтлари:

- a. <http://www.cgi.ru>
- b. <http://www.woweb.ru>
- c. <http://www.ru>
- d. <http://www.vanta.ru/script/>
- e. <http://www.vbnet.ru>
- f. <http://www.scriptic.ru/>
- g. <http://www.webacademy.com/>
- h. [http:// www .intuit.ru/](http://www.intuit.ru/)

МУНДАРИЖА

КИРИШ	2
1 ҚИСМ. PHP ТЕХНОЛОГИЯСИ	3
I БОБ. PHP ВА MySQL	3
1.1. PHP тарихи ва унинг имкониятлари	3
1.2. MySQL: тарих ва имкониятлар	8
1.3. ХАМРР дастурий таъминотини ўрнатиш ва созлаш	13
1.4. Виртуал серверлар (хостлар) яратиш	25
II БОБ. HTML	29
2.1. Web саҳифа яратиш ва матнлар билан ишлаш	29
2.2. Матнларни форматлаш. Web саҳифаларда графика	32
2.3. Гипермурожаат (гиперссылкалар) лардан фойдаланиш	36
2.4. Фреймлар ёрдамида сайт яратиш	39
2.5. Жадвал яратиш	44
2.6. Стиллар жадвали, шрифтлар, махсус белгилар. Тасвирлар ва мурожаатлар картасининг қўшимча хусусиятлари	51
III БОБ. PHP ТЕХНОЛОГИЯСИ	59
3.1. PHP технологияси ҳақида қисқача	59
3.2. PHP дескрипторларини ишлатилиши	62
3.3. PHP операторлари	63
3.4. Изоҳлар	64
IV БОБ. ЎЗГАРУВЧИЛАР ВА МАЪЛУМОТЛАР ТИПИ	65
4.1. Ўзгарувчи типлари	65
4.2. Идентификаторлар	77
4.3. Ўзгарувчилар билан ишловчи функциялар	78
V БОБ. АМАЛЛАР	80
5.1. PHP амаллари	80
5.2. Сатрлар билан ишлаш амаллари	81
5.3. Арифметик амаллар	83
5.4. Мантикий бинар амаллар	86
5.5. Тенглик амали	87
5.6. Муносабат(таққослаш) амали	88
5.7. Хатоларни яшириш амали	89
VI БОБ. PHP ЁРДАМИДА СЎРОВЛАРНИ ҚАЙТА ИШЛАШ	90
6.1. Мижоз-сервер технологияси асослари	90
6.2. HTTP протоколи ва маълумотларни серверга жўнатиш методлари	96
6.3. Мижоз сўрови шакли	97
6.4. Методлар	99
6.5. HTML формасини серверга маълумотларни жўнатиш учун қўлланилиши	100
6.6. Сўровларни PHP ёрдамида қайта ишлаш	107
6.7. Формалар билан ишлашга мисоллар	110
VII БОБ. БОШҚАРУВЧИ КОНСТРУКЦИЯЛАР	117
7.1. Танлаш операторлари	117
7.2. Такрорлаш(цикл) операторлари	125
7.3. Ўтиш операторлари	129
7.4. Қамраб олиш операторлари	132
VIII БОБ. ФУНКЦИЯЛАР	139
8.1. Фойдаланувчи томонидан аниқланган функциялар	139
8.2. Регистр ва функция номи	141
8.3. Функцияга мурожаат этиш	142
8.4. Параметрлар (ёки Аргументлар)	144
8.5. Ҳаракатлар майдони	150
8.6. Функция ичида ўзгарувчиларни ишлатилиши	151

8.7. Қайтарилувчи мазмунлар	153
8.8. Қайтарилувчи жўнатмалар	154
8.9. Ўзгарувчи функциялар	155
8.10. Ички функциялар	156
IX БОБ. ФАЙЛЛАР БИЛАН ИШЛАШ	162
9.1. Маълумотларни сақлаш	162
9.2. Файлни очиш ва файлга боғланиш	163
9.3. Файл билан боғланишни ёпиш	167
9.4. Файл мавжудлигини текшириш	168
9.5. Файлга маълумот ёзиш	170
9.6. Файлдан маълумотларни ўқиш	174
9.7. Файллар билан ишловчи функциялар	180
9.8. Файлни муҳофазалаш (блокировка)	185
9.9. Серверга файлни юклаш	187
9.10. Қайта ишлашнинг рационал усули: Маълумотлар базасини бошқариш тизими	191
X БОБ. МАССИВЛАР	193
10.1. Массивлар	193
10.2. Массивлар устида амаллар	196
10.3. Массивлар билан ишловчи функциялар	197
10.4. Массивларни саралаш	204
10.5. Кўп ўлчовли массивларни саралаш	208
10.6. Массивнинг ҳар бир элементи учун ихтиёрий функцияни қўллаш	212
10.7. Қисм массивни белгилаш	214
10.8. Массив элементлари йигиндисини ҳисоблаш	216
10.9. Массив элементларини жойлашган тартибини ўзгартириш	217
10.10. Маълумотларни файлдан массивга юклаб олиш	220
10.11. Массивлар устида бошқа амалларни бажариш	223
XI БОБ. САТРЛАР ВА ДОИМИЙ ИФОДАЛАР (РЕГУЛЯРНЫЕ ВЫРАЖЕНИЕ)	227
11.1. Сатрлар билан ишлаш	227
11.2. Сатрларни таҳрирлаш	229
11.3. Сатрларни таққослаш	235
11.4. Сатрлар билан ишловчи функциялар ёрдамида сатрларни ажратиш ва бирлаштириш	238
11.5. Қисм сатрларни қидириш ва алмаштириш	245
11.6. html-кодлардан таркиб топган сатрлар	251
11.7. Сатрлар билан ишлашга мисоллар	252
XII БОБ. ОБЪЕКТГА ЙЎНАЛТИРИЛГАН ДАСТУРЛАШ	257
12.1. Класслар ва объектлар	257
12.2. Ўзгарувчиларни инициализацияси (лавҳаси)	260
12.3. Объект яратиш ва у билан ишлаш	262
12.4. Ворислик (наследование)	265
12.5. Кўп даражали ворислик (Многоуровневое наследование)	269
12.6. Конструкторлар	270
12.7. Деструкторлар	271
12.8. Мавҳум класслар	272
12.9. Методларни ортиқчалиги (Перегрузкой методов)	273
12.10. Масаланинг қўйилиши	274
2 ҚИСМ. МАЪЛУМОТЛАР БАЗАСИ БИЛАН ИШЛАШ	281
XIII БОБ. МАЪЛУМОТЛАР БАЗАСИ	281
13.1. Маълумотлар базаси ҳақида асосий тушунчалар	281
13.2. SQL тили тарихи ва вазифалари	291
13.3. WEB –маълумотлар базасини ташкил этиш	293
13.4. MySQL маълумотлар базасини бошқариш тизими (МББТ)	295
13.5. MySQL МББТ нинг асосий операторлари	300
XIV. PHP ВА MYSQL НИНГ ЎЗАРО БОҒЛАНИШИ	321
14.1. PhpMyAdmin ёрдамида MySQL маълумотлар базаси билан ишлаш	321

14.2. MySQL тизими билан боғланишни ўрнатиш ва тизимда ишлаш учун маълумотлар базасини танлаш	326
14.3. Танланган базада сўровларни қайта ишлаш	331
3 ҚИСМ. ФОЙДАЛАНИШ УЧУН РУХСАТ	349
XV БОБ. СЕССИЯЛАР БИЛАН ИШЛАШ	349
15.1. Фойдаланиш учун рухсат	349
15.2. Сессия механизми	354
15.3. Сессияни созлаш	355
15.4. Сессиялар билан ишлаш	357
15.5. Хавфсизлик	365
АДАБИЁТЛАР РЎЙХАТИ	368





