

**O'ZBEKISTON RESPUBLIKASI AXBOROT  
TEKNOLOGIYALARI VA KOMMUNIKATSIYALARNI  
RIVOJLANTIRISH VAZIRLIGI**

**TOSHKENT AXBOROT TEKNOLOGIYALARI UNIVERSITETI  
URGANCH FILIALI**

---

“Kompyuter injiniring” fakulteti

“Dasturiy injiniring” kafedrası

**XO'JAYEV O.Q., ARTIKOV M.E.**

**«Veb ilovalarni yaratish» fanidan tajriba ishlarini  
bajarish bo'yicha**

**USLUBIY KO'RSATMA**

Yo'nalish:

5330500 – Kompyuter injiniringi (Kompyuter injiniringi)

III Kurs

V Semestr

Urganch 2016

Xo'jayev O.Q., Artikov M.E. "Web ilovalarni yaratish" fanidan tajriba ishlarini bajarish bo'yicha uslubiy ko'rsatma. Urganch, TATU Urganch filiali, 2016. – 84 b.

"Web ilovalarni yaratish" fanidan tajriba ishlarini bajarish bo'yicha uslubiy ko'rsatma, bakalavriatura bosqichining, 5330500 – Kompyuter injiniringi (Kompyuter injiniringi), 5330600 – Dasturiy injiniring yo'nalishini 3 bosqich talabalari uchun mo'ljallangan. Talabalar, ushbu uslubiy ko'rsatmani o'qish davomida V semestrda o'qitiladigan, "Web ilovalarni yaratish" fanidan tajriba mashg'ulotlarini o'tkazish ko'nikmasiga ega bo'ladilar.

Taqrizchilar:

TATU Urganch filiali, "Dasturiy injiniring" kafedrası  
dotsenti F.Yusupov

UrDU, "Informatika" kafedrası  
mudiri t.f.n. Matlatipov G'.

.

Uslubiy ko'rsatma "DI" kafedrasining  
"22" fevral 2016 yildagi yig'ilishida ma'qullangan va  
nashrga tavsiya etilgan (Bayonnoma №17)

Uslubiy ko'rsatma TATU Urganch filialining  
"26" fevral 2016 yildagi ilmiy-uslubiy kengashida  
ma'qullangan va nashrga tavsiya etilgan (Bayonnoma №6)

## MUNDARIJA:

<b>Tajriba ishi №1</b> PhpStorm matn muharririni o'rnatish va sozlash. Google Chrome brauzerini o'rnatish.....	4
<b>Tajriba ishi №2</b> HTML teglari: H1-H6, p, a, table, ul, ol, div. Teglar atributlarini o'rganish.....	8
<b>Tajriba ishi №3</b> HTML5 yangi teglarini o'rganish: header, footer, section, article, aside, audio, video, svg, canvas.....	10
<b>Tajriba ishi №4</b> HTML5da formalar bilan ishlash teglari: form, input. Ularning xususiyatlarini o'rganish.....	13
<b>Tajriba ishi №5</b> CSSda sahifaga stil berish. Selektorlar tushunchasini o'rganish.....	19
<b>Tajriba ishi №6</b> Twitter Bootstrap CSS frameworkidan foydalanib sahifalar yaratish. Bootstrap komponentalarini ishlatish.....	26
<b>Tajriba ishi №7</b> JavaScript tilidan foydalanib, veb sahifaga interaktivlik qo'shish. Funktsiyalar yaratish.....	29
<b>Tajriba ishi №8</b> JQuery kutubxonasidan foydalanib elementlarga interaktivlik berish. Hodisalarni tadqiq qilish.....	33
<b>Tajriba ishi №9</b> XAMPP HTTP-serverini o'rnatish va sozlash. HTTP protokoli ishlash prinsipini o'rganish.....	37
<b>Tajriba ishi №10</b> PHPda ishlash asoslari. O'zgaruvchilarni e'lon qilish. PHP ichki funksiyalari bilan ishlash.....	41
<b>Tajriba ishi №11</b> PHPda funksiyalar yaratish. Satrlar bilan ishlovchi funksiyalarni qo'llash....	44
<b>Tajriba ishi №12</b> PHPda MySQL ma'lumotlar ombori bilan ishlash. MySQL asoslari. SELECT, INSERT, UPDATE va DELETE operatorlarini o'rganish.....	55
<b>Tajriba ishi №13</b> PHPda sinflar yaratish. Ob'yekt tushunchasi. Sinf maydonlari va metodlarini o'rganib chiqish.....	67
<b>Tajriba ishi №14-17</b> MVC loyihalash shablonini o'rganish. Zamonaviy PHP frameworklar: Zend, Yii, CodeIgniter, Laravel kabilarni taqqoslash.....	74
<b>Tajriba ishi №18</b> Tayyor veb ilovani hostingga joylashtirish va sozlash.....	81



## Tajriba ishi №1

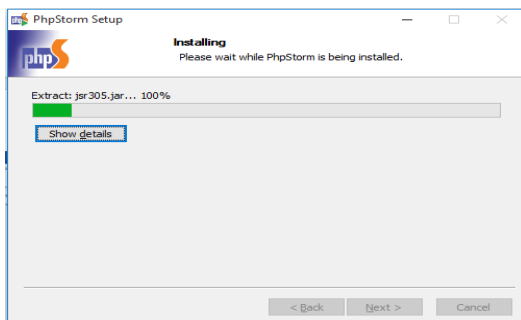
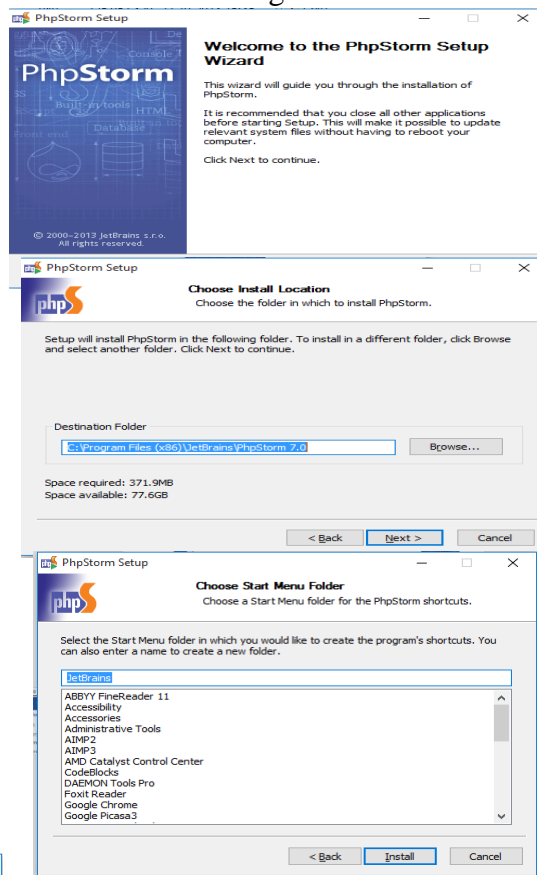
### Mavzu: PhpStorm matn muharririni o'rnatish va sozlash. Google Chrome brauzerini o'rnatish

**Ishdan maqsad:** PhpStorm matn muharririni o'rnatish va sozlashni o'rganish.

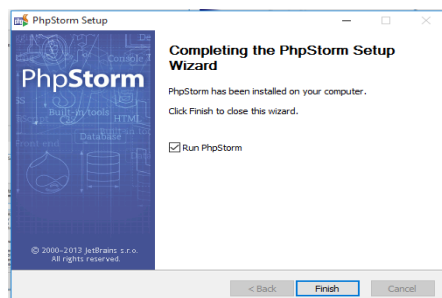
PhpStorm dasturi Php, Html va JavaScript dasturlash tillari uchun tezkor kodlashtirishni amalga oshiruvchi intellektual muharrir.

PhpStorm matn muharririni o'rnatish quyidagi ketma-ketlikda amalga oshiriladi:

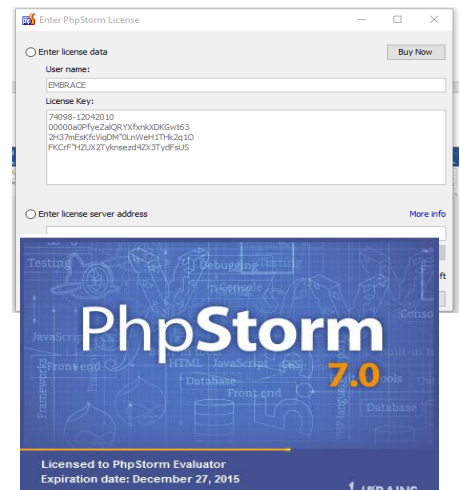
1. PhpStorm matn muharririni setup.exe fayli kerak bo'ladi. Quyidagi oyna hosil bo'ladi va NEXT tugmasi bosiladi:
2. Keyingi bosqichda PhpStorm qaysi popkaga ustanovka bo'layotgani ko'rsatadi. Bunda siz browse tugmasi orqali boshqa joyni ko'rsatishingiz mumkin. Keying bosqichga o'tish uchun NEXT tugmasi bosiladi
3. Quyidagi oyna paydo bo'lgach INSTALL tugmasi bosiladi va PhpStorm matn muharririni o'rnatish boshlanadi.



4. PhpStorm matn muharriri o'rnatib bo'lgach quyidagi oyna hosil bo'ladi. Bunda FINISH tugmasini bosib PhpStorm matn muharririni ishga tushirish mumkin.

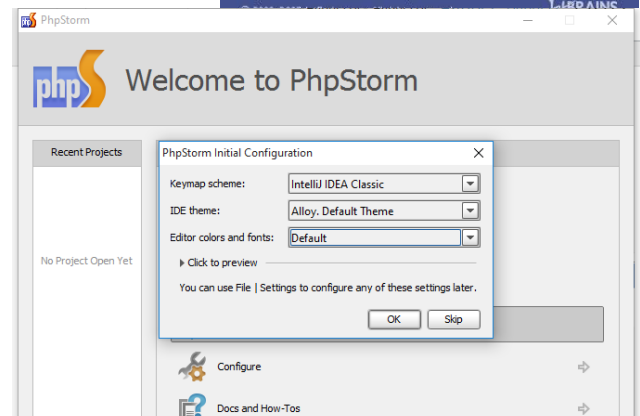


5. Bu dastur litsenziyaga ega bo'lgani uchun kalit so'zini so'raydi. Biz kalit so'zini kiritib, oynaning pastki qismidagi 30 kun uchun mo'ljallanganligiga belgi qo'yamiz va OK tugmasini bosamiz.

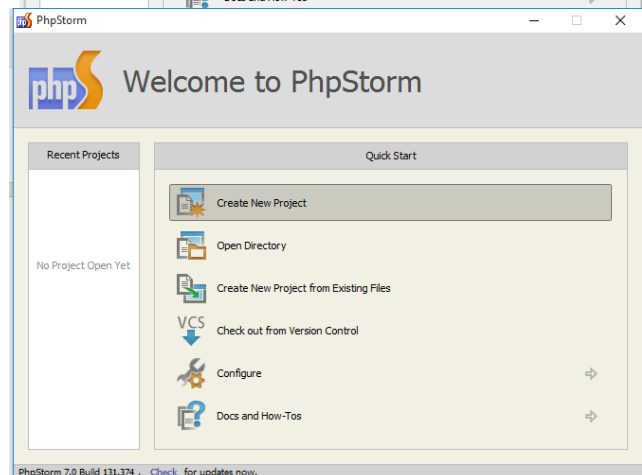


6. Shundan so'ng PhpStorm matn muharririni to'liq ishlash uchun tayyor bo'ladi.

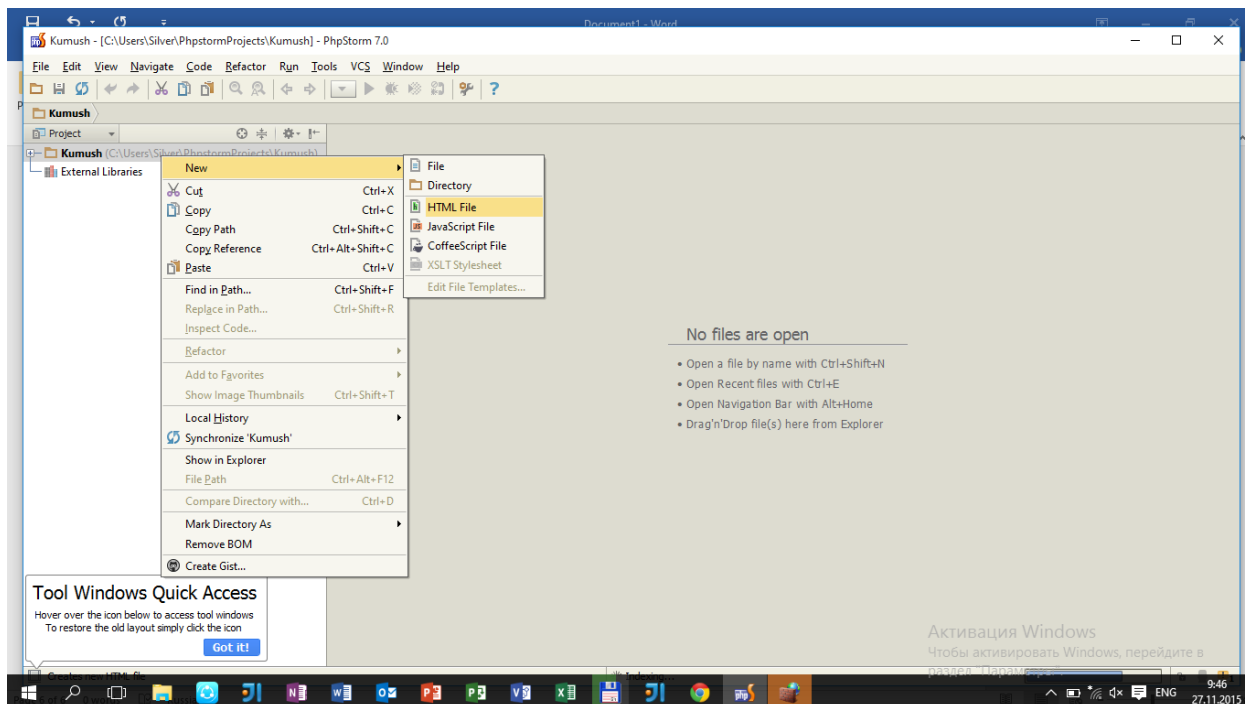
7. Oldin kichkina sozlashni amalga oshiramiz. Bunda PhpStorm matn muharririni qanday ko'rinishda bo'lishini va qanday rangda bo'lishini tanlab OK tugmasini bosamiz.



8. PhpStorm matn muharririda ishlash muhitini tanab bo'lgach yangi project yaratamiz va nomini yozamiz.



Yangi project ochilgach Html file yaratiladi va html kod yoziladi. Yozilgan kodni browser orqali ochib ko'rish mumkin.



## Html teglari haqida qisqacha ma'lumot.

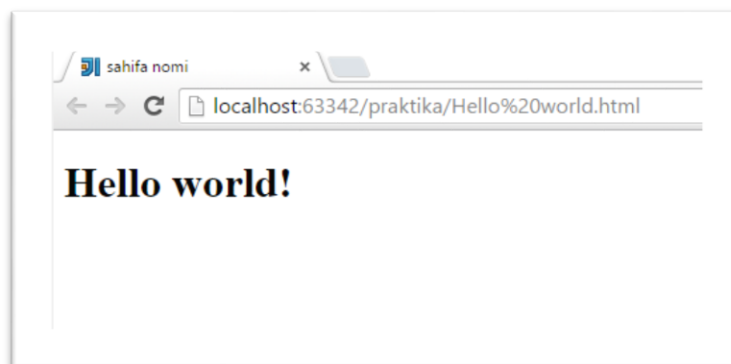
«<>» belgi ichidagi yozuvlar teg deb ataladi. Ko'p teglarning ochiq(<>) va yopiq(</>) hollari ishlatiladi.

1. **<html>...</html>** — ochiluvchi va yopiluvchi «html» nomli teg. Bu teg HTML hujjat boshlanishini bildiradi va barcha kodlar aynan shu teg ichida yoziladi.
2. **<head>...</head>** sahifaning bosh qismi. Bu tegdagi kodlar foydalanuvchilarga ko'rinmaydi. Oddiy sahifalar yaratishda judayam muhim teg hisoblanmaydi. Bu teg ichida quyidagi teglar joylashishi mumkin:
  - html meta teglari(meta)
  - sahifa nomi(title)
  - JavaScript kodlarini ulash(JavaScript)
  - stillarni qo'shish(link)
3. **<title>...</title>** — sahifa nomini berish. Bu yerga yozilgan yozuvlar brauzerning yuqori qismida namoyon bo'ladi.
4. **<body>...</body>** — saqifaning tana qismi. Asosiy qism hisoblanadi va bu qism foydalanuvchiga bevosita namoyon bo'ladi. Barcha asosiy kodlar aynan shu teg ichida joylashgan bo'ladi.
5. **<!-- ... -->** — bu belgilar kommentariyani bildiradi, brauzerda bularning ichidagi yozuv ko'rinmaydi. Kodlarni tushuntirishda kommentariyalardan foydalanish foydali bo'ladi.

“Hello world” so'zini browser ekraniga chiqarish

```
<!DOCTYPE html>
<html>
<head>
  <title>Sahifa nomi</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

Natija:





## Tajriba ishi № 2

**Mavzu: HTML teglari: H1-H6, p, a, table, ul, ol, div. Teglar atributlarini o'rganish.**

**Ishdan maqsad:** HTML teglari: H1-H6, p, a, table, ul, ol, div teglari va atributlari bilan tanishish.

### Nazariy qism

1. **<center>...</center>** — bu teg ichidagi barcha yozuvlar(teglar ham bo'lishi mumkin) sahifaning markazidan boshlab yoziladi, shunda umumiy yozuv o'rtada namoyon bo'ladi.

2. **<hr>** — bu tegning yopiluvchi sherigi yo'q. Teg oddiy chiziq chizib beradi. Uning quyidagi atributlari mavjud:

**align** — chiziqni tekisligini o'rnatadi, quyidagi qiymatlarni qabul qilishi mumkin:

\*left — chap tomondan tekislash

\*right — o'ng tomondan tekislash

\*center — markazga tekislash.

**color** — chiziq rangini berish(rang nomi yoki 16lik tizimdagi raqami).

**noshade** — effektlarsiz chizish.

**size** — chiziq qalinligi(size=«20»).

**width** — chiziq uzunligi, piksel yoki fozda beriladi(width=«20%»).

3. **<br>** — kursorni bitta pastga tushirish, undan keyin keladigan barcha narsalar(jadval, yozuv,...) bitta pastdan yoziladi(qisqasi Enter). Yopiluvchi sherigi yo'q.

4. **<h1>...</h1>** — Matnga nom qo'yish(zagalovka). Bu klass 6 xil ko'rinishda yozilishi mumkin(h1, h2, h3, h4, h5, h6). Raqamlar kattalashgan sari yozuvlar o'lchami kichiklashib boradi. Bu teg har xil o'lchamdagi nomlar(bo'lim nomi, band nomi, bob nomi,...) berishga qulay. Uning bitta atributi bor:

**align** — yozuvni tekislash, qiymatlari quyidagilar:

\*left — chap tomonga to'g'irlash(odatiy holi).

\*center — markazga to'g'irlash.

\*right — o'ng tomonga to'g'irlash.

\*justify — butun qator bo'ylab to'g'irlash.

5. **<p>...</p>** — paragraflar qo'shish uchun ishlatiladi. Har bir ochilgan «p» teg yozuvni, yangi qatordan boshlab ekranga chiqaradi. Bu tegning quyidagi atributi bor:

**align** — yozuvni tekislash, u quyidagi qiymatlarni qabul qiladi:

\*left — chap tomonga tekislash;

\*right — o'ng tomonga tekisash;

\*center — markazga tekislash;

\*justify — butun qator bo'ylab tekislash;

6. **<font>...</font>** — yozuvni formatlangan holda chiqarib beradi. Quyidagi atributlari mavjud:

**color** — yozuv rangini o'rnatish;

**face** — yozuv shriftini oʻrnatish(times new roman, verdana, arial);

**size** — yozuv oʻlchamini oʻrnatish(1 dan 7 gacha qiymat).

**Misol:**

```
<html>
<head> <title>Hello</title></head>
<body bgcolor=«orange" text=«black">
MarkaZ EmaS
<center>MarkaZ</center>
<hr color=«red" >
Mashhur Manchester United klubi<br>
1878 yilda tashkil etilgan!
<h1 align="center">Giggs</h1>
<h2 align="right">Young</h2>
<h3 align="left">Valencia</h3>
<h4>Phil John</h4>
<h5>Wayne Rooney</h5>
<h6>Ferdinand</h6>
<p>1. AkmX.uz</p>
<p align="center"><font size=«4" color=«green" face=«verdana">Manchester United 2012/2013 yili
g'olibi</font></p>
</body>
</html>
```

## Tajriba ishi №3

**Mavzu: HTML5 yangi teglarini o'rganish: header, footer, section, article, aside, audio, video, svg, canvas.**

**Ishdan maqsad:** HTML5 yangi teglarini o'rganish.

### Nazariy qism

Sayt strukturasini yaratishda asosan biz <div> tegidan yoki jadvallardan foydalanamiz. Saytning qismini <div> tegi orqali ajratamiz. Html5 da esa sayt strukturasini uchun alohida teglar yaratilgan. Bu teglar quyidagilar:

<HEADER>- sahifa nomi uchun teg.

```
<header>

  <h1>This is the Title of the Content</h1>

  <div class="post-meta">

    <p>By Author</a> <span class="category">Filed in Web 2.0</span></p>

  </div>

</header>
```

<FOOTER> - saytning eng pastki qismi uchun zarur bo'lgan teg.

Asosan bu tegning ichiga "copyright" kabi ma'lumotlar yoziladi.

```
<footer>
  <div class="tags">
    <span class="tags-title">Tags:</span> <a href="#" rel="tag">Command
    Prompt</a>, <a href="#" rel="tag">Compass</a>, <a href="#" rel="tag">CSS</a>, <a href="#"
    rel="tag">Sass</a>, <a href="#" rel="tag">Terminal</a>
  </div>
  <div class="facebook-like">
    <div>10 likes</div> <!-- let's pretend it to be the facebook like -->
  </div>
</footer>
```

<NAV> - sayt navigatsiyasi uchun teg.

```
<nav>
  <ul>
    <li><a href="#overview">Overview</a></li>
    <li><a href="#history">History</a></li>
    <li><a href="#development">Development</a></li>
  </ul>
</nav>
```

<ASIDE> - asosan o'ng yoki chap menyular yaratish uchun ishlatish mumkin bo'lgan teg.

<SECTION> - sahifadagi asosiy ma'lumotlarni chiqarish uchun.

<ARTICLE> - yangiliklar, kommentariylar, maqolalarning kontentlarini chiqarish uchun.

<article>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus id felis et augue sagittis euismod quis at sem. Nunc sit amet magna ac velit congue ultricies. Sed eros justo, lacinia in fringilla sollicitudin, congue id massa. Nunc dignissim bibendum nibh, sed dictum massa pharetra sit amet.</p>

</article>

<VIDEO> - saytga video joylashtirish uchun.

<video width=«600" height=«400" controls=«controls" poster=«video/poster.jpg">

<source src=«video/video.mp4" type='video/mp4; codecs=«avc1.42E01E, mp4a.40.2'>

Teg video ne podderjivaetsya vashim brauzerom.

<a href="video/video.mp4">Skachayte video</a>.

</video>

<AUDIO> - saytga audio joylashtirish uchun.

<audio src=«video/1.mp3" controls=«controls"></audio>

<CANVAS> - bazadan qiymat olib diagramma shaklida chiqarish uchun.

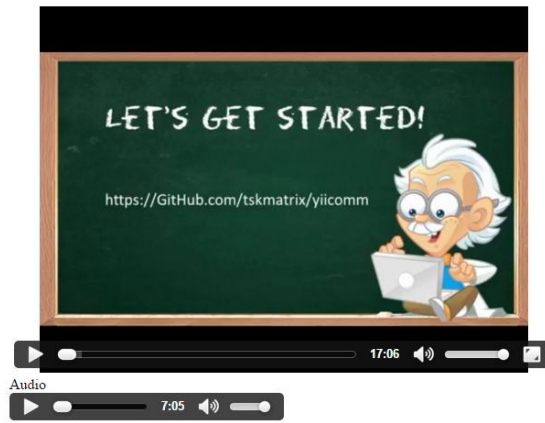
<canvas id=«smile" width=«200" height=«200">

<p>Vash brauzer ne podderjivaet risovanie.</p>

</canvas>

Bu teglarni kerakli joylarda to'g'ri ishlatish sayt strukturasini tushunarli (tartibli) bo'lishiga olib keladi.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>video</title>
</head>
<body>
  <video width="600" height="400" controls="controls" poster="video/poster.jpg">
    <source src="video/video.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
    <a href="video/video.mp4">Скачайте видео</a>.
  </video>
  <br>
  Audio<br>
  <audio src="video/1.mp3" controls="controls"></audio>
</body>
</html>
```



## Tajriba ishi №4

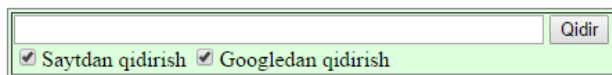
**Mavzu: HTML5ning formalar bilan ishlash: form, input. Ularning xususiyatlarini o'rganish.**

**Ishdan maqsad:** HTMLda formalar bilan ishlovchi teglar va ularning xususiyatlarini o'rganish.

### Nazariy qism

Web saytlarda formalar juda ko'p joyda ishlatiladi. Misol uchun, **registratsiyada, so'rovlarda, avtorizatsiyada, kommentariyalarda** va hokazo.

```
<html>
<head> <title>Form of Form</title></head>
<body>
<form name="forma" action="index.php" method="get">
<table border=1 bgcolor=#ddffdd align="center">
<tr>
<td align="left">
<input type="text" name="search" size=«50" value="Qidirish tizimi...">
<input type="submit" value="Qidir">
<br>
<input type="checkbox" name="site">Saytdan qidirish
<input type="checkbox" name="google">Googledan qidirish
</td>
</tr>
</table>
</form>
</body>
</html>
```



Bu kodlarda quyidagi teglar ishlatilgan:

1. **<form>...</form>** — veb sahifaga forma joylashtirish, forma foydalanuvchi va veb server bilan aloqani ta'minlab beradi. Agar forma elementlari(**button**, **checkbox**, **radiobutton**,...) kiritilmasa, bu teg brauzerga hech nima chiqarmaydi, uning atributlari:

**accept-charset** — serverga jo'natilayotgan axborot kodirovkasini o'rnatish.

```
<form accept-charset=«windows-1251">...</form>
```

**action** — forma orqali jo'natilayotgan ma'lumotlar qaysi sahifa orqali qayta ishlanishi kerakligi beriladi, biror adres ko'rsatiladi.

```
<form action=«index.php»>...</form>
```

formadagi barcha axborotlar «**index.php**» sahifasiga olib o'tilib, u yerda tahrirlanadi.

**autocomplete** — formani avtomat to'ldirishni boshqarish, brauzer keshida qandaydir ma'lumotlar bo'lsa, shular orqali forma to'latilishi mumkin, uning qiymati:**on**(yoqig'), **off**(o'chiq).

**method** — axborotlarni jo'natish turi.

**name** — formaga nom berish, bu nom skriptlar yozishda ishlatiladi.

**novalidate** — formaga kiritilgan ma'lumotlarni brouzer tomonidan tekshirish. Brouzer pochta adreslarni, url adreslarni tekshirishi mumkin.

**target** — jo'natilayotgan axborotni qayta ishlovchi veb sahifani ochilish turi, uning qiymatlari: `_blank`, `_self`, `_parent`, `_top`. «a» tegidagi «**target**» atributi bilan bir hil.

2. **<input>...</input>** — bu teg orqali forma elementlarini yaratish mumkin, ya'ni formani asosi hisoblanadi. Bu elementlarni yuqoridagi kodda keltirilgan

Bu tegning atributlari:

**type** — forma elementi turi, bu atribut formada har xil ko'rinishdagi elementlar hosil qilib beradi, ularni ba'zilarini misollarda ko'rish mumkin, ularning qiymatlari quyidagilar:

\***button** — tugma(knopka) hosil qilish.

\***checkbox** — bir necha tanlovni amalga oshirib beruvchi bayroqcha hosil qilish.

\***file** — serverga jo'natiladigan fayl adresini kiritish, fayl elementi.

\***hidden** — ko'rinmas element, sahifada ko'rinmaydi, lekin ko'p hollarda ishlatiladi.

\***image** — rasmi qator, rasmga bosilganda, axborotlar serverga jo'natiladi.

\***password** — oddiy matnli element, lekin yozuvlari ko'rinmaydi, parol yozish uchun.

\***radio** — bir necha tanlovalardan faqat bittasini tanlash imkonini beruvchi element.

\***reset** — forma qiymatlarini dastlabki holatga qaytarish.

\***submit** — formadagidagi axborotlarni serverga jo'natish.

\***text** — oddiy matn kiritish elementi.

**align** — tip rasm(**image**) bo'lganda, rasmni tekislash, qiymatlari: `bottom`, `left`, `middle`, `right`, `top`

**alt** — rasmi tugmada, rasm ochilmasa, uning o'rnida paydo bo'ladigan yozuv.

**autocomplete** — elementni avtomat to'ldirishni o'rnatish.

**border** — rasm atrofidagi ramka qalinligini o'rnatish.

**checked** — **checkbox** yoki **radio** uchun dastlabki holatni «belgilash» ko'rinishida namoyon qilish.

`<input type=«checkbox»>Messi`

`<input type=«checkbox» checked=«checked»>Rooney`

`<input type=«checkbox» >Neymar`

**disabled** — elementni blokirovka qilish, ya'ni dastlabki holatni o'zgartirib bo'lmaydi.

`<input type=«text» disabled value=«Yozolmaysan»>`

**form** — boshqa forma bilan identifikator orqali bog'lanish.

**formaction** — axborotlarni qayta ishlaydigan sahifa adresini berish(form tegi action atributi bilan bir hil).

**formmethod** — axborotlari qaysi metod orqali serverga jo‘natishni belgilash(Get, Post. Form tegi method atributi bilan bir hil).

**maxlength** — foydalanuvchi tomonidan maksimal kiritilishi mumkin bo‘lgan axborot o‘lchami(kiritilgan raqamgacha bo‘lgan simvollar).

**smultiple** — element tipi fayl bo‘lganda, bir paytda bir necha fayllar kiritilishi mumkinligini bildiradi.

**name** — forma elementiga takrorlanmas nom berish, bu nom axborot serverga jo‘natilayotganda ishlatiladi.

**readonly** — yozuvli elementda ishlatilsa, faqat yozuvni ko‘rish imkonini yaratadi, o‘zgartirib bo‘lmaydi.

**size** — yozuvli elementlarda(text, password) yozuv yozish mumkin bo‘lgan joyning uzunligini o‘rnatish.

**src** — tip rasm(type=qiimage) bo‘lgan elementni adresini ko‘rsatish.

**tabindex** — klaviaturadan nech marotaba «tab» tugmasi bosilsa, shu elementga o‘tish kerakligini o‘rnatish. Qiymat raqamlardan iborat bo‘ladi.

**value** — forma elementiga dastlabki qiymatni berish.

```
1 <html>
2 <head> <title>Form of Form</title><head>
3 <body>
4 <p align="center"><font color="blue" size="4" face="verdana">Registratsiya</p>
5 <form name="forma" action="index.php" method="post">
6 <table border=1 bgcolor="#ddffdd" align="center">
7 <tr>
8 <td align="left">
9 <font size="2" color="red" face="verdana">Login kiriting: </font>
10 <input type="text" name="search" size="25" value="user name"><br><br>
11 <font size="2" color="red" face="verdana">Parol kiriting: </font>
12 <input type="password" name="password" value="password" size="15">
13 <br>
14 <hr color="green">
15 <fieldset><legend>Shaxsiy ma'lumotlar</legend>
16 <font size="2" color="red" face="verdana">Jinsingiz?</font><br>
17 <input type="radio" name="club" value="male" checked="checked"> Erkak<br>
18 <input type="radio" name="club" value="female"> Ayol
19 <hr color="green">
20 <font size="2" color="red" face="verdana">Yoshingiz?</font><br>
21 <select name="age">
22 <option value="yigirma">20
23 <option value="yigirma bir">21
24 <option value="yigirma ikki" selected="selected">22
25 </select>
26 <hr color="green">
27 <font size="2" color="red" face="verdana">Kasbingiz?</font><br>
28 <select name="notebook" multiple size="2">
29 <option value="oshpaz">Oshpaz
30 <option value="prog">Dasurchi
31 <option value="admin">Administrator
32 <option value="ras">Rassom
33 <option value="sartosh">Sartarosh
34 <option value="taxi">Taksist
35 </select>
36 </fieldset>
37 <br>
38 <hr color="green">
39 <textarea rows="3" cols="45">
40 Katta hajmdagi matnlar textarea tegi ichida yoziladi
41 </textarea>
42 <center><input type="button" value="OK"></center>
43 </tr>
44 </table>
45 </form>
46 </body>
47 </html>
```



1. **<fieldset>...</fieldset>** — formadagi bir necha elementlarni guruhlash uchun ishlatiladi. guruhlangan elementlar chiziq bilan chegaralanadi. uning atributlari:

**disabled** — guruhning barcha elementlarini bloklash.

**form** — boshqa formalar bilan bog‘lanish(identifikator orqali).

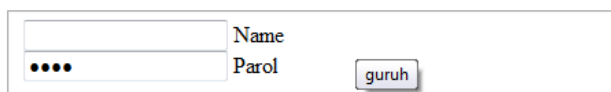
**title** — formadagi guruhga yordam sifatida yozuv chiqarish.

`<fieldset>`

`<input type=«text»> Name<br><input type=«Password» value=«akmx»> Parol<br>`

`<input type=«checkbox» checked=«checked»> Belgilash`

`</fieldset>`



2. **<legend>...</legend>** — «**fieldset**» orqali hosil qilingan guruhga nom berish, bu nom chizilgan chegara chizig‘ining tepa qismida paydo bo‘ladi(misolda «**shaxsiy ma’lumotlar**» degan nom).

Atributlari quyidagilar:

**accesskey** — klaviatura tugmalari orqali tez o‘tishni amalga oshiradi.

**align** — guruh nomini tekislash: left, right, center.

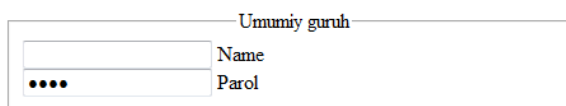
**title** — forma nomiga sichqoncha oborilsa, yordam sifatida yozuv chiqarish.

`<fieldset title=«guruh»><legend align=«center»>Umumiy guruh</legend>`

`<input type=«text»> Name<br>`

`<input type=«Password» value=«akmx»> Parol<br>`

`</fieldset>`



3. **<select>...</select>** — foydalanuvchi tanlashi uchun bir necha variantlarga ega bo‘lgan ro‘yxat hosil qilish. Bu teg «**option**» tegi bilan birga ishlatiladi. Bir yoki bir nechta variantni tanlash imkonini beradi(**size** atributi orqali o‘rnatiladi). Uning atributlari:

**accesskey** — klaviatura tugmalari orqali tez o‘tishni amalga oshiradi.

**disabled** — o‘zgartirib bo‘lmaydigan holga olib keladi, ya’ni blokirovka.

**form** — boshqa forma bilan identifikator orqali bog‘lanishni amalga oshirib beradi.

**multiple** — bir vaqtning o‘zida bir necha qiymatni tanlash imkonini beradi.

**name** — serverga jo‘natish uchun element nomini o‘rnatish.

**size** — qiymatlarni nech qator bo‘lib ekranga chiqarishini o‘rnatish, qolgan qiymatlar ko‘rinmas holda bo‘lib turadi va menyu ochilsa ko‘rinadi.

4. **<option>...</option>** — «select» orqali hosil qilingan ro'yxatdagi elementlarni yaratish.

Nechta «**option**» bo'lsa, ro'yxatda shuncha element bo'ladi. Uning atributlari:

**disabled** — ma'lum bir ro'yxat elementlarini bloklash.

**label** — ro'yxat elementiga metka qo'yish.

**selected** — brauzer yuklanganda shu atribut o'rnatilgan element tanlangan bo'lib chiqadi.

**value** — serverga shu nom orqali element qiymati ketadi.

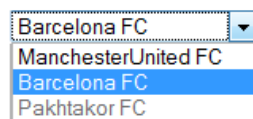
```
<select name="club" size="1">
```

```
<option value="mufc">ManchesterUnited FC
```

```
<option value="fcb" selected>Barcelona FC
```

```
<option value="pfc" disabled>Pakhtakor FC
```

```
</select>
```



5. **<textarea>...</textarea>** — forma elementi bo'lib, katta hajmdagi matnlarni kiritish uchun ishlatiladi. Uning atributlari:

**accesskey** — klaviatura tugmalari orqali tez o'tishni amalga oshiradi.

**cols** — matn yozish uchun ajratilgan joy uzunligi(raqamda beriladi, nechta simvol kiritilishi).

**disabled** — elementni blokirovka qilish.

**form** — boshqa forma bilan identifikator orqali bog'lanishni amalga oshirib beradi.

**maxlength** — foydalanuvchi tomonidan kiritilishi mumkin bo'lgan maksimal simvollar soni.

**name** — elementning nomi.

**readonly** — element ichidagi yozuvni faqat o'qish mumkin bo'lgan holga keltirish(o'zgartirib bo'lmaydi).

**rows** — matn yozish uchun ajratilgan joy balandligi(qatorlar soni), shu sondan so'ng pastga tushish uchun aylantirgich hosil bo'ladi(prokrutka).

**tabindex** — klaviaturadan nechinchi marotaba «tab» tugmasi bosilsa, shu elementga o'tishni.

**wrap** — yozilgan matnni keyingi qatorga o'tkazish qoidasini o'rnatish, uning qiymatlari

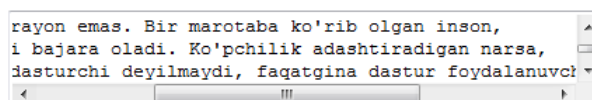
\*soft — matn uzunligi bitta qatorga sig'masa, avtomat keyingi qatorga o'tadi.

\*hard — «cols» da berilgan qiymatdan so'ng uzun matnlar keyingi qatorga o'tadi.

\*off — matn uzunligi tugamaguncha birinchi qator ham tugamaydi(surilgich chiqadi).

```
<textarea rows="3" cols="50" readonly wrap="off">
```

```
</textarea>
```



Yuqoridagi kodning natijasi quyidagi holda bo‘ladi.

Form of Form

### Registratsiya

Login kiriting: sd

Parol kiriting: ●●●●●●●●

Shaxsiy ma'lumotlar

Jinsingiz?

☒ Erkak

☐ Ayol

Yoshingiz?

22 ▾

Kasbingiz?

Oshpaz ▲

Dasurchi ▼

Katta hajmdagi matnlar textarea tegi ichida  
yoziladi

OK

**Tajriba ishi №5**  
**Mavzu: CSSda sahifaga stil berish. Selektorlar.**

**Ishdan maqsad:** CSS da selektorlarni o'rganish. Turli stil xossalarini o'rganish.

**Nazariy qism**

CSS – bu stillar bilan ishlay oladigan kaskadli stillar majmuasidir. Web sahifalarga har xil stillar berish uchun ishlatiladi. HTML xujjat ichida teglar bilan birga foydalaniladi. CSS kodlarini kompilyastiya qilish uchun qandaydir kompillyatorlar, qandaydir dasturlar, va bu kod tushuna oladigan qandaydir redaktorlar kerak bo'lmaydi. Bu kodlarni HTML singari web brauzerning o'zi kompilyastiya qiladi va natijani chiqarib beradi.

CSS ni HTML xujjati ichida ishlatilishini birin ketin ko'rib o'tamiz:

Web sahifada ishlatiladigan CSS kodlari orqali hosil qilinadigan stillarni hammasini bitta faylga yozib, uni serverda saqlaymiz. So'ng, sahifadan, serverdagi saqlab qo'yilgan faylga murojaat qilib, kerakli stilni olamiz. Bu usul katta hajmdagi web sahifalar yaratishda foydali hisoblanadi. Faylga murojaat, <head>tegi ichida amalga oshiriladi va quyidagi ko'rinishga ega bo'ladi.

<LINK REL=STYLESHEET TYPE=»text/css» HREF=»URL»>

<link> tegi sahifa biror faylga murojaatni amalga oshirmoqda degani, REL – fayl qandaydir stillardan iborat degani, TYPE — stillar css kodlaridan iborat va fayl joylashgan manzil yoziladi.

CSS kodlar veb sahifa ichidagi <head> tegi orasida beriladi. Bunda yaratilgan stillar shu sahifaga tegishli bo'ladi, bu degani faqat shu sahifada ishlatilish mumkindir.

<head>< STYLE TYPE=»text/css»>

<!--

A {text-decoration:none; }

-->

</--STYLE></head>

Navbatdagi turi, bu har bir element uchun alohida teg ichida stil berib chiqishdan iborat. Misol uchun, "p" tegi ichida biror matn yozilsa, shu matn uchun teg ichida alohida stil beriladi va bu stil shu teg uchun xususiy bo'ladi. Bu usuldan foydalanishni maslahat bermiman, chunki bu usul kodlarni chalkash va tushunarsiz bo'lishiga olib keladi. Shoshilinch vaziyatlarda yoki stil qay tartibda namoyon bo'lishni ko'rib olish uchun bu usuldan foydalanishni maslahat beraman.

Undan tashqari <style> tegi orqali css faylni import qilish mumkin bo'ladi.

@import: url(mystyles.css);

CSS stillarini ishlatishda, quyidagi guruhlash usullaridan foydalanish, tartibli web sahifa tuzishga olib keladi:

— Kodlarni ixchamlash uchun, xar hil elementlar uchun bir hil tipdagi stillarni guruhlab ishlatish maqsadga muvofiqdir.

H1 {font-family: Verdana}

H2 {font-family: Verdana}

Yuqoridagi stilni guruhlab quyidagicha yozamiz.

H1, H2 {font-family: Verdana}

— Guruhlashda bir hil element uchun bir necha stillarni birlashtirish mumkin.

H2 {font-weight: bold}

H2 {font-size: 14pt}

H2 {font-family: Verdana}

Barcha stillarni bittaga yig'amiz.

H2 {font-weight: bold; font-size: 14pt; font-family: Verdana;}

— Ba'zi stillar bitta stil ichida berilishi ham mumkin.

H2 {font: bold 14pt Verdana}

CSS stillari bilan ishlaganda, element ichidagi elementlar ham o'zidan yuqori turgan element stilini qabul qiladi, ya'ni o'zidan oldingi element "ota" vazifasini bajaradi va o'z "farzandlari"ga ham o'zida borini beradi. Misol uchun, <p> tegi ichidagi matn ko'k rangda yozilishi lozim bo'lsa(P {color: blue}), <p> tegi ichidagi <em> tegiga tegishli bo'lgan matn ham ko'k rangda yoziladi.

Ba'zi stillar faqat yakka tartibda ishlaydi, ya'ni ichki teglarga stillari o'tmaydi(misol uchun, background), shuning uchun barcha stillarni birma bir ishlatib, o'rganib olish lozim. <body> tegiga stil berib, dastlabki o'rnatilish(po umolchaniyu) jarayonini hosil qilish mumkin. Shunda biror elementga stil berish esdan chiqib qolsa, <body> tegida ta'luqli stil esdan chiqqan elementga o'rnatiladi.

BODY {color: green;

font-family: «Verdana»;

background: url(joke.gif) white; }

CSS da asosiy fon va rang uchun javob beradigan stil xususiyatlari quyidagilardan iborat:

— color;

— background-color;

— background-image;

— background-repeat;

— background-attachment;

— background-position;

— background.

Bu stil xususiyatlarini birma-bir ko‘rib chiqamiz.

Color — u yoki bu elementga rang berish uchun ishlatiladi, misol uchun, hamma paragraf teglari qizil rangda, barcha h1 teglar qora rangda bo‘lish kerak bo‘lsa, quyidagicha stil beriladi.

```
1 <html>
2 <head>
3 <title>Rang_1</title>
4 <style type="text/css">
5   p{color:red}
6   h1{color:black}
7 </style>
8 </head>
9 <body>
10  <h1>Qora</h1>
11  <p>Qizil</p>
12 </body>
13 </html>
```

Qayerda bu teglar ishlatilsa, shu tegga mos holda stillar ham ishlaydi, natijani ko‘ramiz:

**Qora**

Qizil

Background-color — biror element yoki butun veb sahifa foniga rang berish. HTML dagi «bgcolor»dan farqli ravishda, bu stil xususiyati istalgan element foniga rang berish mumkin: havolalarga, paragraflarga, jadvallarga,...

Misol ko‘ramiz, butun sahifa foniga va barcha H3 teglari foniga rang beramiz.

```
1 <html>
2 <head>
3 <title>Rang_2</title>
4 <style type="text/css">
5   body{background-color:gray}
6   h3{color:orange}
7 </style>
8 </head>
9 <body>
10  <h3>Fongini rangini qara, daxshatu a!</h3>
11 </body>
12 </html>
```

Rang berishda odatagidek rang nomi(ingliz tilida) yoki 16 lik sanoq tizimda berish lozim(16lik tizimdagi rangi aniqlash), endi natijani ko‘ramiz:

## Fongini rangini qara, daxshatu a!

Background-image — sahifa foni sifatida biror rasmni qo'yish, nafaqat sahifa boshqa elementlarga ham qo'yish mumkin, faqat yaxshi chiqmasligi ehtimoli katta. Misol ko'ramiz,

```
1 <html>
2 <head>
3 <title>Rang_3</title>
4 <style type="text/css">
5   body{
6     background-image: url(bum.jpg);
7   }
8 </style>
9 </head>
10 <body>
11 </body>
12 </html>
```



E'tibor bergan bo'lsangiz, rasmni ko'rsatish HTMLga nisbatan boshqacharoq. «url» kalit so'zi yozilib, so'ng rasm adresi berilmoqda, bu holda rasm sahifa fayli saqlangan papka bilan bir joyda joylashgan, agar boshqa joyda bo'lsa, (image/bum.jpg) ko'rinishida(rasmning to'liq adresi) berish kerak.

Background-repeat — qaytarilishni amalga oshirish. Agar yuqoridagi misolga e'tibor bergan bo'lsangiz, kichik rasmlar yuqori chap burchak bo'lab qaytarilib takrorlanmoqda va shu tariqa butun veb sahifa foni rasimga to'lmoqda. Background-repeat shu qaytarilishni boshqara oladi.

Bu stil xususiyatining quyidagi qiymatlari mavjud:

Background-repeat: repeat-x ; — gorizontal bo'yicha qaytarilish.

Background-repeat: repeat-y ; — vertikal bo'yicha qaytarilish.

Background-repeat: repeat ; — vertikal va gorizontal bo'yicha qaytarilish.

Background-repeat: no-repeat ; — qaytarilishni amalga oshmaydi.

Bunday xususiyatlar HTML gippermatn tilida mavjud emas. Keling misol ko'ramiz,

```
1 BODY {  
2   background-image : url(bum.jpg) ;  
3   background-repeat: repeat-x;  
4 }
```

Fon sifatida tanlangan rasm, faqat gorizontal bo'yicha takrorlanadi.



Background-attachment — agar fon sifatida rasm bo'lsa, bu stil xususiyati o'rnatiladi. Bunda, agar sahifada o'tkazgich(prokrutka)bo'lsa, fon rasmi ham o'tish kerak yoki fon rasmi sahifaga qotirib qo'yilishi kerakligi ko'rsatiladi. Qiymatlari:

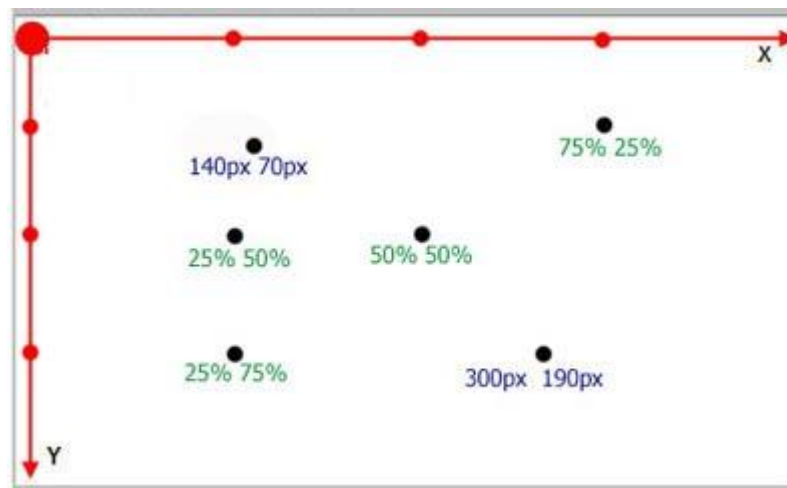
scroll — fon sifatida o'rnatilgan rasm, veb sahifa oynasi bilan birga o'tadi(prokrutka).

fixed— fon sifatida o'rnatilgan rasm, veb sahifa oynasiga qotirib qo'yilgan bo'ladi va o'zgarmaydi.

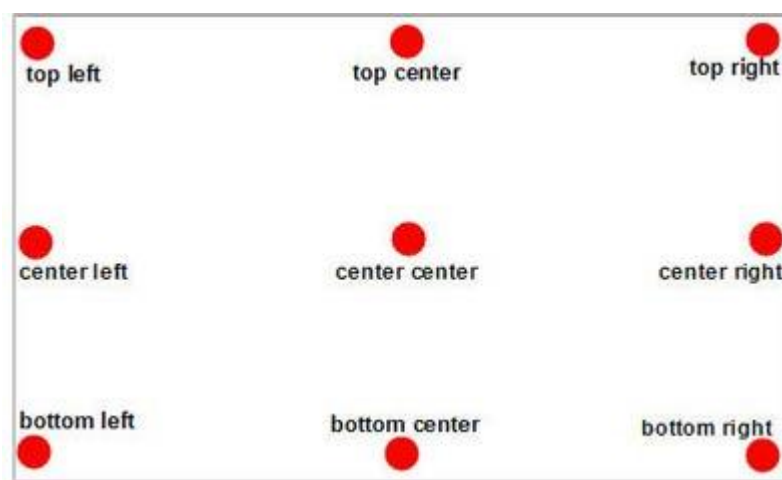
```
1 BODY {  
2   background-image : url(bum.png) ;  
3   background-repeat: no-repeat;  
4   background-attachment: scroll ;  
5 }
```

Background-position — fon sifatida tanlangan rasmni pozitsiyasini o'rnatadi, ya'ni fon rasmi aynan qayerda joylashishini aniqlashtirish, qiymat foiz yoki uzunlik o'lchovida beriladi. Boshlanish, chap yuqori tomondan boshlanadi. Quyidagi rasmda pozitsiya haqida yaqqol ko'rsatilgan.





Dastlab, gorizontaal bo'yicha qiymat beriladi, keyin vertikal bo'yicha. Qiymat brauzer o'lchamiga qarab piksel yoki foizda beriladi. Undan tashqari kalit so'zlar orqali ham pozitsiyani berish mumkin.



Keling endi misol ko'ramiz,

```

1  <html>
2  <head>
3  <title>Rang_4</title>
4  <style type="text/css">
5  BODY {
6  background-image : url(bum.jpg);
7  background-repeat: no-repeat;
8  background-position: center center;
9  }
10 </style>
11 </head>
12 <body>
13 </body>
14 </html>

```



Yana bir misol,

```
1 BODY {  
2   background-image : url(bum.jpg) ;  
3   background-repeat: no-repeat;  
4   background-position: 300px 500px ;  
5 }
```

Background — bu umumiy, hamma xarakteristikalarini ishlatish uchun foydalaniladi, ya'ni background-color, background-image, background-repeat, background-attachment, background-position stil xususiyatlarini bitta shu stil orqali ishlatish mumkin, ya'ni

```
1 BODY {  
2   background-color:#ffee8c ;  
3   background-image : url(bum.png) ;  
4   background-repeat: no-repeat;  
5   background-attachment: fixed;  
6   background-position: top right;  
7 }
```

Shu uzun kodni, bitta «background» orqali yozish mumkin:

```
1 BODY {  
2   background: #ffee8c url(bum.png) no-repeat fixed top right ;  
3 }
```

Agar biror xususiyatini kiritmasak, doimiy qiymatni qabul qilib, ishlatadi.

## Tajriba ishi №6

### Twitter bootstrap. Frameworklarda ishlash. Grid system tushunchasi.

**Ishdan maqsad:** Twitter bootstrap CSS frameworki haqida bilimlarga ega bo'lish.

#### Nazariy qism

Bootstrapga twitterning asoschisi asos solgan va shuning uchun ham nomi twitter bootstrap deb nomlangan. Bootstrap bu soda va oson quriladigan HTML, CSS va Javascript framework hisoblanadi. Zamonaviy, standartlashgan, qulay va tez ishlaydigan saytlarni qurishimiz uchun yordam beradi.

Bootstrapning qulayliklari:

1. Qisqa vaqt ichida sayni va shablonlarni noldan, osongina qurish.
2. Adaptiv saytlarni va shablonlarni qurishda yengillik.
3. Aloqalarni tashkil qilishda qo'shimcha kutubxonalarning ko'pligi.
4. Tayyor CSS fayllar

Bu bootstrapning bizga kerak bo'ladigan, asosiy qulayliklari edi.

Bootstrapning asosiy qurilmalari:

Setkalar- bu Grid system deyiladi

Shablonlar- feksirovanniy yoki dokumentning rezina shabloni

Tipografika- shriftlar haqida ma'lumot

Media- rasmlar va videolarni boshqarish uchun

Tablitsalar- tablitsalarni oformleniya qilish, ularni saralash uchun

Formalar- faqatgina formalarni oformit qilish uchun va ular ustida ish olib boorish uchun

Navigatsiya- Navigation bar yoki navbar deyiladi. Saytlarning yuqori qismidagi instrumentlar paneli xisoblanadi. U orqali web saytning boshqa sahifalarini ko'rishimiz mumkin.

Alertlar- ogohlantiruvchi vosita

Bootstrapning fayllar strukturasi quyidagicha:

Bootstrap/

css/

bootstrap.css

bootstrap.min.css

js/

bootstrap.js

bootstrap.min.js

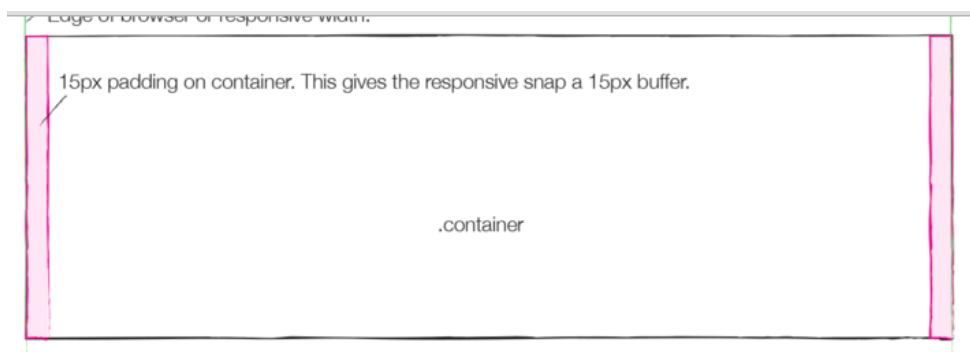
img/

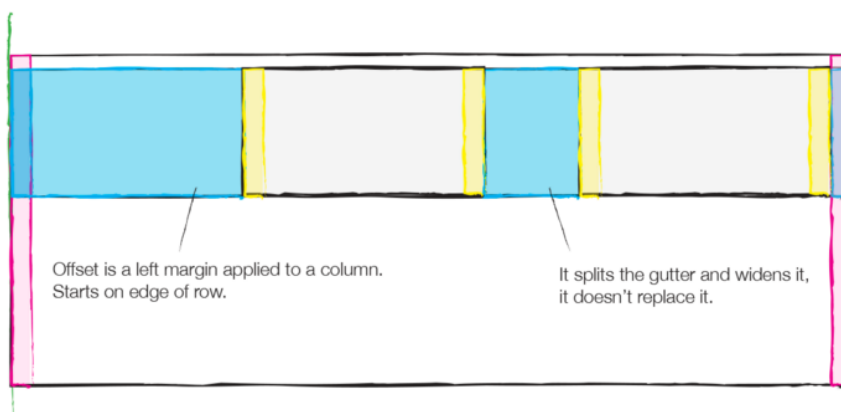
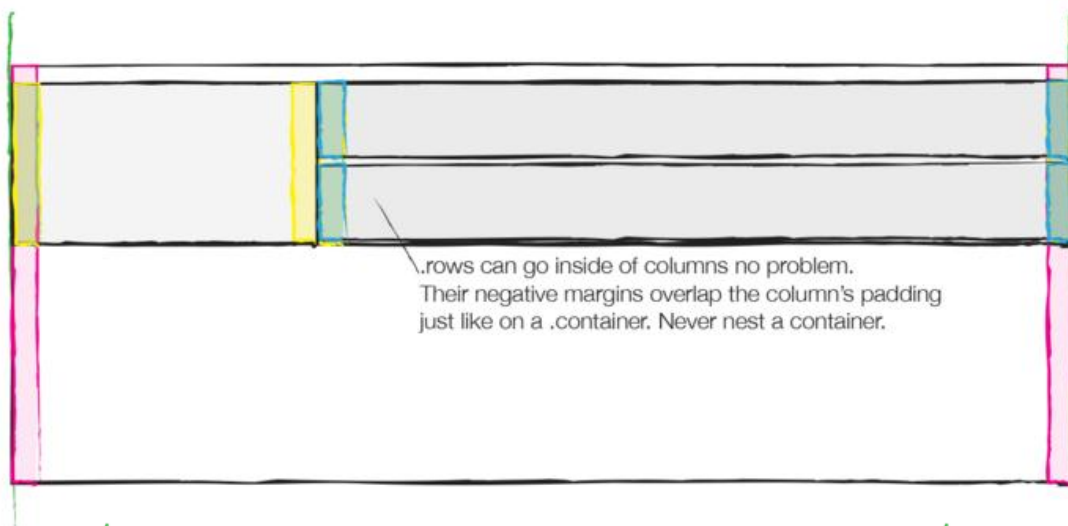
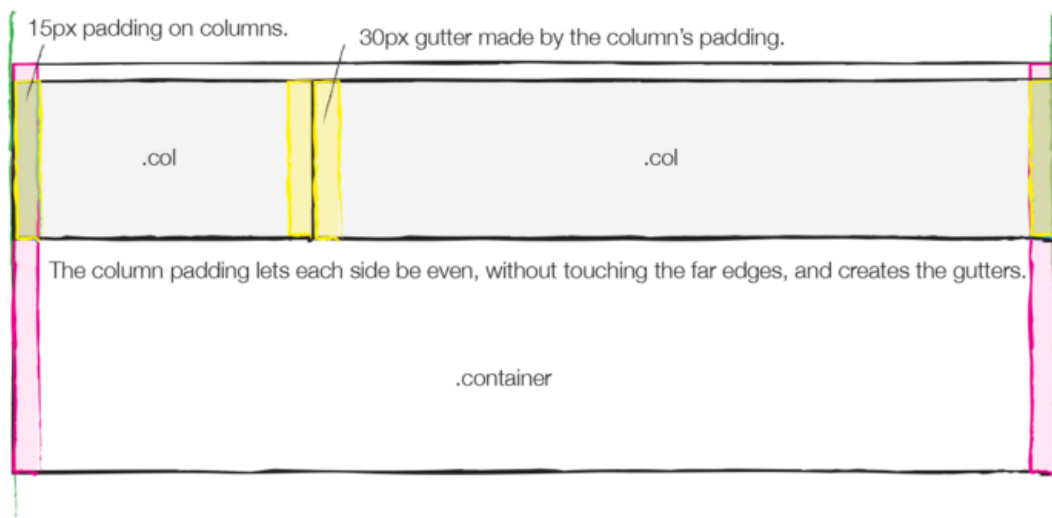
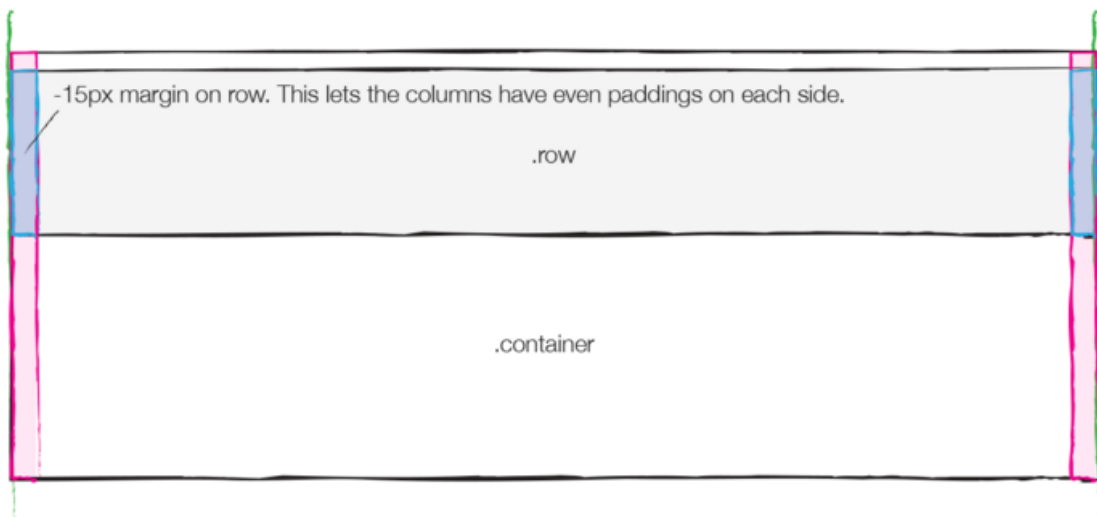
glyphicon-halflings.png

glyphicon-halflings.whi

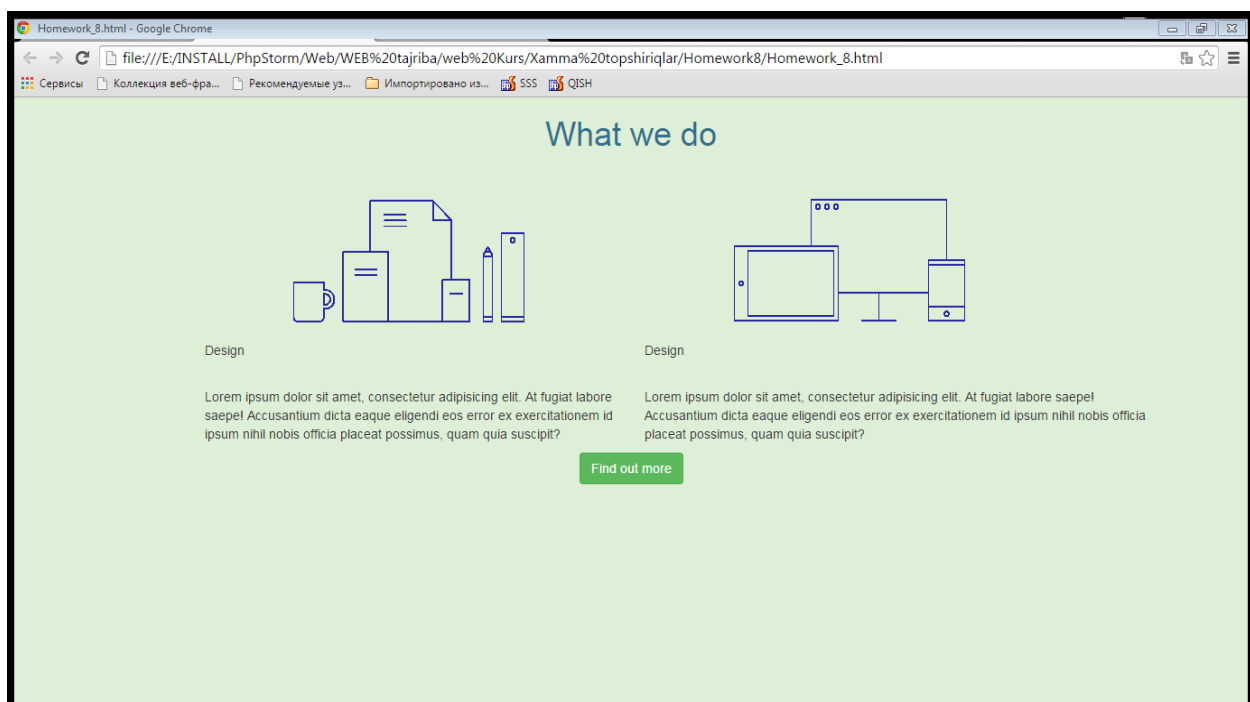
README.md

Grid systemda container, row, col-lg-2, col-md-6, col-sm-3, col-xs-4, col-offcet-lg-1 va shu kabi boshqa asosiy tushunchalar bor. Ular saytlarning standart holda bo'lishini taminlab beradi.





```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title></title>
5     <link rel="stylesheet" href="resources/bootstrap/css/bootstrap.min.css"/>
6 </head>
7 <body style="background-color: #deefd7">
8 <div class="container">
9     <h1 class="text-center text-info">What we do</h1><br/><br/>
10
11     <div class="row">
12         <div class="col-md-4 col-md-offset-2">
13             <br/><br/>
14         </div>
15
16         <div class="col-md-4 col-md-offset-1">
17             <br/><br/>
18
19         </div>
20     </div>
21     <div class="row">
22         <div class="col-md-5 col-md-offset-1 ">
23             <p>Design</p><br/>
24             <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. At fugiat labore saepe! Accusantium dicta eaque eligendi eos error ex exercitationem id ipsum nihil nobis officia placeat possimus, quam quia suscipit?</p>
25         </div>
26         <div class="col-md-6 ">
27             <p>Design</p><br/>
28             <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. At fugiat labore saepe! Accusantium dicta eaque eligendi eos error ex exercitationem id ipsum nihil nobis officia placeat possimus, quam quia suscipit?</p>
29         </div>
30     </div>
31     <button class="btn btn-success center-block">Find out more</button>
32
33 </div>
34 </body>
35 </html>
```



## Tajriba ishi №7

### **Mavzu: JavaScript tilidan foydalanib veb sahifaga interaktivlik qo'shish. Funksiyalar yaratish.**

**Ishdan maqsad:** Javascript tili asosiy konstruksiyalarini o'rganish.

#### **Nazariy qism**

Javascript – bu Webni gipermatnli sahifalarini klient tomonida ko'rish tsenariyalarini boshqarish tili. Yanada aniqroq aytadigan bo'lsa, Javascript – bu nafaqat klient tomonidagi dasturlash tili. Liveware Javascript tilining avlodi bo'lib, Netscape serveri tomonida ishlovchi vosita bo'ladi. Ammo Javascript tilini mashhur qilgan narsa bu klient tomonida dasturlashdir. Javascriptning asosiy vazifasi – HTML-konteynerlar atributlarining qiymatlarini va ko'rsatuvchi muhitining hossalirini HTML-sarlavxalarni ko'rish jarayonida foydalanuvchi tomonidan o'zgartirish imkoniyatlarida, boshqacha aytganda ularni dinamik sarlavxalar qilish (DHTML). Yana shuni aytish joizki, sarlavxalar qayta yuklanmaydi. Amalda buni, masalan, quydagicha ifodalash mumkin, sarlavxaning fonini rangini yoki xujjatdagi rasmni o'zgartirish, yangi oyna ochish yoki o'g'xantirish oynasini chiqarish. “JavaScript” nomi Netscape kompaniyasining hususiy maxsuloti. Microsoft tomonidan amalga oshirilgan til rasman Jscript deb nomlanadi. Jscript versiyalari Javascriptning mos versiyalari bilan mos keladi (aniqroq qilib aytganda oxirigacha emas). Javascript – ECMA (European Computer Manufacturers Association – Evropa Kompyuter Ishlab Chiqaruvchilar Assotsiyatsiyasi) tomonidan standartlashtirilgan. Mos standartlar quydagicha nomlanadi ECMA-262 va ISO-16262. Ushbu standartlar bilan Javascript 1.1 ga taqriban ekvivalent ECMAScript tili aniqlanadi. Eslatish joizki, bugungi kunda Javascript ning hamma versiyalari ham ECMA standartlariga mos kelavermaydi. Mazkur kurs yoki qo'llanmada barcha hollarda biz Javascript nomidan foydalanamiz. Javascriptning asosiy hususiyatlari. Javascript – bu Internet uchun katta bo'lmagan klient va server ilovalarni yaratishga mo'ljallangan nisbatan oddiy ob'ektga yo'naltirilgan til. Javascripttilida tuzilgan dasturlar HTML-xujjatning ichiga joylashtirilib ular bilan birga uzatiladi. Kurish dasturlari (brauzerlar – browser ingliz suzidan) Netscape Navigator va Microsoft Internet Explorer xujjat matniga joylashtirilgan dasturlarni (Scriptkod) uzatishadi va bojarishadi. Shunday qilib, Javascript – interpretatorli dasturlash tili xisoblanadi. Javascriptda tuzilgan dasturlarga foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni tekshirayotgan yoki xujjatni ochganda yoki yopganda biror bir amallarni bajaruvchi dasturlar misol bo'lishi mumkin. JavaScript da yaratilgan dasturlarga misol sifatida foydalanuvchi tomonidan kiritilgan ma'lumotlarni tekshiruvchi, dokumentni ochish yoki yopish vaqtida qandaydir amallarni bajaruvchi dasturlarni keltirish mumkin. Bunday dasturlar foydalanuvchi tomonidan berilgan kursatmalarga – sichqoncha tugmachasini bosilishiga, ma'lumotlarni ekran orqali kiritishiga yoki sichqonchani sahifa buylab siljilishiga kura ish bajaradi. Bundan tashqari JavaScript dagi dasturlar brauzerning uzini va dokumentning atributlarini ham boshqarishi mumkin. JavaScript dasturlash tili sintaktik jihatdan Java dasturlash tiliga, ob'ektni modellashni istisno qilgan holda, o'hshab ketsada, lekin ma'lumotlarni statik tiplari va qat'iy tiplashtirish kabi hususiyatlarga ega bulmaydi. JavaScript da Java dasturlash tilidan farq qilib, sinf (klass) tushunchasi bu tilning asosiy sintaktik qurilmasi hisoblanmaydi. Bunday asos sifatida foydalanilayotgan tizim tomonidan qullab-quvvatlanayotgan, oldindan aniqlangan ma'lumot tiplari: sonli, mantiqiy va satrli; mustaqil ham bulishi, ob'ektning metodi (JavaScriptda metod tushunchasi funktsiya/qism-dastur ning uzi) sifatida ham ishlatilishi mumkin bulgan funktsiyalar; katta sondagi uz hossalriga va metodlariga ega bulgan oldindan aniqlangan ob'ektlardan iborat ob'ektni model va yana dastur ichida foydalanuvchi tomonidan yangi ob'ektlarni berish qoidalari hisoblanadi. JavaScript da dasturlar yaratish uchun hech qanday qo'shimcha vositalar kerak bulmaydi – faqatgina tegishli versiyadagi JavaScript qullanishi mumkin bulgan brauzer va DHTML-dokumentlarni yaratishga imkon beruvchi matn muharriri kerak bo'ladi. JavaScript dagi dastur bevosita HTML –dokumentlarni ichiga joylashtirilganligi uchun dastur natijasini dokumentni brauzer yordamida kurish orqali tekshirish mumkin va kerakli

holda uzgartirishlar kiritilishi mumkin. JavaScript dasturlash tilining imkoniyatlari. Uning yordamida HTML –dokumentlarning ko‘rinishi va tuzilishini dinamik ravishda boshqarish mumkin. Ekranda tasvirlanayotgan dokumentga brauzer tomonidan yuklangan dokumentning sintaktik tahlil qilish jarayonida istalgan HTML-kodlarni joylashtirish mumkin. “Dokument” ob’ekti yordamida foydalanuvchining oldingi bajargan amallari yoki boshqa bir faktorlarga kura yangi dokumentlarni avtomatik hosil qilish mumkin. JavaScript yordamida brauzer ishini boshqarish mumkin. Masalan, Window ob’ekti suzib yuruvchi oynalarni ekranga chiqarish, brauzerning yangi oynalarini yaratish, ochish va yopish, oynalarning yugurdagi va o‘lchamlarining rejimlarini o‘rnatish va hokazolarni imkoniyatini beruvchi metodlarga ega. JavaScript dokumentdagi ma’lumotlar bilan bog‘lanish imkoniyatini beradi. Document ob’ekti va undagi mavjud ob’ektlar dasturlarga HTML-dokumentlarning qismlarini o‘qish va bazida ular bilan bog‘lanish imkoniyatini beradi. Matnning o‘zini o‘qish mumkin emas, lekin masalan berilgan dokumentdagi gipermatnli o‘tishlar ro‘yhatini olish mumkin. Hozirgi vaqtda Form ob’ekti va undagi mavjud bo‘lishi mumkin bo‘lgan ob’ektlar: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text va Textarealar dokumentdagi ma’lumotlar bilan bo‘g‘lanish uchun keng imkoniyatlar beradi. JavaScript foydalanuvchi bilan aloqa qilishga imkon beradi. Bu tilning eng muhim hususiyati unda amalga oshirilgan hodisalarni qayta ishlashni aniqlash imkoniyati – ma’lum bir hodisaning (odatda foydalanuvchi tomonidan bajarilgan amal) ro‘y berish vaqtida bajariladigan dastur kodining ixtiyoriy qismi hisoblanadi. JavaScript hodisalarni qayta ishlovchi sifatida ixtiyoriy yangi oldindan berilgan funktsiyalardan foydalanish imkoniyatini beradi. Masalan, foydalanuvchi sichqoncha ko‘rsatkichini gipermatnli o‘tishlar ustiga keltirsa, holatlar satrida mahsus habarni chiqaruvchi yoki ma’lum bir amalni bajarishni tasdiqlashni so‘rovchi dialogli oynani ekranga chiqaruvchi yoki foydalanuvchi tomonidan kiritilgan qiymatlarni tekshiruvchi va hatolik yuz bergan holda kerakli kursatmalarni berib, tug‘ri qiymatni kiritishni so‘rovchi dasturlarni yaratish mumkin. JavaScript ixtiyoriy matematik hisoblashlarni bajarish imkoniyatini beradi. Bundan tashqari bu tilda vaqt va sanalarning qiymatlari bilan ishlovchi yuqori darajada rivojlangan vositalar mavjud. JavaScript CGI-dasturlarga va Perl dasturlash tiliga va to‘diruvchi sifatida ayrim hollarda Java tiliga muqobil til sifatida yaratilgan. Har bir boshlovchi dasturchining asosiy savoli: “Dasturlar qanday tuziladi va bajariladi?”. Bu savolga iloji boricha soddaroq, lekin JavaScript-kodlarini qo‘llanilishining barcha usullarini unutmagan holda javob berishga harakat qilamiz. Birinchidan, JavaScript-kodlari brauzer tomonidan bajariladi. Unda mahsus JavaScript interpretatori mavjud. Unga kura programmaning bajarilishi interpretator tomonidan boshqaruvni qachon va qay tarzda olishiga bog‘liq bo‘ladi. Bu esa, o‘z navbatida kodning funktsiyaviy qo‘llanilishiga bo‘g‘liq bo‘ladi. Umuman olganda JavaScript ning funktsional qo‘llanilishining 4 hil usulini ajratib ko‘rsatish mumkin:

1. gipermatnli o‘tish (URL sxemasi);
2. hodisalarni qayta ishlash (handler);
3. o‘rniga qo‘yish(entity)
4. qo‘yish (SCRIPT konteyneri).

JavaScript bo‘yicha o‘quv qo‘llanmalarida JavaScript ni qo‘llashning bayoni odatda SCRIPT konteyneridan boshlanadi. Lekin dasturlash nuqtai nazaridan bu unchalik ham to‘g‘ri emas, chunki bunday tartib asosiy savol: “JavaScript-kodi boshqaruvni qanday oladi?” ga javob bermaydi. Ya’ni JavaScript da yozilgan va HTML-dokumentning ichiga joylashtirilgan dastur qanday tarzda chaqiriladi va bajariladi. HTML-sahifa muallifining kasbi va uning dasturlash asoslaridan habardarligining darajasiga qarab JavaScript ni o‘zlashtirishga kirishishni bir necha hil variantlari mavjud. Agar siz klassik tillar (C, Fortran, Pascal va h.) bo‘yicha dasturlovchi bo‘lsangiz, u holda dokument matni ichida dasturlashdan boshlagan ma’qul, agar siz Windows sistemasida dasturlashga o‘rgangan bo‘lsangiz, u holda hodisalarni qayta ishlashni dasturlashdan boshlaganingiz ma’qul, agar siz faqat HTML bo‘yichagina tajribaga ega bo‘lsangiz yoki anchadan beri dasturlash bilan shug‘ullanmayotgan bo‘lsangiz, u holda gipermatnli o‘tishlarni dasturlashdan boshlaganingiz ma’qul.

URL-sxemali JavaScript. URL sxemasi (Uniform Resource Locator)- bu Web-texnologiyalarning

asosiy elementlaridan biri. Web dagi har bir informatsion resurs uzining o'ziga hos URLiga ega bo'ladi. URL A konteynerining HREF atributida, IMG konteynerining SRC atributida, FORM konteynerining ACTION atributida va h.larda ko'rsatiladi. Barcha URL lar resursga ruhsatning protokoliga bog'liq bo'lgan ruhsat sxemalariga bo'linadi, masalan, FTP-arxiviga kirish uchun ftp sxemasi, Gopher-arxiviga kirish uchun gopher sxemasi, elektron maktublarni jo'natish uchun smtp sxemasi qo'llaniladi. Sxemaning tipi URLning birinchi komponentasiga ko'ra aniqlanadi: http://directory/page.html

Bu holatda URL http bilan boshlanayapti- mana shu kirish sxemasini aniqlashdir (sxema http). Gipermatnli sistemalar uchun dasturlash tillarining asosiy vazifasi gipermatnli o'tishlarni dasturlashdir. Bu shuni bildiradiki, u yoki bu gipermatnli o'tishlarni tanlashda gipermatnli o'tishni amalga oshiruvchi dastur chaqiriladi. Web-texnologiyalarida standart dastur sifatida sahifani yuklash dasturi hisoblanadi. HTTP protokoli bo'yicha standart o'tishni JavaScript da dasturlangan o'tishdan farq qilish uchun yaratuvchilar yangi URL sxemasi – JavaScript ni kiritishdi:

```
<A HREF=«JavaScript:JavaScript_kod">...</A>
```

```
<IMG SRC=«JavaScript:JavaScript_kod">
```

Bu holatda "JavaScript\_kod" matni birinchi holatda gipermatnli o'tishni tanlanganda chaqiriladigan va ikkinchi holatda rasmni yuklashda chaqiriladigan JavaScript da yaratilgan dasturiy-qayta ishlovchilarni bildiradi.

Masalan, Vnimanie!!! gipermatnli o'tishga keltirilgan holda ogohlantiruvchi oynani chiqarish mumkin:

```
<A HREF=«JavaScript: alert('Vnimanie!!!');"> Vnimanie!!!</A>
```



Formadagi submit tipidagi tugmachani bosish orqali shu formadagi matnli maydonni to'ldirish mumkin:

```
<FORM NAMEqf METHODqpost
```

```
ACTION=«JavaScript: window.document.f.i.VALUEq' Click tugmani bosing';void(0);"
```

```
<TABLE BORDERq0>
```

```
<TR>
```

```
<TD><INPUT NAMEqi></TD>
```

```
<TD><INPUT TYPEqsubmit VALUEqClick></TD>
```

```
<TD><INPUT TYPEqreset VALUEqReset></TD>
```

```
</TABLE>
```



</FORM>

URLda murakkab dasturlar va funktsiya chaqirilishlarini joylashtirish mumkin. Faqatgina shuni yodda tutish kerakki, JavaScript sxemasi hamma brauzerlarda ham ishlamaydi, faqatgina Netscape Navigator va Internet Explorer larning to'rtinchi versiyalaridan boshlab ishlaydi. Shunday qilib gipermatnli o'tishlarni dasturlashda interpretator boshqaruvni foydalanuvchi sichqoncha tugmasini gipermatnli o'tishga "bosgandan" keyingina oladi.

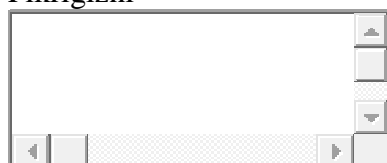
O`qish: 752

**Fikringiz uchun rahmat:**

**Ma`lumot qoldirish:**

Ismingiz:

Fikrigizni



qoldiring:

Rasmdagi raqamni kiriting

**3+2=**

Tasdiqlash

## Tajriba ishi №8

### Mavzu: JQuery kutubxonasi. Selektorlar. Hodisalar.

**Ishdan maqsad:** JQuery kutubxonasi haqida tushunchaga ega bo'lish. Selektor va hodisalarni o'rganish.

#### Nazariy qism

JQuery – bu tez, kichik va boy JavaScript kutubxona. U orqali HTML hujjat ob'yekt modeli manipulyatsiyalari, hodisalarni qayta ishlash, animatsiyalar bajarish, AJAX so'rovlari bilan ishlashni osonlashtirish chun mo'ljallangan.

Misollar:

#### 1. Yashirish va ko'rsatish.

```
<html>

<head>

  <link rel="stylesheet" href="bootstrap.min.css"/>

  <script src="jquery.min.js"></script>

<script>

  $(document).ready(function(){

    $("#hide").click(function(){

      $("p").hide();

    });

    $("#show").click(function(){

      $("p").show();

    });

  });

</script>

</head>

<body>

<div class="container well">

<p class="text-center">If you click on the "Hide" button,I will disappear.</p>

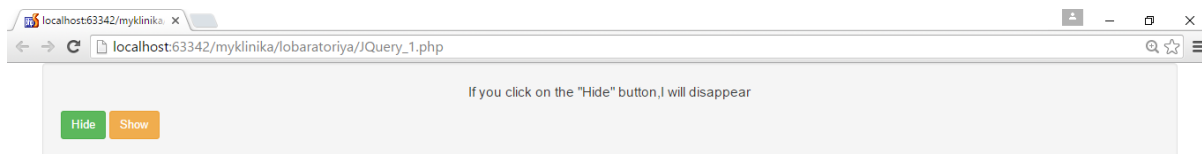
<button id="hide" class="btn btn-success btn-sm">Hide</button>

<button id="show" class="btn btn-warning btn-sm">Show</button>

</div>

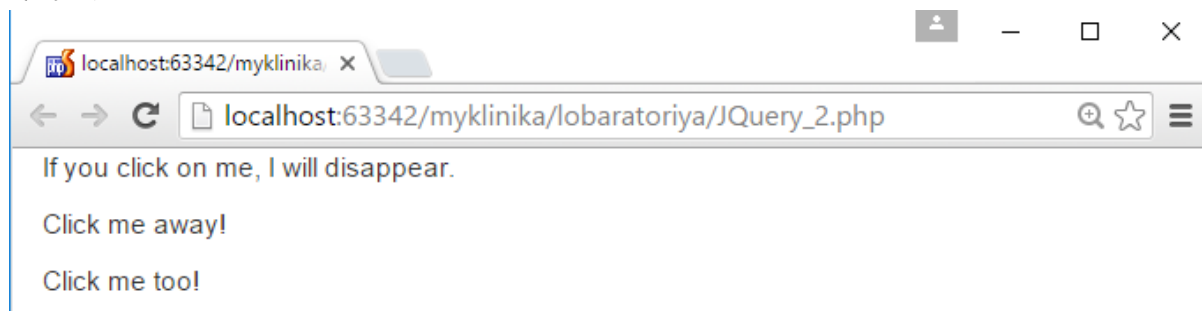
</body>

</html>
```



## 2. Matnni yashirish.

```
<html>
<head>
  <link rel="stylesheet" href="bootstrap.min.css"/>
  <script src="jquery.min.js"></script>
  <script>
    $(document).ready(function(){
      $("p").click(function(){
        $(this).hide();
      })
    })
  </script>
</head>
<body>
<div class="container">
  <p>If you click on me, I will disappear.</p>
  <p>Click me away!</p>
  <p>Click me too!</p></div>
</body>
</html>
```



## 3. Focus va Blur

```
<html>
<head>
  <link rel="stylesheet" href="bootstrap.min.css"/>
  <script src="jquery.min.js"></script>
  <script>
    $(document).ready(function(){
      $("input").focus(function(){
        $(this).css("background", "#cccccc");
      })
    })
  </script>
</head>
<body>
  <input type="text" value="Focus and Blur" />
</body>
</html>
```



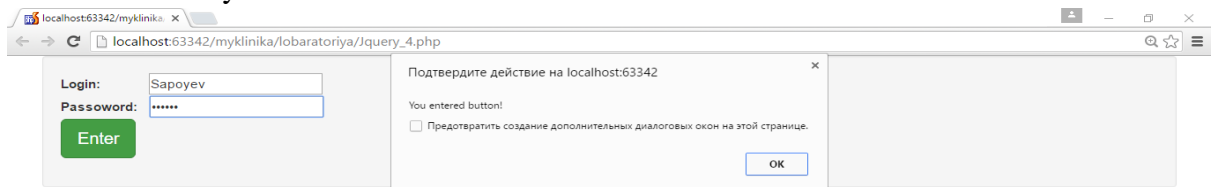
#### 4. Hover hodisasi.

```

<html>
<head>
    <link rel=«stylesheet" href="bootstrap.min.css"/>
    <script src=«jquery.min.js"></script>
    <script>
        $(document).ready(
            function(){
                $("#btn").hover(function(){
                    alert("You entered button!");
                },
                function(){
                    alert("By! You now leave button!");
                })
            })
    </script>
</head>
<body>
<div class=«container well">
    <form action="JQuery_1.php">
        <label for=«ism">Login: </label><input id=«ism" style="margin-left: 50px"
type="text"/><br/>
        <label for=«fam">Passoword: </label><input id=«fam" style="margin-left: 12px"
type="password"/> <br/>

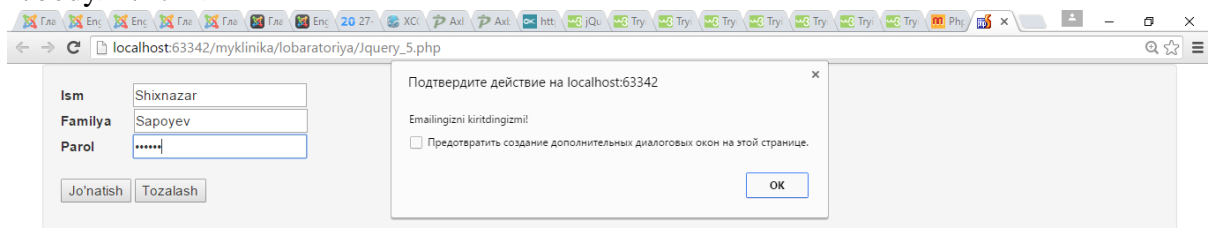
```

```
<button class=«btn btn-success btn-lg" id=«btn">Enter</button>
</form></body></html>
```



## 5. Mouseleave hodisasi.

```
<html>
<head>
  <link rel=«stylesheet" href="bootstrap.min.css"/>
  <script src=«jquery.min.js"></script>
  <script>
    $(document).ready(function(){
      $("#in").mouseleave(function(){
        alert("Emailingizni kiritdingizmi!")
      })
    })
  </script>
</head>
<body>
  <div class=«container well">
    <form action="JQuery_1.php">
      <label for=«ism">Ism </label><input id=«ism" style="margin-left: 50px"
type="text"/><br/>
      <label for=«fam">Familiya </label><input id=«fam" style="margin-left: 22px"
type="text"/><br/>
      <label>Parol </label><input id=«in" style="margin-left: 40px" type="password"/><br/>
      <input type="submit" value="Jo'natish"/>
      <input type="reset" value="Tozalash"/>
    </form></div>
  </body></html>
```



## **Tajriba ishi №9**

### **Mavzu: Server tushunchasi. XAMPP HTTP-serverini sozlash. HTTP protokoli.**

**Ishdan maqsad:** Server tushunchasini o'rganish. XAMPP paketini o'rnatish.

#### **Nazariy qism**

Hozirgi kunda internet, juda katta tezlikda rivojlanib ketmoqda. Shu bilan birga internetga oid xar hil terminlar ko'paymoqda. Misol uchun, sayt, portal, veb sahifa, veb server,...Bu terminlarni o'rganish, hozirgi zamon uchun muhim hisoblanmoqda. Biror inson bilan internet haqida gaplashsangiz, bu terminlarni bilmasangiz, boshqa tildagi inson bilan gaplashganday bo'lasiz va uni tushunmaysiz. Bu maqolada shunday terminlardan biri – veb server haqida to'xtalib o'taman.

Biror bir internet saytga kirish uchun, brouzerga adres yozsangiz, brouzer shu sayt joylashgan kompyuterga, saytdagi ma'lumotlarni ko'rsatish haqida so'rov jo'natadi, kompyuter oddiy temir bo'lgani uchun bu so'rovni tushunmaydi. Shuning uchun, ya'ni bu so'rovni tushinishi uchun, kompyuterga dastur o'rnatiladi. Mana shu dastur veb dastur deyiladi. Bu kompyuter esa, veb serverga aylanadi. Hozirda veb dastur deyilmasdan, umumiy holda veb server deyilmoqda.

Demak, veb server — kliyentdan kelayotgan so'rovlarni qabul qilib, tegishli javobni jo'natadigan dasturdir. Kliyent sifatida veb brouzer(http, https protokoli orqali), mobil telefon(wap protokoli orqali), ba'zi dasturlar(antiviruslar) bo'lishi mumkin. Veb server orqali butun sayt boshqariladi, ya'ni sayt kataloglariga ruxsatlarni beradi, xavfsizlik jihatidan IP adreslarni blokka oladi, foydalanuvchilarni ro'yxatdan o'tkazish va boshqa ko'plab ishlarni amalga oshiradi.

Veb serverni siz o'z uyingizda yaratishingiz mumkin. Uning uchun bitta kompyuter ajratib, unga biror veb dastur o'rnatasiz va bu dasturni yaxshilab sozlashingiz kerak bo'ladi. So'ng o'z saytingizni shu serverga joylashtirasiz va biror domen sotib olasiz. Bundan kelib chiqadiki, saytingiz ishlashi uchun siz ajratgan kompyuter doim yoqilgan holatda bo'lishi kerak.

Veb serverni tushuntirish uchun bitta lirik chekinish, o'xshatish qilaman(qo'polroq bo'lishi mumkin. Tushuntirish uchun nimalar qilishmidi-a). Biror saytni bitta somsa deb tasavvur qilamiz. Somsa ichidagi masalig'lar, saytning ichidagi ma'lumotlar(matn, rasm, video,...). Bu somsani biror bir inson pishirgan, demak pishirgan odam, veb dasturchidir(kimlardir somsa pishishiga yordam bergan bo'lishi mumkin, bular veb dizaynerlar, verstkachilar). Bu somsani sotish kerak, uning uchun joy topish kerak va ruhsat olish kerak. Bu o'xshatish domen va xostingga mos tushadi. Sotuvchi bo'lsa, veb serverdir, ya'ni sotishni boshqaradi(puli borlarga sotadi, hohlasa sotmasligi ham mumkin), hullas sotishni boshqaradi. :)

Veb serverga o'rnatiladigan veb dastur turlari ko'pdir. Eng zamonaviy va eng ommabop dastur bu – albatta "Apache"dir. Apache Software Foundation korxonasi tomonidan chiqarilgan. Quyida ba'zi bir veb serverlarni nomini keltiramiz.

Apache – Apache Software Foundation.

IIS – Microsoft mahsuloti hisoblanadi. Server operatsion tizimlariga o'rnatilgan(Windows Server 2003, Windows Server 2008,...).

Nginx – ochiq veb server. Katta hajmdagi saytlar uchun.

Lighttpd – ochiq veb server.

Google Web Server – Google tomonidan ishlab chiqilgan veb server.

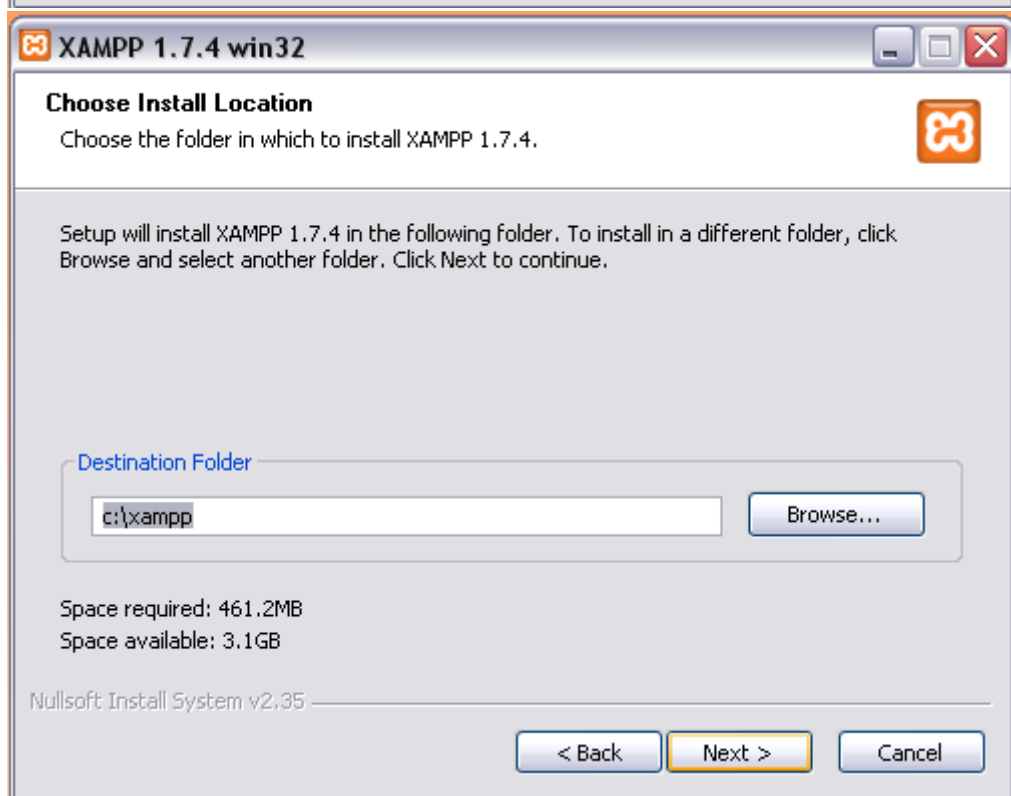
Resin

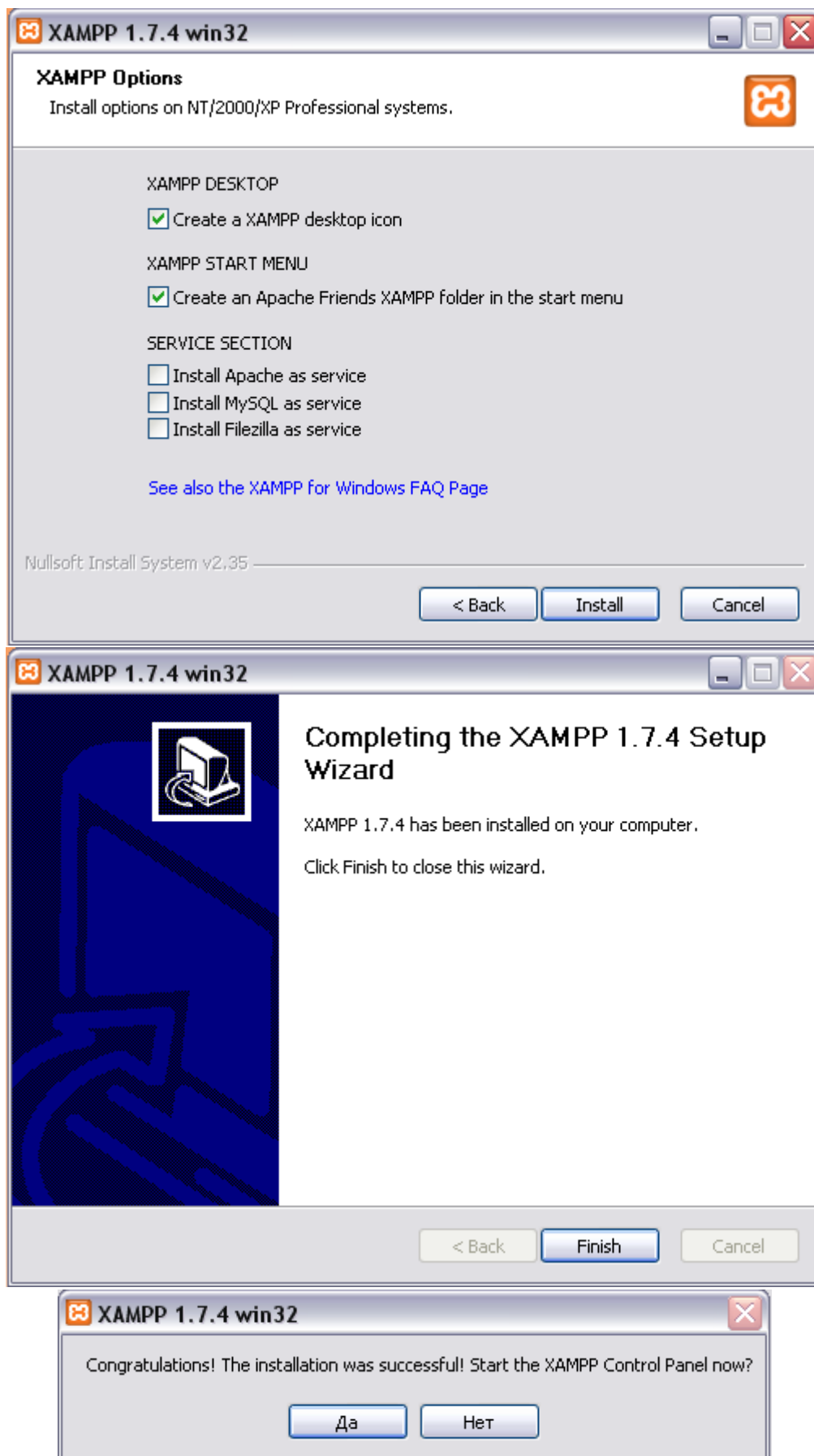
Cherokee – web interfeys orqali boshqariladi.

Rootage – Java tilida yozilgan veb server.

THHTTPD – oddiy, kichik hajmdagi, tez va xavfsiz veb server.

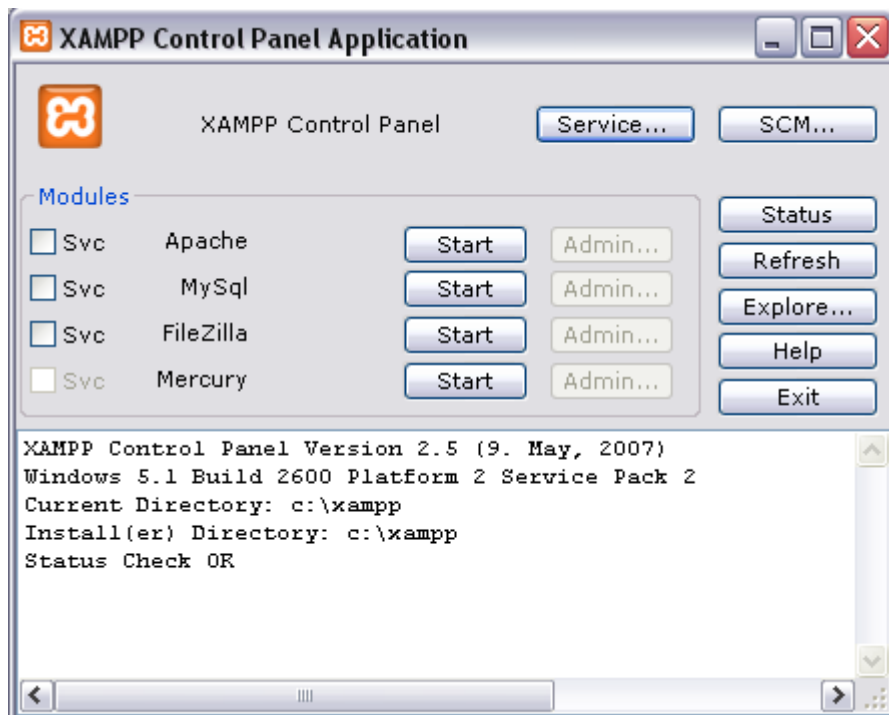
XAMPP paketini o'rnatish ketma-ketligi:





O'rnatish tugagandan so'ng mazkur ko'rinishdagi dasturni ko'rishimiz mumkin bo'ladi.





## **Tajriba ishi №10**

### **Mavzu: PHP dasturlash tili. O'zgaruvchilar, operatorlar va sikllar.**

**Ishdan maqsad:** PHP dasturlash tili umumiy tushunchalarini o'rganish.

#### **Nazariy qism**

PHP 1995 yili, Rasmus Lerdorf Perl/CGI da internet site'ga kirganlarning qanchaligini hisoblovchi kichik bir programmacha yozdi. O'sha vaqtda bundan ishlarni bajaruvchi hech qanday programma bo'lmagan, shuning uchun unga judayam ko'p hatlar kelib, uni qanday qilganligini so'rab surishtirishni boshlashgan.

Shunda Lerdorf o'zining bu kichik programmasini yani PHP(Personal Home Page yoki Hypertext Processor) ni tekinga tarqatishni boshlagan. Bundan tez tarqalib, juda ko'p qulanishni boshlanganini ko'rgan Lerdorf PHP ni ishini yani hususiyatlarini, qo'shimcha komandalarini kengaytirishni boshlaydi. Vaqti kelib Perl uning uchun kamliq qilib u bu programmani C programmalash tilida yozishni boshladi, va PHP'ni ishlab chiqarishda butun dunyo programistlari qatnasha boshlashdi.

1997 yili PHP 3.0 chiqdi, va butun dunyo bo'yicha 50 000 ortiq insonlar shu programmani ishlatishni boshlagan edi.

1999 yilga kelib esa PHP ishlatuvchilar soni 1 000 000 dan ortib ketdi. Bu PHP tarhini kichik bir qismi edi.

PHP tilida dasturlash asoslari

PHP dasturlash tili tilidatuzilgan har qanday dastur .php kengaytmali fayllarda saqlanadi. Misol: uchun functions.php, index.php, admin.php kabi. Birinchi misolni ko'ramaiz. test.php nomli fayl tashkil etamiz va unga quyidagi kodlarni yozamiz:

```
<html>
<head>
<title>HTML hujjat</title>
</head>
<body>
Hujjat tanasi
</body>
</html>
```

Bumatin HTML asosida yozilgan. 1-misoldan ko'rinib turibdiki php hujjat HTML hujjat kabi shakllantirilmoqda. Lekin php hujjatda biz qo'shimcha imkoniyatlarga ega bo'lamiz. test.php faylga quyidagi o'zgarishlarni kiritamaiz:

```
<?php
Echo "<html>";
Echo "<head>";
Echo "<title>HTML hujjat</title>";
Echo "<head>";
Echo "<body>";
Echo "Bu matn HTML asosida yozilgan";
Echo "</body>";
Echo "</html>";
?>
```

2-misol ham 1-si kabi hujjat hosil qiladi. Farqi shundaki ikkinchi holda HTML teglarni PHP dasturi yordamida hosil qilinmoqda. Lekin ko'p hollarda HTML teglari butunicha PHP yordamida hosil qilinmay ora-oralarda PHP kodlari yoziladi:

```
<?php
$title = "HTML hujjat";
$text = "Bu matn PHP asosida yaratilgan";
?>
```

```
<html>
<head>
<title><?php echo $title; ?></title>
</head>
<body>
<?php echo $text;?>
</body>
</html>
```

Yuqoridagi misolni bajarib ko'ring va internet sharhlovchidan HTML kodni ko'rish opsiyasini tanlang. Va siz yana yuqoridagidek HTML teglarini ko'rasiz. Bu yerda PHP kod HTML teglarni generatsiya qilyapti. Demak PHP dasturining vazifalaridan biri HTML teglarini generatsiya qilish ekanini bilib oldik. PHP ning bundan tashqari yana ko'plab imkoniyatlari mavjud bo'lib unga ma'lumotlarbazasiga murojaat qilish, turli hisob-kitoblarni amalga oshirish kabilar ham kiradi. Lekin PHPdan ko'pchilik aynan klent kompyuterida emas, balki serverda bajarilishi uchun foydalanadi. PHP dasuri tuzish haqida shu vaqtgacha bilib olgnlarimizni umumlashtiramiz:

- 1)\*.php kengaytmali fayl yaratiladi;
- 2) faylda istalgancha HTML teglarini ishlatish mumkin;
- 3) PHP kodlarini ochish va yopish belgilari orasida yoziladi:
  - a)<?php –ochish belgisi va ?> -yopish belgisi;
  - b)<? –ochish belgisi va?> - yopish belgisi;
  - c)<script language = "php"> -ochish belgisi va </script>- yopish belgisi;
  - d)<?= -ochish belgisi ba ?> yopish belgisi.

PHP kodlarini yozish bo'yicha yana bitta misol ko'ramiz:

```
<?php // birinchi usuldan foydalanilmoqda

$title = "HTML hujjat";
$text = "Bu matn PHP asosida yaratilgan";
?>

<html>
```

```
<head>
<title>
<? // ikkinchi usuldan foydalanilmoqda
echo $title;
?>
</title>
</head>
<body>

<script language =“php”> // uchunchi usuldan foydalanilmoqda
echo $text;
</script>
</body>
</html>
```

## Tajriba ishi №11

### Mavzu: PHPda funksiyalar yaratish. Satrlar bilan ishlovchi funksiyalar.

**Ishdan maqsad:** PHP da funksiyalar yaratishni va satrli funksiyalar bilan ishlashni o'rganish.

#### Nazariy qism

Funksiyalar nima uchun kerak? Bu savolga javob berish uchun, funksiya o'zi nima ekanligini tushunib olish lozim bo'ladi. Dasturlashda, xuddi matematikadagi kabi, unga bog'liq ko'pgina argumentlarning uning ko'pgina ma'nolarida aks etishidir. Demak, funksiya argumentning har bir ma'nolari jamlanmasi uchun uning bajargan ishi natijasi sifatida qandaydir ma'no qaytaradi. Funksiyalar nima uchun kerak, buni misollar bilan oydinlashtirishga harakat qilamiz. Dasturlashdagi funktsiyaga klassik misol – bu sonning faktorial ahamiyatini hisoblab beruvchi funksiya. Demak, biz unga son beramiz, u esa bizga uning faktorialini qaytaradi. Bunda biz faktorialini olishni xohlagan har bir son uchun aynan bir xil kodni qaytaravermaymiz – bu songa teng bo'lgan argumentli funktsiyani chaqirishning o'zi kifoya qiladi.

Natural son faktorialini hisoblash funktsiyasi

```
<?php
function fact($n){
    if ($n==0) return 1;
    else return $fact = $n * fact($n-1);
}
echo fact(3);
    // echo (3*2) deb yozish mumkin edi;
    // lekin son katta bo'lsa,
echo fact(50);
    // funktsiyadan foydalanish qulayroq,
    // echo (50*49*48*...*3*2) deb yozgandan;
?>
```

Shu yo'l bilan biz biron-bir ma'lumotga bog'liqlik zaruriyati tug'ilgan amalni bajarganimizda, bu holda ham biz aynan shunday amallarni bajarishimiz oshirishimiz lozim bo'ladi, faqat boshqa boshlang'ich ma'lumotlardan foydalanamiz, funktsiyalar mexanizmidan foydalanish– funksiya tanasi ko'rinishidagi amallar blokini taxt qilish, o'zgaruvchan ma'lumotlarni esa – uning parametrlari sifatida foydalanish qulayroq bo'ladi.

Funktsiya topshirig'i (e'loni) umumiy tarzda qanday bo'lishini ko'ramiz. Funktsiya quyidagi sintaksis yordamida aniqlanadi:

```
function Funktsiya_nomi (1-parametr, 2-parametr,
... N-parametr){
    Amallar bloki
    return "funktsiyaga aylanuvchi ma'no";
}
```

Agar php-dasturda to'g'ridan-to'g'ri yozilsa, hech narsani ishlab bo'lmaydi. Birinchidan, funksiya nomi funksiya parametrlari nomlari (1-parametr, 2-parametr va b.) PHP da nomlanish qoidalariga muvofiq kelishi kerak (unda yaxshisi kirillcha simvollarni ham ishlatmagan ma'qul). Funktsiya nomlari registrga nisbatan sezuvchan bo'ladi. Ikkinchidan, funksiya parametrlari – tilning o'zgaruvchan qismlari, shuning uchun ularning har birining nomlari oldidan \$ belgisi turishi lozim bo'ladi. Parametrlar ro'yxatida hech qanday ko'p nuqtalarni qo'yish mumkin emas.

Uchinchidan, amallar bloki so'zi bilan birga funktsiya tanasida istalgan to'g'ri PHP-kod mavjud bo'lishi kerak (parametrlarga muvofiq bo'lishi majburiy emas). Va nihoyat, return kalit so'zidan so'ng tartibli php-ifoda kelishi lozim (ma'noga ega bo'lgan qandaydir simvollar). Bundan tashqari, funktsiyada qaytariluvchi ma'no kabi parametrlar bo'lmasligi ham mumkin. Funktsiyani to'g'ri e'lon qilishga misol – yuqorida keltirilgan faktorialni hisoblash funktsiyasi.

Funktsiya chaqirish qanday amalga oshadi? Funktsiya nomi va yumaloq qavslar ichida uning parametrlari ma'nolari ro'yxati ko'rsatiladi, agar shundaylari mavjud bo'lsa:

```
<?php
Funktsiya_nomi
("1-parametr_uchun_ma'no",
 "2-parametr_uchun_ma'no ",...);
// Funktsiyani chaqirishga misol – funktsiyani chaqirish
// faktorialni hisoblash yuqorida bor,
// 3 soni faktorialini hisoblash uchun u erda biz
// fact(3) deb yozganmiz;
// u erda fact – chaqiriluvchi funktsiya nomi,
// a 3 –$n nomli uning parametri ma'nosi
?>
```

Funktsiyani qachon chaqirish mumkin? Bu g'alati savol bo'lib tuyulishi mumkin. Funktsiyani uni aniqlangandan keyin chaqirish mumkin, ya'ni function f\_name(){...} blokidan pastda istalgan dastur qatorida. PHP3 da bu aynan shunday. Lekin PHP4 da bunday talab yo'q. Hamma gap interpretator olingan kodni qanday qayta ishlashida. Birgina istisno shartli ravishda aniqlanadigan funktsiyadan tashkil topadi (shartli operatorlar yoki boshqa funktsiyalar ichida). Funktsiya shu tarzda aniqlangan taqdirda, uni aniqlash uni chaqirishdan oldin bajariladi.

```
<?
$make = true;
/* bu erda Make_event() ni chaqirish mumkin emas;
Chunki u hali mavjud emas, lekin Save_info() ni chaqirish mumkin*/
```

```
Save_info("Sobir", "Sodiqov",
 "Men PHP kursini tanladim");

if ($make){
// Make_event() funktsiyasini aniqlash
function Make_event(){
    echo "<p>Python<br> ni o'rganmoqchiman";
}
}
// endi Make_event() ni chaqirish mumkin
Make_event();
// Save_info funktsiyasini aniqlanadi
function Save_info($first, $last, $message){
    echo "<br>$message<br>";
    echo "Ism: ". $first . " ". $last . "<br>";
}
Save_info("Murod", "Yoqubov",
 "Men Lisp ni tanladim");
// Save_info ni bu erda ham chaqirish mumkin
?>
```

misol. Shartli funktsiya ichida funktsiyani aniqlash

Agar funktsiya dastur ichida aniqlangan bo'lsa, uni keyin qayta aniqlash yoki o'chirib tashlash mumkin emas. Funktsiya nomlariga registr ta'sir qilmasligiga qaramasdan, yaxshisi funktsiyani aniqlash paytida berilgan nom bilan chaqirish mumkin bo'ladi.

```
<?php
```

```
/* ma'lumotlarni saqlash, ya'ni DataSave() funktsiyasini chaqirish mumkin emas. Uning to'g'riligi tekshirilmasdan oldin, ya'ni DataCheck() funktsiyasi chaqirilmasdan oldin bu mumkin emas.*/
```

```
DataCheck();
```

```
DataSave();
```

```
function DataCheck(){
```

```
// ma'lumotlar to'g'riligini tekshirish
```

```
function DataSave(){
```

```
// ma'lumotlarni saqlaymiz
```

```
}
```

```
}
```

```
?>
```

misol. Funktsiya ichidagi funktsiyani aniqlash

Funktsiya argumentlari, ularning ma'nolari va ishlatilishini batafsil ko'rib chiqamiz.

Funktsiyalarning argumentlari

Har bir funktsiyada, avval aytganimizday, argumentlar ro'yxati bo'lishi mumkin. Bu argumentlar yordamida funktsiyaga har xil ma'lumotlar beriladi (masalan, faktoriali hisoblanishi kerak bo'lganson ma'nosi). Har bir argument o'zgaruvchi va konstantaga ega bo'ladi.

Argumentlar yordamida ma'lumotlar funktsiyaga uch xil turli usullar bilan o'tkazilishi mumkin. Bu argumentlarni ma'nosiga ko'ra (o'zgarmas holatda foydalaniladi), ilovalarga ko'ra va o'zgarmas holatda argumentlarga ma'no berishga ko'ra o'tkazish.

Bu usullarni atroflicha ko'rib chiqamiz.

Argument funktsiyaga ma'nosiga ko'ra o'tkazilsa, funktsiya ichidagi argument ma'nosining o'zgarishi uning funktsiya tashqarisidagi ma'nosiga ta'sir qilmaydi. Funktsiyaga uning argumentlarini o'zgartirishga yo'l qo'yish uchun ularni havolalarga ko'ra o'tkazish kerak. Buning uchun argument nomi oldidan funktsiyani aniqlashda ampersand "&" belgisini yozish kerak.

```
<?php
```

```
// qo'shimcha qilishi mumkin bo'lgan funktsiyani yozamiz
```

```
// checked so'zi qatoriga
```

```
function add_label(&$data_str){
```

```
$data_str .= "checked";
```

```
}
```

```
$str = "<input type=radio name=article ";
```

```
// bunday qator mavjud bo'lsin
```

```
echo $str."><br>";
```

```
// forma elementini keltiradi –
```

```
// belgilanmagan radio knopkasini
```

```
add_label($str);
```

```
// funktsiyani chaqiramiz
```

```
echo $str."><br>";
```

```
// bu endi belgilangan
```

```
// radio knopkani keltiradi
```

```
?>
```

misol. Argumentlarni havolasiga ko'ra o'tkazish

Funktsiyada tinch holatda foydalanilayotgan argumentlar ma'nosini aniqlash mumkin. Ayni paytdagi ma'noning o'zi konstant ifoda bo'lishi, o'zgartirish va sinf vakili yoki boshqa funktsiya chaqiruvi bo'lmasligi lozim.

Bizda informatsion xabar tuzuvchi funktsiya, unga berilgan parametr ma'nosiga muvofiq tarzda o'zgaruvchi imzo bor. Agar parametr ma'nosi berilmagan bo'lsa, "Tashkiliy qo'mita" imzosidan foydalaniladi.

```
<?php
```

```
function Message($sign=«Tash.qo'mita.»){
```

```
// bu erda parametr sign ayni paytda "Tash.qo'mita" ma'nosiga ega
```

```
echo "Keyingi yig'ilish ertaga bo'lib o'tadi.<br>";
```

```
echo "$sign<br>";
```

```
}
```

```
Message();
```

```
// Parametrsiz funktsiyani chaqiramiz.
```

```
// Bu holda imzo – Bu Tashkiliy qo'mita
```

```
Message("Hurmat bilan Kamoliddin");
```

```
// Bu holda imzo
```

```
// "Hurmat bilan Kamoliddin." bo'ladi
```

```
?>
```

misol. Tinch holatdagi argument ma'nosi

Bu skript ishining natijasi quyidagicha:

Keyingi yig'ilish ertaga bo'lib o'tadi.

Tashkiliy qo'mita.

Keyingi yig'ilish ertaga bo'lib o'tadi.

Hurmat bilan Kamoliddin.

Agar funktsiyaning bir necha parametrlari bo'lsa, tinch holatda ma'no beriluvchi bu argumentlar funktsiya aniqlanishida boshqa barcha argumentlardan keyin yozilishi kerak. Aks holda, agar bu argumentlar funktsiyani chiqarish paytida ko'zdan qochirilsa xato yuzaga kelishi ehtimoli bor.

Masalan, biz katalogga maqola tavsifini kiritmoqchimiz. Foydalanuvchi maqolaga uning nomlanishi, muallifi va qisqa tavsif kabi xarakteristikalarini keltirishi lozim bo'ladi. Agar Foydalanuvchi maqola muallifi ismini kiritmadi, u Murod Yoqubov deb olaylik.

```
<?php
```

```
function Add_article($title, $description,
```

```
$author=«Murod Yoqubov»){
```

```
echo "Maqolani katalogga kiritamiz: $title,";
```

```
echo "muallif $author";
```

```
echo "<br>Qisqa tavsif: ";
```

```
echo "$description <hr>";
```

```
}
```

```
Add_article("Informatika va biz",
```

```
"Bu maqola informatikaga oid ...",
```

```
"Zayniddin Saidov");
```

```
Add_article("Xarakterlar kim",
```

```
"Bu maqola xarakterlar haqida ...");
```



?>

Skript ishi natijasi sifatida quyidagilarni olamiz  
Katalogga maqola kiritamiz: Informatika va biz,  
Muallif Murod Yoqubov.  
Qisqa tavsif:  
Bu maqola informatikaga oid...

Katalogga maqola kiritamiz: Xarakterlar kim,  
Muallif Odil Ziyaev.  
Qisqa tavsif:

Bu maqola xarakterlar haqida...  
Agar biz quyidagicha yozsak:  
<?php  
function Add\_article(\$author=«Odil Ziyaev»,  
    \$title, \$description){  
// ... avvalgi misoldagi kabi amal  
}  
Add\_article("Xarakterlar kim",  
    "Bu maqola xarakterlar haqida...");

?>

Natija quyidagicha bo'ladi:  
Warning: Missing argument 3 for  
    add\_article() in  
c:/users/nina/tasks/func/def\_bad.php  
    on line 2

O'zgaruvchan uzunlik argumentlari ro'yxatlari

PHP4 da argumentlarning o'zgaruvchan soni bilan funktsiya tuzish mumkin. Ya'ni biz uni necha argumentlar bilan chaqirilishini bilmasdan, funktsiya tuzamiz. Bu kabi funktsiya yozish uchun hech qanday maxsus sintaksis kerak bo'lmaydi. Hammasi uning ichiga o'rnatilgan funktsiyalar func\_num\_args(), func\_get\_arg(), func\_get\_args() yordami bilan qilinadi.

func\_num\_args() funktsiyasi argumentlar sonini qaytaradi. Bu funktsiya faqat foydalanuvchi funktsiyasini aniqlash mobaynida foydalanishi mumkin. Agar u funktsiyadan tashqarida paydo bo'lsa, interpretator ogohlantirish beradi.

<?php  
function DataCheck(){  
    \$n = func\_num\_args();  
    echo "Funktsiya argumentlari soni \$n";  
}  
DataCheck();  
    // qatorni keltiradi  
    // "0 funktsiya argumentlari soni"  
DataCheck(1,2,3);  
    // qatorni keltiradi  
    // "3-funktsiya argumentlari soni"  
?>

misol. func\_num\_args() funktsiyasidan foydalanish

func\_get\_arg funktsiyasi (argument\_raqami to'raligicha) argumentni o'zgarishlar ro'yxatidan argumentlar funktsiyasiga qaytaradi, uning tartib raqami func\_get\_arg parametri

bilan beriladi. Funktsiya argumentlari noldan boshlab hisoblanadi. func\_num\_args() kabi bu funktsiya faqat biron-bir funktsiyani aniqlashda foydalaniladi.

Argument raqami funktsiyaga o'zgargan argumentlar sonidan ortib ketishi mumkin emas. Aks holda ogohlantirish umumlashtiriladi va func\_num\_args() funktsiyasiga False qiymat qaytadi.

Ma'lumotlarni tekshirish uchun funktsiyaga uning argumentlarini tuzamiz. Agar funktsiyaning birinchi argumenti – butun son, ikkinchisi – qator bo'lsa, tekshirish muvaffaqiyatli o'tdi, deb hisoblaymiz.

<?

```
function DataCheck(){
    $check =true;
    $n = func_num_args();
    // funktsiyaga o'zgargan
    // argumentlar sonini
    /* tekshiramiz, birinchi o'zgargan argument butun sonmi-yo'qmi */
    if ($n>q1) if (!is_int(func_get_arg(0)))
        $check = false;
    /* tekshiramiz, ikkinchi o'zgargan argument qatormi-yo'qmi */
    if ($n>q2)
        if (!is_string(func_get_arg(1)))
            $check = false;
    return $check;
}
if (DataCheck(123,"text"))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
if (DataCheck(324))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
?>
```

misol. Ma'lumotlar tipini, uning argumentlarini tekshirish

Ishning natijasi quyidagicha bo'ladi.

Ma'lumotlar shartlarni qondirmaydi. Tekshiruv to'g'ri o'tdi.

func\_get\_args() funtsiyasi argumentlar ro'yxatidan tashkil topgan massiv qaytaradi. Massivning har bir elementi argumentga, funktsiya o'zgarishiga to'g'ri keladi. Agar funktsiya foydalanuvchi funktsiyasi aniqligidan tashqarida foydalanilsa ogohlantirish umumlashtiriladi.

Avvalgi misolni ko'chiramiz, bu funktsiyadan foydalanamiz. Funktsiyani harakatlantiruvchi juft argument butun son ekanligini tekshiramiz:

<?

```
function DataCheck(){
    $check =true;
    $n = func_num_args();
    // funktsiyaga o'zgargan
    // argumentlar soni

    $args = func_get_args();
    // funktsiya argumentlari massivi
    for ($i=0;$i<$n;$i++){
```

```

    $v = $args[$i];
    if ($i % 2 == 0){
        if (!is_int($v)) $check = false;
        // tekshiramiz,
        // juft argument butunmi-yo'qmi
    }
}
return $check;
}
if (DataCheck(array("text", 324)))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi <br>";
?>

```

Ayon bo'ldiki, func\_num\_args(), func\_get\_arg() va func\_get\_args() funktsiya kombinatsiyasi funktsiyalar o'zgaruvchan argumentlar ro'yxatiga ega bo'la olishi uchun foydalaniladi. Bu funktsiyalar faqat PHP4 ga kiritilgan. PHP3 da shunday natijaga erishish uchun, argument sifatida massiv funktsiyasidan foydalanish mumkin bo'ladi. Masalan, har bir toq funktsiyalar parametri butun sonligini tekshiruvchi skriptni quyidagicha yozish mumkin:

```

<?
function DataCheck($params){
    $check =true;
    $n = count($params);
    // funktsiyaga o'zgargan
    // argumentlar soni

    for ($i=0;$i<$n;$i++){
        $v = $params[$i];
        if ($i % 2 !== 0){
            // tekshiramiz, toq argument
            // butunmi-yo'qmi
            if (!is_int($v)) $check = false;
        }
    }
    return $check;
}
if (DataCheck("text", 324))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi<br>";
?>

```

Funktsiyalar ichida o'zgaruvchilardan foydalanish  
Global o'zgaruvchilar

Funktsiyalar ichida undan tashqarida berilgan o'zgaruvchilardan foydalanish uchun, bu o'zgaruvchilarni global deb e'lon qilish kerak. Buning uchun funktsiya tanasida uning nomlarini global kalit so'zidan keyin keltirish lozim bo'ladi:

```

global $var1, $var2;
<?
$aq1;
function Test_g(){
global $a;

```

```

$a = $a*2;
echo ' $a=',$a funktsiya ishida natija;
}
echo 'funktsiyadan tashqarida $a=',$a,' ';
Test_g();
echo "<br>";
echo funktsiyadan tashqarida $a=',$a,' ';
Test_g();
?>

```

misol. Global o'zgaruvchilar

Bu skript ishidan quyidagi natijalarni olamiz:

\$aq2 funktsiyadan tashqarida, \$q2 funktsiya ishida natija

\$aq2 funktsiyadan tashqarida, \$q4 funktsiya ishida natija

O'zgaruvchi global deb e'lon qilinganda, aniq global o'zgaruvchi uchun havola tuziladi. Buning uchun bunday yozuv quyidagiga ekvivalent (GLOBALS massivi mavjud ko'rinish sohaslariga muvofiq barcha global o'zgaruvchilarni o'z ichiga oladi):

```
$var1 = & $GLOBALS["var1"];
```

```
$var2 = & $GLOBALS["var2"];
```

Bundan kelib chiqadiki, \$var1 o'zgaruvchini o'chirish \$\_GLOBALS["var1"] global o'zgaruvchiini o'chirib tashlamaydi.

Statistik o'zgaruvchilar

O'zgaruvchilardan faqat funktsiya ichida foydalanish uchun bunda uning ma'nosini saqlagan holda va funktsiyadan chiqqandan so'ng, bu o'zgaruvchilarni statistik deb e'lon qilish kerak. Statistik o'zgaruvchilar faqat funktsiyalar ichida ko'rinadi va dasturni yuklash funktsiya doirasidan tashqariga chiqsa o'z ma'nosini yo'qotmaydi. Bu o'zgaruvchilarni e'lon qilish static kalit so'zi yordamida amalga oshiriladi:

```
static $var1, $var2;
```

Har qanday ma'no statistik o'zgaruvchi sifatida talqin qilinishi mumkin, faqat havola emas.

```
<?
```

```
function Test_s(){
```

```
static $a = 1;
```

```
// ifoda yoki havolani o'zlashtirib bo'lmaydi
```

```
    $a = $a*2;
```

```
    echo $a;
```

```
}
```

```
Test_s(); // 2 chiqadi
```

```
echo $a; // hech narsa chiqmaydi, zero
```

```
    // $a faqat funktsiya ichida
```

```
    // kirish yo'lagi bor
```

```
Test_s(); // $aq2 funktsiya ichida, shuning uchun
```

```
    // funktsiya ishi natijasi
```

```
    // 4 soni bo'ladi
```

```
?>
```

misol. Statistik o'zgaruvchilardan foydalanish

Qaytariluvchan ma'nolar

Yuqorida misol qilib keltirilgan barcha funktsiyalar biror-bir amal bajarishgan. Bunday hollardan tashqari, har qanday funktsiya o'z ishi natijasi sifatida qandaydir qiymat qaytaradi. Bu return tasdig'i yordamida qilinadi. Qaytariluvchan qiymat har qanday turda, shu jumladan, ro'yxat

va ob'ektlar bo'lishi mumkin. Interpretator funktsiya tanasida return komandasiga uchraganda, u darhol uni bajarishni to'xtatadi va funktsiya chaqirilgan qatorga o'tib ketadi.

Masalan, inson yoshini qaytaruvchi funktsiya tuzamiz. Agar inson vafot etmagan bo'lsa, yosh joriy yilga muvofiq hisoblanadi.

```
<?php
/* agar ikkinchi parametr true kabi hisoblab chiqilsa, u vafot etgan sanaday ko'rib chiqiladi, */

function Age($birth, $is_dead){
    if ($is_dead) return $is_dead-$birth;
    else return date("Y")-$birth;
}
echo Age(1971, false); // vo'vedet 33
echo Age(1971, 2001); // vo'vedet 30
?>
```

Bu misolda return funktsiyasidan foydalanmasa ham bo'ladi, shunchaki uni chiqarish funktsiyasini echo ga almashtiriladi. Aksincha, agar biz funktsiya biror-bir qiymat qaytaradigan qilsak (bu misolda inson yoshi), biz dasturda o'zgaruvchini bu funktsiya qiymatini istalgan o'zgaruvchiga o'zlashtirishimiz mumkin.

```
$my_age = Age(1981, 2004);
```

Funktsiya ishi natijasida faqat bitta qiymat qaytarilishi mumkin. Bir necha qiymatni qiymatlar ro'yxati qaytarilgan taqdirda olish mumkin (bir o'lchamli massiv). Biz inson yoshini kunigacha aniqlikda olmoqchimiz, deb hisoblaylik.

```
<?php
function Full_age($b_day, $b_month, $b_year)
{
    $y = date("Y");
    $m = intval(date("m"));
    $d = intval(date("d"));
    $b_month = intval($b_month);
    $b_day = intval($b_day);
    $b_year = intval($b_year);

    $day = ($b_day > $d ? 30 - $b_day + $d : $d - $b_day);
    $tmpMonth = ($b_day > $d ? -1 : 0);
    $month = ($b_month > $m + $tmpMonth
        ? $b_month + $tmpMonth - $m : $mQ$tmpMonth - $b_month);
    $tmpYear = ($b_month > $m + $tmpMonth ? -1 : 0);
    if ($b_year > $y + $tmpYear)
    {
        $year = 0; $month = 0; $day = 0;
    }
    else
    {
        $year = $y + $tmpYear - $b_year;
    }
    return array ($day,$month,$year);
}
```

```
$age = Full_age("29","06","1986");
echo "Siz $age[2] yosh, $age[1] oylar va $age[0] kunlar";
?>
```

Funktsiya bir necha qiymatlarni ularni dasturda qayta ishlash uchun qaytarganda, bir amal bilan ma'noni birdaniga bir necha o'zgaruvchilarni o'zlashtirishga imkon beruvchi list() til konstruktsiyasidan foydalanish qulay bo'ladi. Masalan, yuqoridagi misolda funktsiyani, uning qiymatiga o'zgartirish kiritmay qayta ishlash quyidagicha bo'lishi mumkin:

```
<?
// Full_age() funktsiya kiritish
list($day,$month,$year) = Full_age("07",
    "08","1974");
echo "Sizning yoshingiz $year, $month oy va
    $day kun";
?>
```

list() konstruktsiyasini umuman o'zgaruvchini o'zlashtirish uchun istalgan massiv elementi qiymatidan foydalanish mumkin.

```
<?
$arr = array("first","second");
list($a,$b) = $arr;
    // o'zgaruvchi $a o'zlashtiriladi, birinchi
    // massiv qiymati, $b – ikkinchi
echo $a, " ", $b;
    // «first second» qatori keltiriladi
?>
```

misol. list() dan foydalanish

Havolani qaytarish

Funktsiya o'z ishi natijasida shuningdek havolani biror-bir o'zgaruvchiga qaytarishi mumkin. Bu funktsiyani qanday o'zgaruvchi havolaga o'zlashtirish kerakligini aniqlash uchun foydalaniladi. Funktsiyadan havola olish uchun, e'lon oldidan ampersand (&) belgisini yozish kerak bo'ladi va har safar funktsiya chaqiruvi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo'ladi. Ko'pincha funktsiya havolani biror-bir global o'zgaruvchiga (yoki uning qismini – havolani global massiv elementiga), havolani statistik o'zgaruvchiga (yoki uning qismini) yoki havolani argumentlardan biriga qaytaradi, agar u havola bo'yicha berilgan bo'lsa.

```
<?
$a = 3; $b = 2;
function & ref($par){
    global $a, $b;
    if ($par % 2 qq 0) return $b;
    else return $a;
}
$var =& ref(4);
echo $var, " i ", $b, "<br>";
    // 2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "<br>";
    // 10 va 10 keltiriladi
?>
```

misol. Havolani qaytarish

Havola sintaksisidan foydalanishda bizning misoldagi \$var o'zgaruvchi o'zgaruvchining \$b qiymati \$ref qaytarilgan funktsiyasiga ko'chirilmaydi, bu o'zgaruvchiga havola tuziladi. Demak, endi \$var va \$b teng kuchli o'zgaruvchilar va ular bir paytda o'zgartiriladi.

Funktsiyaning o'zgaruvchilari

PHP funktsiyalar o'zgaruvchilariga ko'maklashadi. Bu degani, agar o'zgaruvchi nomi oddiy qavslar bilan tugasa, PHP shu kabi nomli funktsiyani qidiradi va uni bajarishga harakat qiladi.

<?

/\* Ikkita oddiy funktsiya tuzamiz:

Add\_sign – qatorga imzo qo'shadi va

Show\_text –matn qatorini chiqarib beradi\*/

```
function Add_sign($string,
    $sign=«Hurmat bilan, Murod"){
    echo $string ." ".$sign;
}
function Show_text(){
    echo "Xabarni pochtadan jo'natish<br>";
}
$func = "Show_text";
// ma'noga ega o'zgaruvchi tuzamiz,
// u funktsiya nomiga teng Show_text
```

## Tajriba ishi №12

### Mavzu: PHPda MySQL ma'lumotlar ombori bilan ishlash. MySQL asoslari. SELECT, INSERT, UPDATE va DELETE operatorlari.

**Ishdan maqsad:** PHP da MySQL ma'lumotlar ombori va SELECT, INSERT, UPDATE va DELETE operatorlari bilan ishlashni o'rganish.

#### Nazariy qism

Ushbu bo'lim PHP va MySQL MBBT (CYБД) o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funktsiyalari va javoblarni (`mysql_connect`, `mysql_query`, `mysql_result`, `mysql_num_rows`, `mysql_close`) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'lumotlar bazasining ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributiviga MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan funktsiyalarni oluvchi kengaytma kiradi. Bu bo'limda MySQL bilan ishlash uchun ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish topshirig'ini echish uchun kerak bo'ladigan asosiy funktsiyalar bilan tanishamiz. Savol tug'iladi: bunday interfeyslarni tuzish nega kerak? SQL so'rovlar tili bilan notanish odamlar ma'lumotni ma'lumotlar bazasiga kiritish va uning tarkibini ko'rib turish imkoniyati bo'lishi uchun shunday qilinadi. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi, bizning skript esa qolgan barcha amallarni bajaradi. Jadval tarkibini ko'rib turish uchun havola ustiga bir marta bosish va kerakli sahifaga kirish kifoya.

Ko'rinib turishi uchun bu interfeyslarni virtual muzey eksponatlari haqidagi ma'lumotlar joylanadigan Artifacts jadvallari uchun tuzamiz. Avvalgi bo'limda bu kolleksiya strukturaning hamda uning shaxs (Persons) va tasvirlar (Images) tavsiflari kolleksiyalari bilan aloqasini kiritgan edik. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:

- nom (title);
- muallif (author);
- ifoda (description);
- o'rindosh nom (alternative);
- tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni `VARCHAR(255)`), tasvirlash – matnli maydon (`TEXT` turiga mansub) hisoblanadi, “muallif” va “tasvir” maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

#### Ma'lumotni qo'shish uchun interfeys tuzish

Demak, bizda ma'lumotlar bazasida biron-bir jadval bor. Bu jadvalga ma'lumotni qo'shishga mo'ljallangan interfeys tuzish uchun uning strukturasini (ya'ni uning maydonlari jamlanmasini) *html-formada* tasvirlash kerak bo'ladi.

Bu topshiriqni quyidagi tarkibiy topshiriqlarga bo'lib chiqamiz:

- *MB bilan ulanishni o'rnatish;*
- *MB ishini tanlash;*
- *Jadval maydonlari ro'yxatini olish;*
- *html-formada maydonlarni tasvirlash.*

*Bundan keyin formaga kiritilgan ma'lumotlarni ma'lumotlar bazasiga kiritish kerak bo'ladi.*

*Bu topshiriqlarning barchasini tartib bilan ko'rib chiqamiz.*



## **Aloqa o'rnatish**

Demak, birinchi galdagi vazifa – bu ma'lumotlar bazasi bilan aloqa o'rnatish. *mysql\_connect* funksiyasidan foydalanamiz. *mysql\_connect* sintaksisi *mysql\_connect* resursi ([server qatori[, username qatori[, password qatori[, mantiqiy new\_link[, butun client\_flags]]]])

Bu funktsiya MySQL serveri bilan aloqa o'rnatadi va bu aloqaga ko'rsatkich qaytaradi yoki muvaffaqiyatsiz chiqqanda FALSE ko'rsatadi. Etishmayotgan parametrlar uchun quyidagi ma'nolar muvaqqat qo'yiladi:

server = 'localhost:3306'

username = server jarayoni egasidan foydalanuvchi ismi

password = bo'sh parol

Agar funktsiya aynan bir xil parametrlar bilan ikki marta chaqiriladigan bo'lsa, yangi ulanish o'rnatilmaydi, aksincha havola eski aloqaga qaytariladi. Bundan qochish uchun esa, har qanday holatda yangi bir aloqa ochishga majbur qiluvchi *new\_link* parametridan foydalaniladi.

*client\_flags* parametri – bu quyidagi konstantalar kombinatsiyasi:

*MYSQL\_CLIENT\_COMPRESS* (siqish protokolidan foydalaniladi),

*MYSQL\_CLIENT\_IGNORE\_SPACE*

(*funktsiya nomidan so'ng probel qo'yishga ruxsat beradi*),

*MYSQL\_CLIENT\_INTERACTIVE* (*interactive\_timeout* bir soniya kutish - *wait\_timeout* – ulanish tugatilishigacha). *new\_link* parametri PHP 4.2.0 da mavjud, *client\_flags* esa – PHP 4.3.0 da.

Server bilan ulanish, agar u bungacha *mysql\_close()* yordamida yopilgan bo'lsa, skriptni amalga oshirish tugallanishida berkiladi.

Shunday qilib, "123" parolli nina foydalanuvchisi uchun lokal serverda ma'lumotlar bazasi bilan ulanishni o'rnatamiz.

<?

```
$conn = mysql_connect(
    "localhost", "nina", "123")
```

```
or die("Ulanishni amalga oshirib bo'lmaydi: ". mysql_error());
```

```
echo "Ulanish amalga oshdi";
```

```
mysql_close($conn);
```

```
?>
```

### ***mysql\_connect* amali**

shell>mysql -u nina -p123 buyrug'i bilan teng kuchli.

## **Ma'lumotlar bazalarini tanlash**

*Ulanishni amalga oshirgandan* so'ng birga ishlashimiz kerak bo'lgan ma'lumotlar bazasini tanlashimiz kerak bo'ladi. Bizning ma'lumotlarimiz book ma'lumotlar bazasida saqlanadi.

MySQL da ma'lumotlar bazasini tanlash use buyrug'i yordamida amalga oshiriladi:

```
mysql>use book;
```

PHP da buning uchun *mysql\_select\_db* funktsiyasi mavjud

*mysql\_select\_db*: sintaksisi

mantiqiy *mysql\_select\_db* (

database\_name qatori

[,link\_identifier resursi])

Bu funktsiya TRUE qiymatni ma'lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo'lganda.

Book ma'lumotlar bazasini ishga tushiramiz:

<?

```
$conn = mysql_connect(
    "localhost", "nina", "123")
```

```

or die("Ulanishni amalga oshirib bo'lmaydi: ". mysql_error());
echo "Ulanish amalga oshdi";
mysql_select_db("book");
?>

```

### Jadval maydonlari ro'yxatini olish

Endi topshiriqni echish bilan alohida shug'ullansa bo'ladi. Jadval maydonlari ro'yxatini qanday olish mumkin? Juda oddiy. PHP da bu holda ham o'ziga tegishli buyruq mavjud - `mysql_list_fields`.

`mysql_list_fields` sintaksisi

```

mysql_list_fields ( resursi
    database_name qatori,
    table_name qatori
    [, resurs link_identifier])

```

Bu funktsiya maydonlar ro'yxatini `database_name` ma'lumotlar bazasidagi `table_name` jadvaliga qaytaradi. Kelib chiqadiki, ma'lumotlar bazasini tanlash bizga majburiy emas, lekin keyinroq asqotadi. Bu funktsiya ishining natijasini – resurs tipi o'zgaruvchisini qanday baholash mumkin. Ya'ni bu biz olishni xohlagan narsa emas. Bu ularning nomlari, tiplari va bayroqlari kiradigan jadval maydonlari haqidagi ma'lumotlarni olish uchun foydalanish mumkin bo'lgan havola. `mysql_field_name` funktsiyasi so'rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi. `mysql_field_len` funktsiyasi maydon uzunligini qaytaradi. `mysql_field_type` funktsiyasi maydon tipini qaytaradi, `mysql_field_flags` funktsiyasi esa probel bilan yozilgan maydon bayroqlari ro'yxatini qaytaradi. Maydon tiplari int, real, string, blob va b. bo'lishi mumkin. Bayroqlar not\_null, primary\_key, unique\_key, blob, auto\_increment va b. bo'lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

```

mysql_field_name (result qatori, butun field_offset) resursi;
mysql_field_type (result qatori, butun field_offset) resursi;
mysql_field_flags (result qatori, butun field_offset) resursi;
mysql_field_len ( result qatori, butun field_offset)

```

Bu erda *result* – bu so'rov natijasi identifikatori (masalan, `mysql_list_fields` yoki `mysql_query` funktsiyalar bilan jo'natilgan so'rov(bu haqda keyinroq gapirib o'tiladi)), `field_offset` esa – natijadagi maydonning tartib raqami. Umuman olganda, `mysql_list_fields` yoki `mysql_query` turidagi funktsiyalarni qaytaruvchi jadvalni, aniqrog'i uning unga bo'lgan ko'rsatkichini aks ettiradi. Bu jadvallardan aniq ma'nolar olish uchun maxsus bu jadvalni ostki qator sifatida o'quvchi funktsiyalarni ishga tushirish kerak. Bu funktsiyalarga `mysql_field_name` va shu kabilar ham kiradi. So'rovni amalga oshirish natijasi jadvalidagi barcha qatorlarni saralash uchun bu jadvaldagi qatorlar miqdorini bilish kerak.

`mysql_num_rows(result resursi)` buyrug'i result ning ko'pgina natijalari qatorlari miqdorini qaytaradi.

Endi esa Artifacts (eksponatlar kolleksiyasi) jadvali maydonlari ro'yxatini olishga urinib ko'ramiz.

```

<?
$conn = mysql_connect(
    "localhost","nina","123")
or die("Aloqa o'rnatib bo'lmaydi: ". mysql_error());
echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields (
    "book","Artifacts",$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 turi: ". $type;
echo "<br>Maydon uzunligi: ". $len;
echo "<br>Maydon bayroqlari qatori: ".
    $flags_str . "<br>";
}
?>

```

Natija sifatida taxminan quyidagilarni olish mumkin (albatta, jadvalda ikkita maydon bo'lganda):

```

Maydon nomi: id
Maydon turi: int
Maydon uzunligi: 11
Maydon bayroqlari qatori:
    not_null primary_key auto_increment
Maydon nomi: title
Maydon turi: string
Maydon uzunligi: 255
Maydon bayroqlari qatori:

```

### **html-formada maydonlar ro'yxatining aks etishi**

Endi biroz yuqoridagi misolni tekshirib ko'ramiz. Maydon haqidagi ma'lumotni shunchaki chiqarmaymiz, uni munosib *html-forma* elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o'tkazamiz (TEXT turida biz tuzgan description maydoni BLOB turiga ega ekanligini e'tibordan qochirmaymiz), raqamlar va qatorlarni <input type=text> kirish qismi matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma'nosi o'z-o'zidan o'rnatiladi.

Bularning barchasi anchayin oddiy hal qilinadi, bayroqlar ro'yxatidaig auto\_increment bayrog'i bundan mustasno. Buning uchun explode funktsiyasidan foydalaniladi:

*explode*: sintaksisi

```

explode massivi (separator qatori,
    string qatori [, int limit])

```

Bu funktsiya string qatorini separator taqsimlagichi yordamida qismlarga bo'ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsimlagich sifatida probel " " ni olish kera, bo'lish uchun boshlang'ich qator sifatida esa – maydon bayroqlari qatorini.

Shunday qilib, ma'lumotlarni Artifacts jadvaliga kiritish uchun forma tuzamiz:

```

<?
$conn=mysql_connect("localhost","nina","123");
    // aloqa o'rnatamiz
$dbase = "book";
$table_name = "Artifacts";
mysql_select_db($dbase); // ish uchun
    // ma'lumotlar bazasi tanlaymiz
$list_f = mysql_list_fields($dbase,$table_name);
    // bazadan maydonlar ro'yxatini olamiz
$n = mysql_num_fields($list_f); // avvalgi so'rovdagi natijadagi qatorlar soni (Artifacts jadvalida
    ja'mi qancha maydonlar bor)
echo "<form method=post action=insert.php>";
    // ma'lumotlarni kiritish uchun forma tuzamiz

```

```

echo "&nbsp;";<TABLE BORDER=0 CELLSPACING=0 width=50% ><tr>
  <TD BGCOLOR='#005533' align=center><font colorq'FFFFFF'>
    <b> Add new row in $table_name</b></font></td></tr><tr><td></td></tr></TABLE>";
echo "<table border=0 CELLSPACING=1 cellpaddingq0 widthq50% >";
// har bir maydon uchun uning nomi, turi, uzunligi va bayrog'ini olamiz
for($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);
  // bayroqlar qatoridan massiv qilamiz,
  // unda massivning har bir elementi – maydon bayrog'i
  $flags = explode(" ", $flags_str);
  foreach ($flags as $f){
    if ($f == 'auto_increment') $key = $name_f;
    // avtoinkrement nomini eslatib o'tamiz
  }
  /* avtoinkrement hisoblangan har bir maydon uchun, uning turidan kelib chiqib mos keluvchi
  forma elementini chiqaramiz*/
  if ($key <> $name_f){
    echo "<tr><td align=right bgcolorq'#C2E3B6'><font sizeq2>
      <b>&nbsp;". $name_f ."</b></font></td>";
    switch ($type){
      case "string":
        $w = $len/5;
        echo "<td><input type=text name=/'>$name_f/'>
          size = $w ></td>";
        break;
      case "int":
        $w = $len/4;
        echo "<td><input type=text name=/'>$name_f/'>
          size = $w ></td>";
        break;
      case "blob":
        echo "<td><textarea rowsq6 colsq60 name=/'>$name_f/'>></textarea></td>";
        break;
    }
  }
  echo "</tr>";
}
echo "</table>";
echo "<input type=submit name='add' value='Add'>";
echo "</form>";
?>

```

**Misol. Artifacts jadvaliga ma'lumot kiritish uchun forma**

## Ma'lumotlar bazasiga ma'lumotlar yozish

Shunday qilib forma tuzildi. Endi eng asosiysini bajarish qoldi – bu formadagi ma'lumotlarni bizning ma'lumotlar bazasiga jo'natish. Ma'lumki, ma'lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug'i ishlatiladi. Masalan:

```
mysql> INSERT INTO Artifacts
```

SET titleq'Kamolov';

PHP skriptda bunday buyruqdan (yoki SQL dagi istalgan buyruqdan) qanday foydalaniladi, degan savol tug'iladi. Buning uchun *mysql\_query()* funktsiyasi mavjud.

*mysql\_query sintaksisi*

*mysql\_query* resursi (query qatori

[, resurs link\_identifier])

*mysql\_query()* SQL-so'rovni MySQL ma'lumotlar bazasining link\_identifier ko'rsatkichi yordamida aniqlanadigan aktiv ma'lumotlar bazasiga jo'natadi (bu MySQL serveri bilan biron-bir aloqaga havola). Agar link\_identifier parametri o'tkazib yuborilgan bo'lsa, so'nggi ochiq aloqa ishlatiladi. Agar ochiq aloqa bo'lmasa, funktsiya parametrsiz *mysql\_connect()* funktsiyasiga o'xshash holda MBBT (SUBD) bilan bog'lanishga urinadi.

So'rov natijasi buferlanadi.

### Topshiriqlar:

#### SELECT OPERATORIGA MISOL.

<?php

```
$ulanish = new PDO("mysql:host=localhost;dbname=tatu","root","");
```

```
$talabalar = $ulanish -> query("SELECT t.* ,g.nomi AS gr FROM talabalar t JOIN guruh g ON t.guruh_id=g.id")
```

```
->fetchAll();
```

?>

<html>

<head>

</head>

<body>

<table border="2px" align="center">

<h2 align="center">TATU talabalari.</h2>

<tr>

<th>ID</th>

<th>Ism</th>

<th>Familya</th>

<th>Sharif</th>

<th>Tel number</th>

<th>manzil</th>

<th>Guruh\_ID</th>

</tr>

<?php foreach(\$talabalar as \$talaba):?>

<tr>

<th><?= \$talaba["id"]; ?></th>

<th><?= \$talaba["ism"]; ?></th>

<th><?= \$talaba["familya"]; ?></th>

<th><?= \$talaba["sharif"]; ?></th>

<th><?= \$talaba["tel"]; ?></th>

<th><?= \$talaba["manzil"]; ?></th>

<th><?= \$talaba["gr"]; ?></th>

```

</tr>
<?php endforeach;?>
</table>
</body>
</html>

```



#### TATU talabalari.

| ID | Ism       | Familiya   | Sharif         | Tel number    | manzil          | Guruh_ID |
|----|-----------|------------|----------------|---------------|-----------------|----------|
| 1  | Shixnazar | Sapoyev    | Omon o'g'li    | +998972110794 | Xonqa           | 911-13   |
| 2  | Hursand   | O'rinboyev | Husayn o'g'li  | +99999999     | Xiva            | 911-13   |
| 3  | Sanjar    | Sapoyev    | G'ayrat o'g'li | +998943151971 | Xonqa 7-maxalla | 912-13   |
| 4  | Arslon    | Rajabov    | Bekturdiyevich | +00000000     | Yangariq        | 911-13   |

#### INSERT operatoriga misol:

```

<?php
$conn = new PDO("mysql:host=localhost;dbname=tatu",'root','');
$guruhlar=$conn->query("SELECT * FROM guruh")->fetchAll();
if($_POST){
    $ism=$_POST["ism"];
    $fam=$_POST["familiya"];
    $shar=$_POST["sharif"];
    $tel=$_POST["tel"];
    $man=$_POST["manzil"];
    $guruh=$_POST["guruh"];
    $conn->prepare("INSERT INTO talabalar(ism,familiya,sharif,tel,manzil,guruh_id)
        VALUES (?,?,?,?,?,?)")
        ->execute(array($ism,$fam,$shar,$tel,$man,$guruh));
    header("Location: myklinika.uz/myklinika/guruh.php");
}
?>

```

```

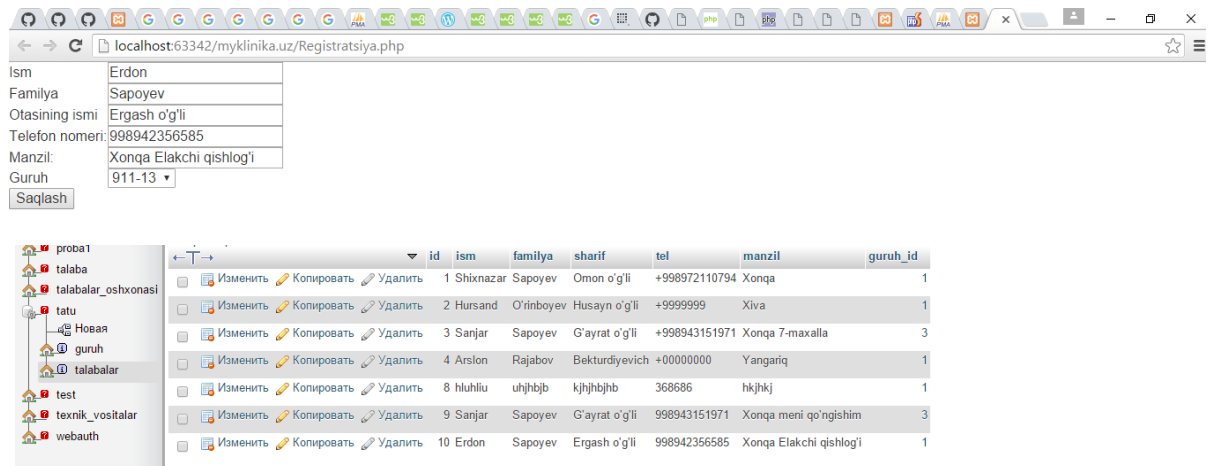
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Ro'yxatdan o'tish</title>
    <link rel="stylesheet" href="myklinika/bootstrap/css/bootstrap.min.css"/>
</head>
<body>
<form action="Registratsiya.php" method="post">
    <table>
        <tr>
            <td>Ism</td>
            <td><input type="text" name="ism"/></td>
        </tr>
        <tr>
            <td>Familiya</td>
            <td><input type="text" name="familiya"/></td>
        </tr>
        <tr>
            <td>Otasining ismi</td>

```

```

        <td><input type="text" name="sharif"/></td>
    </tr>
    <tr>
        <td>Telefon nomeri:</td>
        <td><input type="number" name="tel"/></td>
    </tr>
    <tr>
        <td>Manzil:</td>
        <td><input type="tel" name="manzil"/></td>
    </tr>
    <tr>
        <td>Guruh</td>
        <td>
            <select name="gur" >
                <?php foreach($guruhlar as $g):?>
                    <option value="<?=$g['id'];?>"><?=$g['nomi'];?></option>
                <?php endforeach;?>
            </select>
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" value="Saqlash" name="sbmt"/>
        </td>
    </tr>
</table>
</form>
</body>
</html>

```



UPDATE opetaroriga misol:

```

<?php
    $ulanish = new PDO("mysql:host=localhost;dbname=tatu","root","");
    $talabalar = $ulanish -> query("SELECT t.* ,g.nomi AS gr FROM talabalar t JOIN guruh g
    ON t.guruh_id=g.id")
    ->fetchAll();
?>

```

```

<html>
  <head>

  </head>
  <body>
    <a style="text-align: center" href="Registratsiya.php" >YANGI TALABA QO'SHISH</a>
    <table border="2px" align="center">
      <h2 align="center">TATU talabalari.</h2>
      <tr>
        <th>ID</th>
        <th>Ism</th>
        <th>Familya</th>
        <th>Sharif</th>
        <th>Tel number</th>
        <th>manzil</th>
        <th>Guruh_ID</th>
        <th>Taxrirlash</th>
      </tr>
      <?php foreach($talabalar as $talaba):?>
      <tr>
        <td><?= $talaba["id"]; ?></td>
        <td><?= $talaba["ism"]; ?></td>
        <td><?= $talaba["familya"]; ?></td>
        <td><?= $talaba["sharif"]; ?></td>
        <td><?= $talaba["tel"]; ?></td>
        <td><?= $talaba["manzil"]; ?></td>
        <td><?= $talaba["gr"]; ?></td>
        <td>
          <a href="Update.php?id=<?=$talaba['id']?>"><button>Taxrirlash</button></a>
        </td>
      </tr>
      <?php endforeach;?>
    </table>
  </body>
</html>

```



localhost x PMA local x localhost x localhost x localhost x localhost x

localhost:63342/myklinika/lobaratoriya/LOB\_12/guruh.php

[YANGI TALABA QO'SHISH](#)

### TATU talabalari.

| ID | Ism       | Familiya   | Sharif          | Tel number    | manzil                  | Guruh_ID | Taxrirlash                 |
|----|-----------|------------|-----------------|---------------|-------------------------|----------|----------------------------|
| 1  | Shixnazar | Sapoyev    | Omon o'g'li     | +998972110794 | Xonqa                   | 911-13   | <a href="#">Taxrirlash</a> |
| 2  | Hursand   | O'rinboyev | Husayn o'g'li   | +99999999     | Xiva                    | 911-13   | <a href="#">Taxrirlash</a> |
| 4  | Arslon    | Rajabov    | Bekturdiyevich  | +00000000     | Yangariq                | 911-13   | <a href="#">Taxrirlash</a> |
| 10 | Erdon     | Sapoyev    | Ergash o'g'li   | 998942356585  | Xonqa Elakchi qishlog'i | 911-13   | <a href="#">Taxrirlash</a> |
| 11 | Komiljon  | Avezov     | Baxtiyor o'g'li | 998972111111  | Xonqa Andalib           | 911-13   | <a href="#">Taxrirlash</a> |
| 3  | Sanjar    | Sapoyev    | G'ayrat o'g'li  | +998943151971 | Xonqa 7-maxalla         | 912-13   | <a href="#">Taxrirlash</a> |
| 9  | Sanjar    | Sapoyev    | G'ayrat o'g'li  | 998943151971  | Xonqa meni qo'ngishim   | 912-13   | <a href="#">Taxrirlash</a> |

Tahrirlashni bosgandan so'ng shu talaba id sini Update.php ga yuboradi.

Update.php ni kodi:

```
<?php
$conn = new PDO("mysql:host=localhost;dbname=tatu","root","");
$guruhlar = $conn->query("SELECT * FROM guruh");
$id=$_GET['id'];
$student = $conn->query("SELECT * FROM talabalar WHERE id=$id")->fetch();
if($_POST){
    $ism=$_POST["ism"];
    $fam=$_POST["familya"];
    $shar=$_POST["sharif"];
    $tel=$_POST["tel"];
    $man=$_POST["manzil"];
    $guruh=$_POST["gur"];
    $conn->prepare("UPDATE talabalar SET ism=?, familya=?,sharif=?,tel=?,
manzil=?,guruh_id=? WHERE id=$id")
    ->execute(array($ism,$fam,$shar,$tel,$man,$guruh));
}
?>
<html>
<head>

</head>
<body>
<a href="Registratsiya.php">Yangi talaba qo'shish</a>
```

```

<form action="Update.php" method="POST">
  <table>
    <tr>
      <td>Ism</td>
      <td><input type="text" name="ism" value="<?=$student['ism']?>" /></td>
    </tr>
    <tr>
      <td>Familyasi</td>
      <td><input type="text" name="familya" value="<?=$student['familya']?>" /></td>
    </tr>
    <tr>
      <td>Sharifi</td>
      <td><input type="text" name="sharif" value="<?=$student['sharif']?>" /></td>
    </tr>
    <tr>
      <td>Telefon no'meri</td>
      <td><input type="text" name="tel" value="<?=$student['tel']?>" /></td>
    </tr>
    <tr>
      <td>Manzili</td>
      <td><input type="text" name="manzil" value="<?=$student['manzil']?>" /></td>
    </tr>
    <tr>
      <td>Guruhi</td>
      <td>
        <select name="gur">
          <?php foreach($guruhlar as $g):?>
            <option value="<?=$g['id'];?>" <?php if($g['id']==$student['guruh_id']) echo
"selected";?>><?=$g['nomi'];?></option>
          <?php endforeach;?>
        </select>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="Saqlash" name="sa=" />
      </td>
    </tr>
  </table>
</form>
</body>
</html>

```

localhost x

PMA localhost x

localhost x

localhost x

localhost x

localhost

← → ↻

localhost:63342/myklinika/lobaratoriya/LOB\_12/Up

[Yangi talaba qo'shish](#)

Ism

Shixnazar

Familyasi

Sapoyev

Sharifi

Omon o'g'li

Telefon no'meri

+998972110794

Manzili

Xonqa

Guruhi

911-13 ▾

Saqlash

## Tajriba ishi №13

### Mavzu: PHPda sinflar yaratish. Ob'yekt tushunchasi. Sinf maydonlari va metodlari.

**Ishdan maqsad:**PHP da sinflar yaratishni va Ob'yekt tushunchasini o'rganish.

#### Nazariy qism

Sinflar va ob'ektlar.

Ob'ektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamada ob'ektlardir. Bu tushunchalarni quyidagicha tushunish mumkin: ob'ekt – bu dasturda qo'llaniladigan tushuncha yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa bu ob'ektlarning tavsifi va ular ustida bajariladigan harakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi:

```
class Sinf_nomi{
    var $xususiyat_nomi;
    /*xususiyatlar ro'yxati*/
    function metod_nomi(){
        /* usullarning tanasi */
    }
    /*usullar ro'yxati*/
}
```

Sinf ob'ektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sinf ob'ektlariga qo'llanilgan usullar funktsiya sifatida ishlatiladi. Sinf tanasi ichida this kalit so'zi yordamida taqdim qilinayotgan joriy sinfga murojaatni amalga oshirish mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sinf tuzishimiz kerak. Har bir maqolaning nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda ko'rsatishimiz kerak bo'ladi. Shunda bu sinfning ifodalanishi quyidagicha holatda bo'ladi:

```
<?
class Articles { // Maqola sinfini tuzamiz
    var $title;
    var $author;
    var $description;
    // maqola atributi
    // ma'nosini o'zlashtiruvchi usul
    function make_article($t, $a, $d){
        $this->title = $t;
        $this->author = $a;
        $this->description = $d;
    }
    //sinf nusxasini ifodalash uchun usul
    function show_article(){
        $art = $this->title . "<br>" .
            $this->description .
            "<br>Muallif: " . $this->author;
        echo $art;
    }
}
?>
```

Shunday qilib “maqola” turidagi fizik ob’ektlarni tasvirlash uchun biz uch o’zgartuvchidan tashkil topgan, maqola xarakteristikasini o’zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funktsiya tuzdik.

Ma’lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo’laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo’yicha va qolaversa har xil fayllar bo’yicha tarqatilmasligi kerak. Ya’ni quyidagicha yozsak:

```
<?php
class Articles { // Sinfni tasvirlashning boshi
    var $title;
?>
<?php
// sinfni tasvirlashning davomi
function show_article(){
    // usulning tarkibi
}
} // sinfni tasvirlashning yakuni
?>
```

bunda dastur tartibli ishlaydi.

Sinf nomi masalasida ayrim narsalarni e’tiborda tutish kerak. Sinfning nomi PHP tilidagi ob’ektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o’z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “\_” quyi chiziqdan boshlanuvchilardir.

Sinflar va funktsiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatiladi.

#### O’zgaruvchilarni initsiallashtirish

Ba’zan ayrim sinf atributlariga ma’noni sinf ishtirokchisini tuzish bilanoq o’zlashtirish kerak bo’ladi. Biz maqola sinfini tuzganimizda, sinf atributlari (xususiyatlari) ma’nolarini o’zlashtirish uchun maxsus funktsiya make\_article() dan foydalandik. Umuman olganda, biz to’g’ri yo’l tutmadik, chunki “velosiped ixtirosi” bilan shug’ullandik. Sinf atributlarining boshlang’ich ma’nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma’noni var operatori yoki konstruktor funktsiyasi yordamida initsiallashtirish mumkin. var yordamida faqat konstant ma’nolarni initsiallashtirish mumkin. Konstant bo’lmagan ma’nolarni berish uchun ob’ekt sinfdan ajrab chiqqanda o’z-o’zidan ishga tushuvchi konstruktor funtsiyasidan foydalaniladi. Konstruktor-funktsiya u ifodalangan butun sinfga mos keluvchi nomga ega bo’lishi kerak.

Misol keltiramiz. Aytaylik, “maqola” ob’ektini tuzishda biz uning xususiyatlarini quyidagicha belgilaylik: mualliflar – “Kamolov” qatoriga teng, nomlanish va qisqa mazmun - \$\_POST global massivi elementlariga mos, maqola nashri – mazkur sanada. Unda sinfning quyidagi ifodasi PHP4da batartib bo’lmaydi:

```
<?
class Articles { // maqola sinfini tuzamiz
    var $titleq $_POST["title"];
    var $author = "Kamolov";
    var $description = $_POST["description"];
    var $published = date("Y-m-d");
// sinf atributi
// ma’nosini o’zlashtiruvchi usul
```

```

    }
?>
Sinfning quyidagi ifodasi esa PHP4da keraklicha yo'sinda ishlaydi:
<?
class Articles { // maqola sinfini tuzish
    var $title;
    var $author = "Kamolov";
    var $description;
    var $published;
// sinf atributi
// ma'nosini o'zlashtiruvchi usul
function Articles(){
    $this->title = $_POST["title"];
    $this->description = $_POST["description"];
    $this->published = date("Y-m-d");
}
}
?>

```

PHP3 va PHP4 da konstruktorlar har xil ishlashini hisobga olish kerak. Funktsiya PHP3 da, agar u sinfniki kabi nomga ega bo'lsa, konstruktorga aylanadi, PHP4 da esa – agar u ifodalar sinfniki kabi nomga ega bo'lsa shunday bo'ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko'rinib turibdi. Lekin biz bu haqda biroz keyinroq gapiramiz. PHP5da sinf konstruktori `_construct` deb nomlanadi. Bundan tashqari, PHPda destruktorga – ob'ektni yo'q qilishda o'z-o'zidan ishga tushuvchi funktsiyalar paydo bo'ldi. PHP5 da funktsiya-destruktorga `destruct` deb nomlanishi kerak bo'ladi.

### Ob'ektlar

Birinchi ma'ruzalardan birida biz PHP da ob'ekt deb ataluvchi tip mavjudligi haqida aytib o'tgandik. Sinf – bu ob'ekt tipidagi ma'lumotlarning bir turidagi ifodalanishidir. Sinflar real o'zgaruvchilar uchun shablon vazifasini o'taydi. Kerakli tipdagi o'zgartuvchi `new` operatori yordamida sinfdan tuziladi. Ob'ektni tuzib, biz barcha usullarni qo'llashimiz va barcha sinf ifodasida ko'rsatib o'tilgan xususiyatlarni olishimiz mumkin bo'ladi. Buning uchun quyidagicha sintaksisdan foydalaniladi: `$ob'ekt_nomi->xususiyat` yoki `_nomi$ob'ekt_nomi->usulning_nomlanishi(argumentlar ro'yxati)`. Xususiyatlar yoki usullar nomlari oldidan `$` belgisi qo'yilmaydi.

```

<?php
$art = new Articles;
    // ob'ekt tuzamiz $art
echo ($art->title);
    // ob'ekga nomlanish beramiz $art
$another_art = new Articles;
    // ob'ekt tuzamiz $another_art
$another_art->show_article();
    // ob'ektning brauzerdagi ifodasi uchun
    // usulni chaqiramiz
?>

```

Misol: Ob'ekt usullari va xususiyatlariga erkin kirish (dostup)

Sinfning har bir ob'ekti aynan bir xil xususiyatlar va usullarga ega bo'ladi. Demak, `$art` ob'ektda va `$another_art` ob'ektda `title`, `description`, `author` xususiyatlari va `Articles()`,

show\_article() usullari mavjud. Lekin bular ikki xil ob'ektlar. Ob'ektni fayllar sistemasidagi direktoriya deb hisoblaymiz, uning xarakteristikasi esa – bu direktoriyadagi fayllar singari bo'lsin. Aniqki, har bir direktoriyada bir xil fayllar yotishi mumkin, lekin shunday bo'lsa-da, ular har xil direktoriyalarda saqlanayotgani uchun har xil hisoblanishi mumkin. Xuddi shuningdek, xususiyatlar va usullar ham, agar ular turli ob'ektlarga qo'llaniladigan bo'lsa, har xil hisoblanadi. Yuqori bosqichdagi direktoriyadan kerakli faylni olish uchun bu faylga yo'lni batafsil yozib chiqamiz. Sinflar bilan ishlash mobaynida biz chaqirishni istagan funktsiyaning nomini to'liq yozishimiz kerak bo'ladi. PHP dagi yuqori bosqich direktoriyalariga global o'zgaruvchilarning bo'sh o'rni bo'ladi, yo'l esa -> taqsimlovchisi yordamida ko'rsatiladi. Shu tarzda \$art->title va \$another\_art->title nomlari ikki xil turli o'zgaruvchilarni anglatadi. PHP da o'zgaruvchi nom oldidan faqat bitta dollar belgisiga ega bo'ladi, shuning uchun \$art->\$title ko'rinishida yozish mumkin emas. Bu konstruktsiya \$art ob'ektining title xususiyatiga murojaat sifatida ko'rib chiqilmaydi, \$title o'zgartuvchi ko'rinishida berilgan nomli xususiyat sifatida ko'riladi (masalan, \$art->"").

```
<?php
```

```
$art->title = " Internet ga kirish";
```

```
// ob'ekt xususiyati ma'nosini
```

```
// shunday o'rnatish mumkin
```

```
$art->$title = "Internet ga kirish";
```

```
// ob'ekt xususiyati ma'nosini
```

```
// bunday o'rnatib bo'lmaydi
```

```
$property = "title";
```

```
$art->$property = "Internet ga kirish";
```

```
// ob'ekt xususiyati ma'nosini
```

```
// shunday o'rnatish mumkin
```

```
?>
```

Misol: Xususiyatlar ma'nosini o'rnatish

Sinfni tuzib, bu sinfnings ob'ekti qanday nomga ega bo'lishini bila olmaymiz, qolaversa ob'ektlar juda ko'p bo'lishi va ularning barchasi har xil nomga ega bo'lishi mumkin. Sinfni yuzaga chiqarish ichida ob'ektga qanday munosabatda bo'lishni bilmaymiz. Sinf yuzaga chiqishi ichida funktsiyalar va o'zgaruvchilarga erkin kirish uchun, \$thid o'rindosh o'zgaruvchisidan foydalanish kerak. Masalan, \$this->title shunday sinf ob'ektining title ini qaytaradi. Ba'zan bu o'zgaruvchini "mening xususiy mulkim" (xususiyatga munosabat tariqasida) deb o'qishni taklif qilinadi.

### Topshiriqlar:

Helper.php

```
<?php
```

```
class Helper {
```

```
static function add($pdo,$ism="", $fam="", $ota="",$tel="",$manzil="",$guruh=""){
```

```
$pdo->prepare("INSERT INTO talabalar(ism,familiya,sharif,tel,manzil,guruh_id)
```

```
VALUES(?,?,?,?,?,?)"
```

```
->execute(array($ism,$fam,$ota,$tel,$manzil,$guruh));
```

```
}
```

```
static function update($pdo,$id,$ism="", $fam="", $ota="", $tel="", $manzil="", $guruh=""){
```

```
$pdo->prepare("UPDATE talabalar SET
```

```
ism=?,familya=?,sharif=?,tel=?,manzil=?,guruh_id=?
```

```
WHERE id=?)"
```

```
->execute(array($ism,$fam,$ota,$tel,$manzil,$guruh,$id));
```

```

}

static function delete($pdo,$id){
    $pdo->exec("DELETE FROM talabalar WHERE idq$id");
}

static function getTalabalar($pdo){
    return $pdo->query("SELECT s.*,g.nomi AS guruh_nomi
        FROM talabalar s
        JOIN guruh g ON s.guruh_idqg.id")->fetchAll(PDO::FETCH_ASSOC);
}

static function getTalaba($pdo,$id){
    return $pdo->query("SELECT * FROM talabalar WHERE idq$id")->fetch();
}

static function getGuruhlar($pdo){
    return $pdo->query("SELECT * FROM guruh")
        ->fetchAll(); }}

```

Index.php

```

<?php
    include "Helper.php";
    $dsn = "mysql:host=localhost;dbname=tatu";
    $conn = new PDO($dsn,'root','');
    $students = Helper::getTalabalar($conn);
?>
<!doctype html>
<html>
    <head>

    </head>
    <body>
    <a href="add.php">Qo'shish</a>
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Familiya</th>
            <th>Ism</th>
            <th>Otasining ismi</th>
            <th>Manzili</th>
            <th>Tel nomer</th>
            <th>Guruh</th>
            <th>Amallar</th>
        </tr>
        <?php foreach($students as $student):?>
        <tr>
            <td><?= $student['id'];?></td>
            <td><?= $student['familya'];?></td>

```



```

<td><?= $student['ism'];?></td>
<td><?= $student['sharif'];?></td>
<td><?= $student['manzil'];?></td>
<td><?= $student['tel'];?></td>
<td><?= $student['guruh_nomi'];?></td>
<td>
    <a href="update.php?idq<?= $student['id']?>">
        <button>Tahrirlash</button>
    </a>
    <a href="delete.php?idq<?= $student['id']?>">
        <button>O'chirish</button>
    </a>
</td>
</tr>
<?php endforeach;?>
</table>
</body>
</html>

```

### Update.php

```

<?php
include "Helper.php";
$dsn = "mysql:host=localhost;dbname=tatu";
$conn = new PDO($dsn,'root','');
$groups=Helper::getGuruhlar($conn);

$Sid = intval($_GET['id']);
$student = Helper::getTalaba($conn,$id);
if($_POST){
    $ism = $_POST['ism'];
    $fam = $_POST['fam'];
    $ota = $_POST['ota_ism'];
    $manzil = $_POST['manzil'];
    $tel = $_POST['tel'];
    $guruh = $_POST['guruh'];
    Helper::update($conn,$id,$ism,$fam,$ota,$tel,$manzil,$guruh);
    header("Location: index.php");
}

?>
<html>
<head>
</head>
<body>
<form method="post">
    <table>
        <tr>
            <td>Ism</td>

```

```

        <td><input name="ism" type="text" value="<?=$student['ism']?>"></td>
    </tr>
    <tr>
        <td>Familiya</td>
        <td><input name="fam" type="text" value="<?=$student['familiya']?>"></td>
    </tr>
    <tr>
        <td>Otasining ismi</td>
        <td><input name="ota_ism" type="text" value="<?=$student['sharif']?>"></td>
    </tr>
    <tr>
        <td>Manzil</td>
        <td><textarea name="manzil"> <?=$student['manzil']?> </textarea></td>
    </tr>
    <tr>
        <td>Tel nomer</td>
        <td><input name="tel" type="tel" value="<?=$student['tel']?>"></td>
    </tr>
    <tr>
        <td>Guruh</td>
        <td>
            <select name="guruh">
                <?php foreach($groups as $g):?>
                    <option value="<?=$g['id'];?>" <?php if($g['id']==$student['guruh_id']) echo
"selected";?> ><?=$g['nomi'];?></option>
                <?php endforeach;?>
            </select>
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" value="Saqlash" name="sbmt">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

## Tajriba ishi №14-15-16-17

**Mavzu: MVC loyihalash shablони. Zamonaviy PHP frameworklar: Zend, Yii, CodeIgniter, Laravel.**

**Ishdan maqsad:** MVC loyihalash shablони haqida tushunchaga ega bo'lish. Zamonaviy PHP frameworklarni strukturasini o'rganish.

MVC (Model View Controller, Model Ko'rinish Nazorat) nima?

Bu umumiy tushuncha bo'lib, loyihalash shablonidir.

MVC jarayoni

Model bu — ma'lumotlar bazasidagi biror bir jadval (table) bilan ishlaydigan obyekt (class) dir.

View bu — HTML kodlarni yozasidan fayl yoki foydalanuvchiga (browserda) ko'rinadigan kod.

Controller bu — foydalanuvchi so'roviga mos holda kerakli model bilan ishlab, kerakli viewni foydalanuvchi jo'natadi.

Framework bu — web sayt qilish jarayonida qiyinchilik tug'diradigan yoki ko'p takrorlanadigan funksiyalar to'plamidir. Umuman olganda dasturchiga yordamchi funksiyalar.

CMS — Content Management System bu deyrali tayyor saytdir.

Zend – Zend kompaniyasi tomonidan ishlab chiqilgan va qo'llab-quvvatlanadigan PHP framework. Zend nomi uning ishlab chiquvchilari Zeev Suraski va Andy Gutmans ismlaridan olingan.

Zendning o'ziga xos jihatlari:

- Barcha komponentlar to'liq ob'yektga mo'ljallangan, PHP 5 standartlariga javob beradi,
- Komponentlarning bir biriga kam bog'liqligi, Dependency Injection'ga mosligi.
- MVCni kengayirilgan qo'llab quvvatlashi, layout'lar va PHPga asoslangan shablonlarning mavjudligi
- Bir qancha MBBTlarni qo'llab-quvvatlashi: MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, va Informix Dynamic Server
- Email jo'natish, mbox, Maildir, POP3 va IMAP4larning implementatsiyasi
- Moslashuvchan keshlash: fayl sistemasi yoki MB orqali.

Yii – bu ko'p tarmoqli web ilovalar qurish uchun yuqori sifatli komponentlar asosidagi PHP – framework strukturasidir. U katta kodlardan qayta va qayta foydalanishda moslashuvchanlikni va web ilovani tezkorligini oshirishiga imkon beradi. Qisqa qilib aytganda oddiy, samarali va keng ko'lamli php – framework.

Keling PHP frameworklarning o'zaro imkoniyatlarini chamalangan holda ko'rib chiqaylik.

PHP framework	PHP_4	PHP_5	MVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules	EDP
Akelos	+	+	+	+	+	+	+	+	+	+	+	+	-
ash.MVC	-	+	+	-	-	+	+	-	+	-	+	+	-
Cake PHP	+	+	+	+	+	+	-	+	+	+	+	+	-

<b>CodeIgniter</b>	+	+	+	+	-	+	+	+	+	-	-	-	-
<b>DIY</b>	-	+	+	-	+	+	+	+	-	+	-	-	-
<b>eZ components</b>	-	+	-	+	-	+	+	+	+	-	-	-	-
<b>Fusebox</b>	+	+	+	+	-	-	-	+	-	+	-	+	-
<b>PHP on TRAX</b>	-	+	+	+	+	+	-	-	+	+	-	+	-
<b>PHPDev Shell</b>	-	+	+	-	+	+	+	+	+	+	+	+	-
<b>PHP Openbiz</b>	-	+	+	+	+	+	+	-	+	+	+	-	-
<b>Prado</b>	-	+	+	+	+	+	+	+	+	+	+	+	+
<b>QPHP</b>	+	+	+	+	-	+	+	-	+	+	+	+	+
<b>Seagull</b>	+	+	+	+	+	+	+	+	+	+	+	+	-
<b>Symfony</b>	-	+	+	+	+	+	-	+	+	+	+	+	-
<b>WACT</b>	+	+	+	+	-	+	+	-	+	-	-	+	-
<b>WASP</b>	-	+	+	-	-	+	+	-	+	+	+	+	-
<b>Yii</b>	-	+	+	+	+	+	+	+	+	+	+	+	<b>Q</b>
<b>Zend</b>	-	+	+	+	+	+	+	+	+	+	+	+	-
<b>ZooP</b>	+	+	+	+	-	+	+	+	+	+	+	-	-

- **MVC:** Model-View-Controller arxitekturasini bo'yicha qurish imkoniyatiga ega.
  - **Multiple DB's:** Bir ma'lumotlar omboridan boshqa bir qancha ma'lumotlar omboriga o'zgartirishlarsiz amalga oshirilishi xususiyatiga ega.
  - **ORM:** ActiveRecord da ma'lumotlar obyekti bog'lanishi arxitekturasini amalga oshirish.
  - **DB Objects:** Ma'lumotlar ombori bilan ishlash xususiyatiga ega. ( Insert, Delete, Update kabilar)
  - **Templates:** Qurilgan shablonlar mexanizmi frameworkiga ega.
  - **Caching:** Berilgan obyektlarni keshlash yoki shu kabi boshqa uslubdagi keshlashga ega.
  - **Validation:** Ma'lumotlar oqimini tekshirish yoki filtrlash xususiyatiga ega.
  - **Ajax:** Frameworkda ajax texnologiyasi orqali ishlash xususiyatiga ega.
  - **Auth Module:** Foydalanuvchilarni authentication qilishda o'zining maxsus moduliga ega.
  - **Modules:** Frameworkning o'zida maxsus komponentlarga ega (misol uchun: PDF moduli, RSS moduli). Ya'ni foydali komponentlar jamlanmasi
  - **EDP:** Event Driven Programming. Eventlar bilan ishlash xususiyatiga ega. Misol uchun jquery kutubxonasining click, dbclick eventlari kabi.
- Qo'shimcha ma'lumot: Har bir framework o'zining biznes logikasiga ega va bu logika frameworkning qaysidir qismida yotadi. Yiining biznes logikasi Modelda joylashgan. Biznes logika deganda – qurilayotgan web ilovaning asosiy qismini tashkil qiluvchilari tushuniladi. Masalan: Validation, beforeSave, afterSave kabilar shular jumlasidan.

Yii DAO [PHP Data Objects \(PDO\)](#) ustiga qurilgan bo'lib, ko'pgina ma'lumotlar omboriga kirish xuquqi uchun foydalaniladi, misol uchun MySQL, PostgreSQL va shu kabilar. Shu sababli Yii DAO dan foydalanishda PDO (dynamic library) dinamik kutubxonasi

o`rnatilgan bo`lishi va foydalanilayotgan ma`lumotlar omboridagi PDO driveri bilan mos bo`lishi kerak.(Misol uchun PDO\_MYSQL)

Yii DAO 4 calassdan tashkil topgan:

- [CdbConnection](#): ma`lumotlar omboriga bog`lanishni ifodalaydi.
  - [CdbCommand](#): ma`lumotlar omborida amalga oshishi kerak bo`lgan so`rovni ifodalaydi.
  - [CdbDataReader](#): So`rovga javob qaytaruvchi bir yo`nalishdagi ma`lumotlar satri oqimini ifodalaydi..
  - [CdbTransaction](#): Ma`lumotlar ombori tranzaksiyasini ifodalaydi.
- Keyinchalik turli xil misollarda Yii DAO dan foydalanishni ko`rib chiqamiz.

### Ma`lumotlar ombori bilan bog`lanish.

Ma`lumotlar ombori bilan bog`lanishni o`rnatish uchun [CdbConnection](#) classini hosil qilishimiz va faollashtirishimiz kerak bo`ladi. Qo`shimcha ma`lumot : ma`lumotlar omboriga bog`lanish uchun

(host, port,foydalanuvchi nomi, - 76 -ttrib, va boshqalarini) va DSN (Data Source Name) ni ko`rsatib qo`yish kerak bo`ladi.

```
$connection=new CdbConnection($dsn,$username,$password);  
// bog`lanishni o`rnatamiz  
// try...catch mumkin bo`lgan qoidalarni to`xtatish uchun konstruktoridan foydalanamiz  
$connection->active=true;  
...  
$connection->active=false; // bog`lanishni yopish
```

DSN formati foydalanilayotgan PDO driveriga bog`liq. DSN ni qurishda foydalanilayotgan driver bog`lanishi sintaksisiga mos PDO - 76 -ttrib nomi va ikki nuqtadan keyin bog`lanish parametrlari ko`rsatiladi.Batafsil [PDO](#) bu yerdan tanishishingiz mumkin.Quyida bir qancha asosiy DSN formatlari ko`rsatilgan:

- SQLite: sqlite:/path/to/dbfile
- MySQL/MariaDB: mysql:host=localhost;dbname=testdb
- PostgreSQL: pgsql:host=localhost;port=5432;dbname=testdb
- SQL Server: mssql:host=localhost;dbname=testdb
- Oracle: oci:dbname=//localhost:1521/testdb

Haqiqiy [ilova komponentlarida](#) undan foydalanishimiz mumkin, shu boisdan [CdbConnection](#) I [CapplicationComponent](#) class vorisidir. Bu uchun db komponentini quyidagicha holatda ilova konfiguratsiyasini qilish kerak:

```
array(  
    ...  
    'components'=>array(  
        ...  
        'db'=>array(  
            'class'=>'CdbConnection',  
            'connectionString'=>'mysql:host=localhost;dbname=testdb',
```

```

        'username'=>'root',
        'password'=>'password',
        'emulatePrepare'=>true, // MySQL ning bir qancha ilovalari uchun kerak
    ),
),
)

```

Endi biz `Yii::app()->db` orqali ma'lumotlar ombori bilan bog'lanishga kirish xuquqini olishimiz mumkin. Faol bo'lmagan bog'lanishlar uchun [CdbConnection::autoConnect](#) da false belgisini o'rnatishimiz kerak. Bu usul kodning ixtiyoriy joyidan ma'lumotlar ombori bilan bog'lanishga imkon beradi.

### SQL so'rovlardan foydalanish

Qachonki biz ma'lumotlar omboriga ulansak unda biz [CdbCommand](#) dan foydalanib SQL so'rovini bajarishimiz mumkin. Buning uchun [CdbCommand](#) class nusxasini hosil qilib [CdbConnection::createCommand\(\)](#) orqali SQL so'rovini ko'rsatishimiz mumkin bo'ladi:

```

$connection=Yii::app()->db; // - 77 -tribute orqali "db" bilan bog'lanish
// Ochiqdan ochiq murojaat uchun bu usulni ham qo'llash mumkin
// $connection=new CdbConnection($dsn,$username,$password);

```

```

$command=$connection->createCommand($sql);
// bizga kerak bo'lgan SQL so'rovlar berishimiz mumkin
// $command->text=$newSQL;

```

[CdbCommand](#) dan foydalanish bilan SQL so'rovlarni bajarishning ikki xil usul mavjud:

- [execute\(\)](#): INSERT, UPDATE va DELETE SQL so'rovlarini ishlatish uchun.
- [query\(\)](#): qaytuvchi ma'lumotlar oqiminig SQL so'rovlarini bajarishda, misol uchun SELECT so'rovi. Agar so'rov muvofaqiyatli bajarilsa [CdbDataReader](#) classi nusxasi yordamida qaytgan ma'lumotlarni olish xuquqiga ega bo'lamiz.

Agar jarayon paytida qandaydir xatolik bo'lsa unda bu jarayonni amalga oshirmaydi. Ya'ni jadvalda sizga halaqit qiluvchi va dasturni bizuvchi qoldiqlar qoldirmaydi.

```

$rowCount=$command->execute(); // `INSERT`, `UPDATE` yoki `DELETE` turidagi
so'rovlarni bajarish uchun

```

```

$dataReader=$command->query(); // `SELECT` turidagi so'rovlarini bajarish uchun

```

```

$rows=$command->queryAll(); // so'rov natijasidagi barcha satrlarni olish

```

```

$row=$command->queryRow(); // so'rov natijasidagi birinchi satrni qaytaradi.

```

```

$column=$command->queryColumn(); // so'rov natijasidagi birinchi ustunni qaytaradi.

```

```
$value=$command->queryScalar(); //birinchi maydondagi birinchi satrni qaytaradi ya`ni  
birinchi katakcha qiymatini.
```

### So`rov natijalari bilan ishlash

[CdbDataReader](#) classi nusxasidagi [CdbCommand::query\(\)](#) orqali natijalarni yig`amiz va [CdbDataReader::read\(\)](#) usuli orqali natijalarni satrma satr o`qiyamiz. Ma`lumotlarni satrma satr o`qib olishimiz uchun shuningdek foreach orqali [CdbDataReader](#) dan foydalanamiz.

```
$dataReader=$command->query();  
// ko`p martalik read() murojaat bo`lib toki false qiymat qaytarmaguncha ishlaydi. Ya`ni olingan  
barcha ma`lumotlar hammasi tugaguncha  
  
while(($row=$dataReader->read())!qqfalse) { ... }  
//ma`lumotlarga satrma satr murojaat uchun foreach dan foydalanamiz.  
Foreach($dataReader as $row) { ... }  
// barcha satrlarni massivga yig`amiz  
$rows=$dataReader->readAll();
```

**Izoh:** Barcha queryXXX() larning [query\(\)](#) dan farqi to`g`ridan to`g`ri ma`lumot qaytarishidadir. Misol uchun , [queryRow\(\)](#) metodi joriy satr natijasi bo`yicha massiv qaytaradi.

### Tranzaksiyadan foydalanish

Bir qancha so`rovlarni bir vaqtning o`zida bajarilishini, ya`ni hamma so`rovlar aniq bajarilishi kerak, biri bajarilmay qolgan holatda qolgan barcha bajarilgan so`rovlarni ortga qaytarib asl holiga qaytarib qo`yish kerak. Bunday holatda tranzaksiya ishlatiladi va [CdbTransaction](#) classidan foydalaniladi. [CdbTransaction](#) classini ishlash usuli quyidagicha:

- Tranzaksiya boshlanadi.
- So`rovlar birma bir amalga oshiriladi. Bunday holatda tranzaksiya o`zgargan ma`lumotlarni keshida saqlaydi, ma`lumotlar omborida xech qanday o`zgarish bo`lmaydi. Lekin ma`lumotlar ombori qaytarishi kerak bo`lgan ma`lumotlarni qaytaradi. Ya`ni siz saqlayotgan \$sql1 satrning id si kerak bo`lsa va bu - 78 - ttrib \$sql2 jadvalingiz uchun tashqi kalit sifatida kerak bo`lsa sizga ushbu idni qaytara oladi garchi ma`lumotlar omboriga saqlanmagan bo`lsa ham.
- Agarda tranzaksiyadagi so`rovlar muvofaqiyatli bo`lsa tranzaksiya amalga oshadi va ma`lumotlar omboridagi ma`lumotlarga o`zgarishlar kiradi.
- Agarda so`rovlar paytida qandaydir xatolik ro`y bersa, barcha so`rovlarni ortga qaytarib qo`yadi.

Bir kichik ko`rinishi quyidagicha:

```
$transaction=$connection->beginTransaction();  
try  
{  
    $connection->createCommand($sql1)->execute();  
    $connection->createCommand($sql2)->execute();  
    // SQL-so`rovlari  
    $transaction->commit();  
}
```

```
catch(Exception $e)
{
    $transaction->rollback();
}
```

**Model** - foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni doimo rolda nazorat qilib boradi. Biz modelni ikki xil ko'rinishda hosil qilishimiz mumkin. Biz ma'lumotlarni olish, qayta ishlash, keyin o' - 79 -ttribu uchun [CformModel](#) dan foydalanamiz; agar ma'lumotni olib uni bazada saqlash bo'ladigan bo'lsa CActiveRecord dan foydalaniladi. Ikkala tur ham bir Cmodel classi asosida ishlaydi.

“**attribute**” – termini classning ayrim xususiyatlarini boshqa xususiyatlaridan ajratish uchun yaratilgan. Bunday xususiyat (- **79 -ttribute**) asosan o'zida ma'lumotlar omboridan olingan yoki foydalanuvchi tomonidan kiritilgan ma'lumotni saqlashda qo'llaniladi.

### **Model classini aniqlaymiz:**

Quyida biz LoginForm modelidan foydalanib, foydalanuvchining ro'yhatdan o'tish pageni yaratamiz.

```
class LoginForm extends CformModel {
    public $username;
    public $password;
    public $rememberMeqfalse;
}
```

Biz uchta o'zgaruvchi e'lon qilamizi, bular: \$username, \$password va \$rememberMe. Ular foydalanuvchi nomini, passwordini va shu bilan birgalikda username va passwordni eslab qolish belgisini (\$rememberMe). Bu belgi boshida **false** qiymatga ega bo'lmoqda.

Tekshiruv qoidalarini aniqlash:

Foydalanuvchi formadan ma'lumotlarni yuborganda, model ularni oladi, biz ularni amalga oshirayotganda ma'lumotlar to'g'ri ekanligiga ishonch hosil qilishimiz kerak. Bu ishni amalga oshirayotganda qoidalar to'plami bilan ma'lumotlarni tekshiramiz. Tekshirish qoidalarini amalga oshirshda bizga rules() metodi yordam beradi.

```
class LoginForm extends CformModel {
    public $username;
    public $password;
    public $rememberMeqfalse;
    private $_identity;
    public function rules()
    {
```



```
        return array(
            array( 'username, password', 'required'),
            array( 'rememberMe', 'boolean'),
            array( 'password', 'authenticate'),
        );
    }
}
```

```
public function authenticate($attribute,$params)
{
    $this->_identity=new UserIdentity($this->username,$this->password); if(!$this->_identity->authenticate())
        $this->addError( 'password', 'Неправильное имя пользователя или пароль. ');
}
}
```

## Tajriba ishi №18

### Mavzu: Tayyor veb ilovani hostingga joylashtirish va sozlash.

**Ishdan maqsad:** tayyor web sayt yoki ilova fayllarini hostingga joylashni va uni sozlashni o'rganish.

#### Nazariy qism

Saytni shaxsiy komputerimizda tayyorlab bo'lgach, uni internet tarmog'iga joylashimiz kerak. Agar internetga joylamasangiz, u holda saytingiz haqida hech kim xabardor bo'lolmaydi. Lekin, internetga joylash bilan ham ish tugamaydi, internetga kirganlar saytingizni qanday topa oladi? - degan savol tug'iladi. Shuning uchun internetga joylagan saytni odamlar topa olishi uchun imkon yaratish ham kerak, ammo bu haqda keyingiz darslarda so'z yuritamiz. Keling, avval, saytni internetga joylash haqida so'zlaymiz. Umuman olganda, saytlar internet tarmog'ida qanday joylanishi va saytlarga qanday qilib nom tanlanishi haqida o'rganamiz.

Har qanday saytni internetga joylash uchun domen va hosting olinadi

#### *Domen nima?*

Domen – bu sayt nomi, masalan [www.saytim.uz](http://www.saytim.uz). Siz qaysidir saytga kirayotganingizda sayt nomini ya'ni domenni yozasiz. Natijada shu domen sayt joylashgan hosting va baza bilan bog'lanadi.

Domen – bu cheksiz internet ummonidagi serverlardan birida joylashgan qaysidir saytga olib boradigan manzil hisoblanadi. Aslida Siz hosting xizmatidan foydalanib, saytingizni qaysidir serverga joylashtirganingizda, saytingiz manzili (server nuqta'i nazaridan) qandaydir IP manzilga teng bo'ladi. Masalan, 192.168.162.134 Bunday IP manzilni eslab qolish qiyin.

Agar domen bo'lmaganida saytingizga tashrif buyurish uchun foydalanuvchilar saytingiz IP manzilini eslab qolishlari va brauzerlariga 192.168.162.134 deb yozish orqali saytingizga kirishlari kerak bo'lar edi. Domen afzalligi shundaki, u tushunarsiz va eslab qolish qiyin bo'lgan IP manzillarni odamlarga tushunarli, eslab qolish oson bo'lgan chiroyli sayt nomlari aylantiradi. Shuning uchun ham domen sotib olganingizda eng avvalo bu domenni hostingingiz IP manziliga ulab qo'yasiz. Shunda domen nomi saytingiz IP manzilining tarjimai bo'lib qoladi. Har safar foydalanuvchilar o'z brauzerlarida domen nomingizni yozishsa, bu domen hostingingiz serverining IP manzilini chaqiradi va foydalanuvchiga saytingiz ko'rsatiladi!

Misol uchun, [qalam.uz](http://qalam.uz) sayti hostingining sayt joylashgan qismi uchun IP manzili 192.168.111.222 deb hisoblaylik. Bu saytga kirish uchun foydalanuvchilarning bu IP manzilni yozishlari ularga qiyinchilik tug'dirishi mumkin. Bu o'rinda foydalanuvchilar uchun eslab qolish oson va tushunarli bo'lgan domen tanlanadi. Sayt esa quyidagi tartibda ularga yetkaziladi:

#### **Qalam.uz -> 192.168.111.222 -> SAYT**

Domen ismlarda "domen hududi" degan tushuncha mavjud. Bu hududlar saytlarning u yoki bu turga ta'luqli ekanligini ko'rsatib turadi. Hozirgi kunda eng mashxur domen ism hududlariga **.com** (kommersiya-tijorat saytlari), **.net** (internet), **.info** (informatsiya-ma'lumot saytlari), **.edu** (education-ta'lim sohasiga oid saytlar), **.org** (organisation-tashkilot saytlari), **.gov** (government-hukumat saytlari) va boshqa davlat kodi yuqori bosqichli domenlar (country code top-level domains – **ccTLD**) kiradi.

ccTLD domenlari qaysidir davlat nomi qisqarmasiga asoslangan domen hududlariga bo'linadi. Masalan, O'zbekiston uchun **.uz** domenlari, Rossiya uchun **.ru** domenlari, Fransiya uchun **.fr** domenlari va hokazo. Har bir domen hududini o'ziga tegishli bo'lgan davlat boshqaradi. Misol uchun, **.uz** domenlariga O'zbekistondan boshqa hech qaysi davlat ro'yxatdan o'tkazish huquqiga ega bo'la olmaydi.

Domen ismlar domen registrarlari orqali amalga oshiriladi. Har bir domen nomi faqat bir marta ro'yxatga olinishi mumkin va uning nusxalari bo'lishi mumkin emas. Masalan, siz korzonangiz sayti uchun [esavdo.uz](http://esavdo.uz) deb nom tanladingiz, Bu domenni sotib olganingizdan keyin (kelishilgan muddat davomida) boshqa hech kim bu nomga egalik qilolmaydi.

U yoki bu domenning bandligi yoki ro'yxatga olinish uchun bo'shligini tekshirish domen registratorlarining whois so'rovi yordamida tekshirilishi mumkin. O'zbekiston domen registratorlari boshlig'i ccTLD.uz hisoblanadi. Bu saytga [www.cctld.uz](http://www.cctld.uz) yoki oddiygina [www.whois.uz](http://www.whois.uz) domeni orqali kirish mumkin. ccTLD.uz saytiga tashrif buyurib Siz O'zbekiston domen registratorlari haqida ma'lumot olishingiz va saytning chap tomon menyudan pastki qismida joylashgan kichik forma orqali qaysidir domenning bo'shligi yoki bandligini tekshirishingiz mumkin.

Domen nomi ro'yxatga olingandan so'ng Siz uning sozlanmalarini saytingiz joylashgan hosting IP manziliga (ular "nameserver" deb yuritiladi) mos ravishda tahrirlashingiz kerak bo'ladi. Shundan so'ng 24 soat ichida domeningiz ko'rsatilgan nameserverda joylashgan saytingizga olib borishni boshlaydi.

### **Hosting nima?**

Har qanday sayt qaysidir serverda joylashgan bo'ladi. Sayt va unga bog'liq fayllar (rasm, video, boshqa yuklab olinadigan fayllar) uchun ma'lum hajmda joy olish kerak. Masalan, 100 MB dan bir necha GB gacha bo'lishi ham mumkin. Sizni saytingiz uchun joy ajratib beruvchi xizmat turiga "hosting" deb ataladi. Odatda domen beruvchi kompaniyalarda ham hosting xizmati bo'ladi. Hosting va domenni bir kompaniyadan olish ham mumkin, yoki ularni boshqa-boshqa kompaniyalardan ham olish mumkin.

Saytlar uchun hosting beruvchi kompaniyalarda katta hajmdagi maxsus serverlar mavjud. Ular tinimsiz ishlab turish orqali saytlarning ishlashini ta'minlaydi. Hosting beruvchi kompaniyalar bir vaqtning o'zida yuzdan ortiq saytlarni o'zida saqlagan bo'lishi mumkin.

Siz internetga ulanib biror saytni ochish uchun kerakli manzilni terganingizda, dastlab, DNS (Domain Name Server) bilan bog'lanasiz. Keyin esa, DNS server tomonidan domen (sayt nomi) haqida ma'lumot tekshiriladi. Masalan [www.qalam.uz](http://www.qalam.uz) nomli manzilni ochganingizda, dastlab DNS server orqali qalam.uz nomi haqiqatan mavjudligi va qaysi hostingda joylashgani haqida ma'lumot olinadi, keyin ana shu hosting bilan bog'lanadi va saytga oid fayllar sizga ko'rsatiladi. Albatta, bular barchasi avtomatik tarzda bir necha soniyada amalga oshadi.

Quyidagi rasmlarda saytga kirish uchun sodir bo'ladigan jarayonlar tasvirlangan.

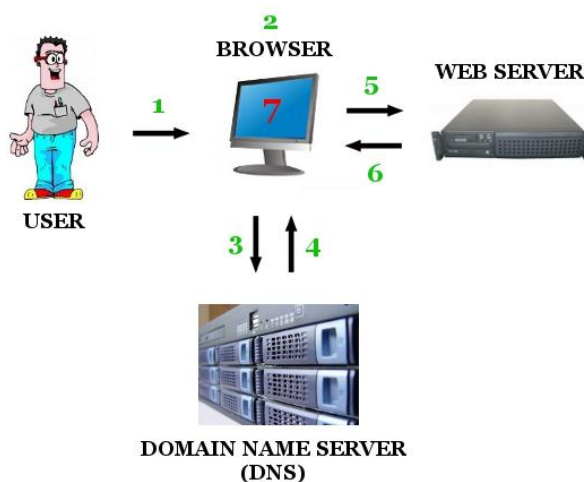
Bunda,

**user** – foydalanuvchi;

**browser** – saytni ochish dasturi (opera, mozilla, chrome);

**webserver** – sayt va uning fayllarini saqlovchi hosting.

**DNS** – domain name server (domenlar haqida ma'lumotni saqlovchi server)



Copyright:  
[freehostingtips.com](http://freehostingtips.com)

Yuqoridagi rasmdan ko'rinib turibdiki, internetdagi biror saytga kirish uchun sayt nomi yozilganda, browser (Opera, Mozilla, Chrome,...) dasturi dastlab DNS serverga murojaat qiladi va undan shu manzilning IP manzilini so'raydi. Boshqacha qilib aytganda, shu sayt qaysi hostingda

joylashganini so'raydi. DNS o'zining bazasini tekshiradi va so'ralgan sayt nomini izlaydi, Agar topsa uning IP manzilini brauzerga yuboradi (ya'ni internetga kirgan odamning komputeriga saytning qayerdaligini bildiradi) Keyin esa shu vaqtning o'zida brouzer berilgan manzilga (ya'ni hosting) bilan bog'lanib undan saytning kerakli sahifasini so'raydi. Shunday qilib hosing tomonidan sayt ko'rinishlari sizning brouzeringizga yetib keladi.

Hosting xizmati uchun mo'ljallangan serverlarning ba'zi ko'rinishlari quyida tasvirlangan (rasm)

*Rasm - Veb server xonalari*



Xulosa qilib aytganda, Hosting – bu sayt uchun ajratilgan joy bo'lib, saytdagi ma'lumotlarni (matn, rasm, fayllar va hokazo) o'zida saqlaydi. Uning hajmi bir necha MB (megabayt) yoki GB (gigabayt) bo'lishi mumkin. Domen yoki hosting olish uchun internet provayderiga murojaat qilish mumkin. Domen uchun yillik to'lov to'lanadi. Hosting uchun esa har bir provayderning tarif narxlariga ko'ra bir necha oylik yoki yillik to'lov to'lash mumkin.

O'zbekistonda domen yoki hosting beruvchi kompaniyalar (provayder yoki registrator) hozirda ko'plab shaharlarda faoliyat yuritmaqda. Masalan: BCC ([www.bcc.uz](http://www.bcc.uz)) , Arsenal-D ([www.webname.uz](http://www.webname.uz)) , Active ([www.active.uz](http://www.active.uz)) , Uzinfocom ([www.uzinfocom.uz](http://www.uzinfocom.uz)) va boshqalar.





**XO'JAYEV O.Q., ARTIKOV M.E.**

**«Web ilovalarni yaratish» fanidan tajriba ishlarini  
bajarish bo'yicha uslubiy ko'rsatma**

Qog'oz formati 60x90 1/8. Shartli bosma tabog' 2,1. Nashr tabog'i 2,0. Adadi 100 nusxa. TATU Urganch filiali ARMda tarqatma material sifatida ko'paytirildi. Urganch shahri, Al-Xorazmiy ko'chasi, 110-uy.

