
А. В. ДИКОВ

КЛИЕНТСКИЕ ТЕХНОЛОГИИ ВЕБ-ДИЗАЙНА HTML5 И CSS3



Учебное пособие



САНКТ-ПЕТЕРБУРГ
МОСКВА
КРАСНОДАР
2019

УДК 004.4(075.8)
ББК 32.973.26-018.1я73

Д 45 Диков А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 : учебное пособие / А. В. Диков. — Санкт-Петербург : Лань, 2019. — 188 с. : ил. — (Учебники для вузов. Специальная литература). — Текст : непосредственный.

ISBN 978-5-8114-3822-8

Учебное пособие в нескольких частях представляет собой полный учебный курс по технологиям веб-дизайна на стороне клиента, который включает в себя учебный материал со множеством примеров и сгруппированные по темам задания для практических работ. Первая половина учебного курса посвящена освоению базовых веб-технологий разметки и стилизации, без которых невозможно познавать интернет-программирование. Вторая половина курса содержит учебный материал по языку веб-программирования на стороне клиента JavaScript и мощной клиентской технологии разработки веб-приложений DOM. В курс не вошли серверные веб-технологии, уровень сложности которых и объем требуют дополнительных учебных часов в учебном плане образовательных учреждений.



УДК 004.4(075.8)
ББК 32.973.26-018.1я73

Рецензент

Д. В. ПАЩЕНКО — доктор технических наук, профессор, проректор по научной работе Пензенского государственного технологического университета, почетный работник сферы образования РФ.



Обложка
Е. А. ВЛАСОВА

© Издательство «Лань», 2019
© А. В. Диков, 2019
© Издательство «Лань»,
художественное оформление, 2019

ОГЛАВЛЕНИЕ

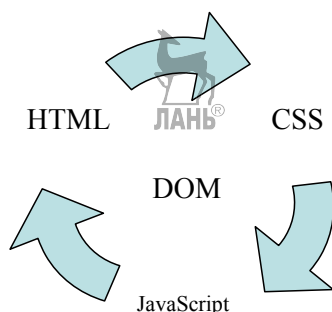
ПРЕДИСЛОВИЕ.....	5
ЧТО ТАКОЕ ИНТЕРНЕТ.....	6
ИСТОРИЯ ИНТЕРНЕТА	6
ПРОТОКОЛЫ ИНТЕРНЕТА.....	7
ВСЕМИРНАЯ ПАУТИНА	11
ЯЗЫК РАЗМЕТКИ ВЕБ-СТРАНИЦЫ HTML5	14
СОЗДАНИЕ И ПРОСМОТР HTML-ДОКУМЕНТОВ	15
СОХРАНЕНИЕ HTML-ДОКУМЕНТА	17
ПРОВЕРКА HTML-ДОКУМЕНТА.....	17
СТРУКТУРА HTML-ДОКУМЕНТА	19
СЕКЦИИ ОСНОВНОГО РАЗДЕЛА HTML-ДОКУМЕНТА.....	21
РАЗМЕТКА ТЕКСТА.....	22
ЛОГИЧЕСКОЕ ФОРМАТИРОВАНИЕ ТЕКСТА.....	27
СПИСКИ	28
СПИСКИ В СПИСКАХ (ВЛОЖЕННЫЕ СПИСКИ).....	30
ТАБЛИЦЫ	30
ПРЕДВАРИТЕЛЬНОЕ ФОРМАТИРОВАНИЕ	35
ГРАФИКА.....	35
ГИПЕРССЫЛКИ	55
ИЗОБРАЖЕНИЯ-КАРТЫ	61
МУЛЬТИМЕДИА	63
СТРУКТУРА САЙТА	76
ФОРМЫ.....	79
МЕТАТЕГИ.....	88
Практическая работа.....	90
Дополнительная литература	94
Приложение. Публикация сайта по ftp.....	94
СТИЛИЗАЦИЯ ВЕБ-СТРАНИЦ СРЕДСТВАМИ CSS.....	96
ВВЕДЕНИЕ.....	96
СЕЛЕКТОРЫ	96
СПОСОБЫ ПРИМЕНЕНИЯ СТИЛЕЙ	105
КОММЕНТАРИИ.....	106
НАСЛЕДОВАНИЕ.....	107
СЕМАНТИЧЕСКАЯ ВЕРСТКА.....	108
ЦВЕТОВОЙ КРУГ.....	111
МОДЕЛЬ КОНТЕЙНЕРОВ	115
ПОЗИЦИОНИРОВАНИЕ.....	118
ЕДИНИЦЫ ИЗМЕРЕНИЯ	131

ОФОРМЛЕНИЕ ГРАНИЦ	133
ФОНОВЫЕ ФАНТАЗИИ	138
ТРАНСФОРМАЦИЯ ЭЛЕМЕНТОВ	147
3D-ТРАНСФОРМАЦИЯ ЭЛЕМЕНТОВ	150
ОТ ПЛОСКОГО ТЕКСТА К 3D	152
СТИЛИЗАЦИЯ СПИСКОВ И ТАБЛИЦ	158
АНИМАЦИЯ В CSS	162
ИМИТАЦИЯ HTML-ЭЛЕМЕНТОВ, ИЛИ ПСЕВДОЭЛЕМЕНТЫ	168
ИГРА С ТЕНЬЮ	173
АДАПТИВНАЯ ВЕРСТКА	175
ЗНАЧОК ВЕБ-САЙТА	182
РЕДАКТОР СТИЛЕЙ В БРАУЗЕРЕ FIREFOX	182
Практическая работа	183
Список литературы	184



ПРЕДИСЛОВИЕ

Учебное пособие включает в себя все основные и самые популярные технологии разработки веб-страниц на стороне клиента (компьютера пользователя Интернетом). К ним относятся в порядке следования освоения: HTML, CSS, JavaScript и DOM. Язык гипертекстовой разметки HTML позволяет формировать содержание веб-страницы (контент) как последовательность различных элементов: заголовки, абзацы, списки, таблицы, гиперссылки и так далее. Всем элементам по умолчанию присвоен определенный стиль (формат) отображения: размер, цвет, расположение и так далее. Для изменения стилевого оформления веб-дизайнерами используется технология каскадных таблиц стилей CSS. Каскадность означает возможность многократного наложения стилей на одни и те же элементы.



Технология DOM, завершающая этот ряд, является объединяющей, так как обеспечивает через JavaScript доступ ко всем элементам html-документа с возможностью динамического изменения этих элементов вплоть до удаления и создания новых, а также позволяет изменять стилевые свойства объектов, за которые отвечает технология CSS. Часто все эти технологии в объединенном варианте называют «Динамический HTML».



ЧТО ТАКОЕ ИНТЕРНЕТ

Интернет – это уникальное и важное создание в мировом сообществе. Он возник неуправляемо. Никто специально не руководил процессом развития сети. Интернет не имеет собственника ни в виде акционерного общества, ни в виде персоны. Существуют большое множество мелких или средних организаций (например провайдеры или университеты), обеспечивающих бесперебойную работу нескольких сетевых серверов, на которых находится доступная для пользователей Интернета информация. Можно условно сказать, что каждый из них владеет маленькой частью Интернета. Появились люди, которых стали называть интернетоманами за то, что они слишком увлечены сетью.

Интернет – это:

- *глобальная мировая компьютерная сеть*
охватывает весь земной шар
- *сеть сетей*
включает в себя более мелкие компьютерные сети
- *информационная магистраль XXI века*
для передачи данных применяются самые последние достижения науки и техники
- *огромное хранилище информации*
на сегодняшний день во Всемирной паутине хранится более 10 миллиардов веб-страниц и 2 миллиона добавляются каждый день
- *мир без границ*
нет необходимости оформлять визу, платить налоги, проходить таможенную для просмотра информации, находящейся на другом конце света, и для обмена ею
- *новый мир*
здесь возникли совершенно новые формы взаимодействия, новые возможности, новые ощущения
- *расширение пространства межличностного общения*
- *быстрый и дешевый способ сказать всему миру что-то важное*
- *свободный мир*
- *возможность совместной разработки какого-либо интересного проекта*

На момент написания этого текста число пользователей Интернета во всем мире составило примерно три миллиарда человек из 7,2 миллиардов населения нашей планеты.

ИСТОРИЯ ИНТЕРНЕТА

5 декабря **1969** года Министерство обороны США организовало компьютерную сеть ARPANet (Advanced Research Project Agency Net – сеть Агентства перспективных исследовательских проектов). Сеть соединяла 4 компьютера, находившихся в следующих учреждениях: Калифорнийском университете (Лос-Анджелес), Стэнфордском НИИ, Университете города Санта-Барбара и Университете штата Юта. Через год сеть включала уже в 4 раза больше узлов (компьютеров).

В 1972 году Рэй Томлинсон изобрёл электронную почту и отправил первое послание по этой сети. Вскоре сеть разрослась и в 1973 году стала международной, благодаря подключению узлов Норвегии и Англии. Всё большее число университетов и других организаций хотели подключиться к сети. Управлять ресурсами становилось всё труднее и труднее. Поэтому в 1983 году от сети ARPANet отделилась сеть MILNet, которая связывала военные объекты. Сети между собой обменивались по протоколу IP (Internet Protocol). Именно в этот момент появилось само слово Internet (1982 г.). В русском языке слово Интернет пишется с большой буквы и склоняется.

В 1984 году ARPANET перешла под покровительство NSF (Национальное научное общество США), а в 1986 году NSF образовало собственную сеть NSFNet. Эта сеть включала пять центров, оснащённых суперкомпьютерами со скоростью передачи 56 Кбит/с, и множество региональных сетей.

В 1990 году ARPANet сменила название на Internet. Появился первый коммерческий провайдер (ISP). В этом же году был зарегистрирован домен бывшего СССР (**su**).

Официальная дата рождения российского Интернета (Рунета) – 7 апреля 1994 года, когда Российский институт развития общественных связей получил в свое управление домен **ru**.

ПРОТОКОЛЫ ИНТЕРНЕТА

В основе сетевых технологий лежит идеальная модель OSI (*Open System Interconnection* – взаимодействие открытых систем). Это многоуровневая модель, включающая в себя 7 уровней сетевых взаимодействий: физический, канальный, сетевой, транспортный, сеансовый, представительский и прикладной.

Каждый уровень выполняет свою специфическую функцию. При сетевом обмене сообщаются соответствующие уровни двух компьютеров посредством запросов на обслуживание у нижележащего уровня. Реализация всех уровней, кроме первого, может отличаться в зависимости от платформы ПК.

Компьютерные сети могут отличаться своей архитектурой и, соответственно, набором протоколов взаимодействия, но все они соответствуют в той или иной степени идеальной модели OSI.

Архитектура глобальной сети Интернет основана на стеке протоколов **TCP/IP**. Это фактически два отдельных протокола разных уровней.

TCP (*Transmission Control Protocol* – протокол управления передачей данных) отвечает за то, как информация передается по сети. Данные распределяются на небольшие пакеты, каждый из которых маркируется и передается своим путем. Это протокол транспортного уровня.

IP (*Internet Protocol* – протокол Интернета) отвечает за то, куда передавать информацию. Это протокол сетевого уровня. Каждый компьютер в глобальной мировой сети должен иметь свой ip-адрес, который выражается четырьмя байтовыми числами, то есть общая длина адреса составляет 32 бита (4×8 бит), например 80.95.32.19. Ip-адрес может быть постоянным, если компьютер постоянно подключен к сети, и динамическим, если компьютер подключается к сети только на время сеанса. Этот динамический адрес

действует только во время сеанса. В каждом новом сеансе адрес может отличаться от предыдущего.

Нетрудно подсчитать, что при такой системе адресации всего может быть сгенерировано около 4 млрд адресов (256^4). Из-за особенностей адресации к некоторым типам локальных сетей число адресов уменьшается до 2 млрд. Однако Интернет переживает период бурного роста, и существующих ip-адресов оказывается недостаточно, что влечет переход к разработке новых версий ip-протокола.

Новый протокол **IPv6** (версия 6) переходит на увеличенную длину адреса с 32 бит (IPv4) до 128 бит. Адресное пространство при этом расширяется до 340282366920938463463374607431768211456 уникальных адресов, то есть это более чем 340 триллионов триллионов триллионов. Адреса отображаются как 8 групп по 4 16-ричные цифры, разделенные двоеточием.

Пр и м е р.

2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d

Каждый компьютер, через который проходит tcp-пакет, может по ip-адресу определить, на какой компьютер необходимо далее передать этот пакет так, чтобы он оказался ближе к получателю. При этом в расчет принимаются условия связи и пропускная способность линии, а не географическое расположение принимающего компьютера.

Решением вопросов выбора следующего пункта назначения занимается специальное средство – *маршрутизатор*, который может быть либо отдельным компьютером, либо специальной программой, работающей на сервере сети.

Пользователю сети неудобно обращаться к компьютеру по ip-адресу. Гораздо легче запомнить словесный ассоциированный адрес. Поэтому существуют и доменные (*domain* – область, территория) имена компьютеров, помещаемые поверх числовых адресов и совершенно от них не зависящие. Например, доменный адрес провайдера «Волгателеком» www.vt.ru, а адрес электронной почты может выглядеть как ivanov@yandex.ru. В Сети находится специальный сервер **DNS** (*Domain Name System* – сервер имен доменов), который связывает числовой и доменный адрес.

Все узлы Интернета классифицированы Международным центром сетевой информации **ICANN** (*Internet Corporation for Assigned Names and Numbers*, читается как *АЙКЭН*) [<http://www.icann.org/>] на некоторое число доменов первого (верхнего) уровня, которые можно разделить на общенациональные и национальные:

Общенациональные домены

com (от commercial)	Появился в 1985 году. Предназначался для коммерческих организаций, но в настоящее время в этом домене регистрация осуществляется без ограничений. В начале 2009 года домен включал около 80 млн имен второго уровня
----------------------------	---

net (от network)	Появился в 1985 году. Первоначально предназначался для организаций, развивающих сетевые технологии. В настоящее время в этом домене регистрация осуществляется без ограничений. В начале 2009 года домен включал около 12 млн имен второго уровня
org (от organizations)	Появился вместе с доменами <i>com</i> и <i>net</i> . Ассоциируется прежде всего с некоммерческими организациями, несмотря на то, что формальных ограничений для регистрации не существует
info (от information)	Появился в 2001 году. Адресован организациям и частным лицам, желающим разместить в Интернете информацию о себе, своем предприятии, товаре или услуге. <i>Information</i> – термин, легко переводимый на различные национальные языки мира, поэтому в этом домене зарегистрированы примерно в равных долях имена второго уровня из различных стран мира
biz (от business)	Появился в 2001 году. Предназначен для коммерческих организаций, предприятий и корпораций. Дополняет ставший универсальным домен <i>com</i>

Специальные домены общего использования

coop (от cooperatives – кооперативы)	Появился в 2001 году. Предназначен для кооперативов (совместных предприятий) и организаций, непосредственно с ними связанных
aero	Появился в 2000 году. Предназначен к использованию в индустрии авиаперевозок и путешествий
museum (от museums – музеи)	Появился в 2000 году. Предназначен для музеев и организаций, имеющих отношение к музейному делу, а также частных лиц, профессионально связанных с этой сферой деятельности
pro (от professionals – профессионалы)	Появился в 2001 году. Предназначен для профессионалов и профессиональных сообществ
name	Появился в 2002 году. Предназначен для частных лиц, желающих зарегистрировать во Всемирной паутине свое имя и фамилию в качестве доменного имени

Домены ограниченного использования

edu (от educational)	Образовался в 1984 году. Предназначен для образовательных учреждений и проектов США. Регистрация доменов второго уровня в этом домене осуществляется под контролем американского правительства. Домен содержит около 5000 имен второго уровня
-----------------------------	---

mil (от military)	Появился в 1984 году. Предназначен только для военных организаций и учреждений, отвечающих за безопасность страны. Регистрация доменов второго уровня в этом домене осуществляется под контролем американского правительства
gov (от government – правительство)	Появился в 1984 году. Используется исключительно для федеральных государственных учреждений США
int (от international organizations – международные организации)	Появился в 1984 году. Предназначен исключительно для международных организаций

Национальные домены

ru (от Russia)	Появился в 1994 году. Предназначен для регистрации ресурсов, рассчитанных на российскую аудиторию. В 2009 году число зарегистрированных имен второго уровня превысило 2 млн
рф (от Российская Федерация)	Появился в 2010 году. Является российским национальным доменом. Благодаря внедрению зоны .рф, в Интернете появились адреса сайтов на русском языке. Введение в употребление таких доменных имен делает Интернет понятнее для всех русскоговорящих жителей планеты и, как следствие, возможности Интернета становятся доступными для тех, кому раньше было сложно воспринимать адреса, написанные латиницей
su (от Soviet Union – Советский Союз)	Появился в 1990 году. Домен бывшего СССР. В настоящее время объединяет ресурсы, налаживающие культурные, научные и деловые связи между странами СНГ
tv (от Tuvalu – Тувалу)	Национальный домен тихоокеанского островного государства Тувалу. Однако также преподносится как домен для сайтов телевизионных компаний и ресурсов, связанных с телевидением, из-за повсеместно принятого сокращения <i>tv</i> для слова <i>television</i> – телевидение

Подробную информацию о доменах верхнего уровня см. по адресу <http://info.nic.ru/> и <http://your-hosting.ru/>.

Домены второго уровня даются специально уполномоченной организацией предприятиям и частным лицам в аренду, как правило, с ежегодной оплатой. В России такой организацией являлась **РосНИИРОС** (Российский НИИ развития общественных сетей). С 2000 года распределением адресного пространства в Рунете стал заниматься Региональный сетевой информаци-

онный центр (RU-CENTER) [<http://www.nic.ru/>]. Он является аккредитованным регистратором доменных имен в зонах RU, SU, NET, COM, ORG, BIZ, INFO и др. На сайте этой организации работает служба Whois Service, где можно бесплатно получить информацию о любом зарегистрированном домене. В рубрике «Национальные домены» хранятся домены всех стран мира.

В качестве доменов второго уровня выбираются названия предприятий, журналов и газет, видов деятельности, фамилии известных людей, географические названия, торговые марки и тому подобное. Например: 1september.ru, putin.ru, generalmotors.ru. Из примеров видно, что домены разных уровней отделяются друг от друга точкой.

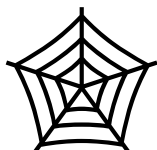
Обладатель домена второго уровня может создавать множество доменов третьего и далее уровней. Так, на общедоступном сервере 0fees.net можно бесплатно зарегистрировать имя для своей личной веб-странички, например petrov.0fees.net

Не только каждый компьютер, но и каждый ресурс (файлы, веб-сайты и т. д.) в Сети обладает уникальным адресом, полный формат которого называют **URL** (*Uniform Resource Locator* – унифицированный указатель ресурсов). URL условно можно разделить на 3 части. Первая часть содержит название протокола доступа к ресурсу (см. следующий раздел), вторая часть – доменное имя компьютера, третья часть представляет собой путь к файлу на данном компьютере. Можно сказать, что URL – это сетевое расширение стандартного понятия «путь к файлу».

ВСЕМИРНАЯ ПАУТИНА

Относительно молодая (1993 года рождения) служба **WWW** (World Wide Web – Всемирная паутина) становится сегодня все более популярной благодаря мультимедийному интерфейсу. В 2004 году изобретатель Веба Тим Бернерс-Ли стал лауреатом премии Millennium Technology Prize. Эта награда считается своеобразным аналогом Нобелевской премии в области высоких технологий. Приз был учрежден фондом Finnish Technology Award Foundation, базирующимся в Финляндии, а его размер составляет 1 млн евро. Тим Бернерс-Ли

удостоился чести стать первым лауреатом за изобретение языка гипертекстовой разметки HTML и основанной на нем системы World Wide Web.



Всемирная паутина представляет собой совокупность веб-страниц (узлы паутины), разбросанных по всему миру на веб-серверах и соединенных гиперсвязями (нити паутины). *Веб-страница* представляет собой документ, в оформление которого можно включать всевозможные виды представления информации: текст, графику, анимацию, видео и аудио, гиперссылки. Основным форматом представления веб-страниц является HTML (*HyperTextMarkupLanguage*). Тим Бернерс-Ли на популярности своего изобретения не заработал ни цента (не считая премии, конечно). Не требовать каких-либо отчислений за свои изобретения было его принципиальной позицией. Он отмечал, что если бы пытался требовать денег за использование своих разработок, то единой Всемирной паутины могло бы и не

возникнуть. Вместо нее работало бы множество мелких независимых гипертекстовых служб, и обмен информацией не был бы так прост.

Можно также в качестве узла паутины рассматривать веб-сервер, хранящий большое число веб-страниц. Совокупность объединенных гиперссылками веб-страниц, посвященных определенной теме и размещенных на одном сервере, называют *веб-сайтом*. Как правило, сайт содержит ссылки и на другие веб-сайты сети Интернет, схожие по тематике. Поэтому, попав на один сайт, посетитель может одним щелчком мышки на гиперссылке переместиться с текущего веб-сервера на другой, расположенный, быть может, совсем в другой стране. Вполне вероятно, он этого даже и не заметит.

Доступ к веб-документам осуществляется по протоколу HTTP (HyperText Transfer Protocol – протокол передачи гипертекста). Каждая веб-страница и веб-сайт имеют свой адрес в сети, который начинается с протокола связи, затем следует доменное имя компьютера, подключенного и зарегистрированного в сети, и заканчивается именем файла на этом компьютере.

<http://www.comptek.ru/alta.html>

<http://www.sura.ru/>

Если имя файла отсутствует (как во втором примере), то загружается начальная страница сервера или сайта. Главная страница сайта имеет имя, как правило, `index.html` или `index.htm`, хотя существует несколько менее известных зарезервированных имен для главной страницы сайта, которые знает и ищет программа-просмотрщик веб-страниц.

Программа просмотра веб-документов называется **браузером** (browser), или обозревателем. Наиболее популярными являются обозреватели: Firefox от Mozilla, Internet Explorer от Microsoft, Opera и др.

Через веб-интерфейс можно организовать доступ к электронной почте, хранящейся на почтовом сервере. Такие серверы являются общедоступными. Это значит, что каждый пользователь Интернета может с любого компьютера, расположенного в любом городе и стране, просмотреть свою почту и отослать электронное письмо посредством веб-страницы. Хотя при определенном умении можно через такие серверы просматривать почту и с сервера провайдера.

Наиболее популярными почтовыми серверами с веб-интерфейсом в нашей стране являются *mail.ru*, *mail.yandex.ru*, *mail.rambler.ru* и *mail.google.com*. Последний пользуется популярностью во всем мире.

Вопросы и упражнения

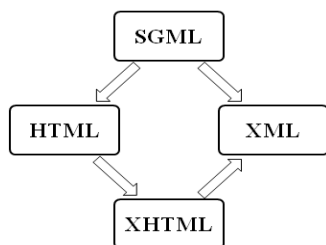
1. Что такое Интернет?
2. Кто является собственником Интернета?
3. Из какой сети возник Интернет?
4. Какой протокол связи в Интернете является главным?
5. Как устроена доменная система имен?
6. Сколько может быть сгенерировано ip-адресов, если они выражаются четырьмя байтовыми числами?

-
7. Кто такой провайдер?
 8. Почему одну из служб Интернета называли Всемирной паутиной?
 9. Какой документ называют веб-страницей?
 10. Что такое сайт?
 11. Что такое гиперссылка?
 12. Что такое браузер?
 13. По какому протоколу работает Всемирная паутина?
 14. Какая организация распределяет доменные имена верхнего уровня?
 15. Назовите национальный домен тихоокеанского островного государства Тувалу.
 16. Какое имя имеет главная страница любого веб-сайта?
 17. Кто создал Всемирную паутину?
 18. Какого числа отмечается день рождения Рунета?



ЯЗЫК РАЗМЕТКИ ВЕБ-СТРАНИЦЫ HTML5

История языка HTML начинается с GML, стандарт которого был принят в 80-х годах прошлого века с названием SGML (Standard Generalized Markup Language – стандартный обобщенный язык разметки). Предложенный в 1974 году Чарльзом Голдфарбом, SGML представляет собой метаязык — систему для описания других языков. При всех своих возможностях он слишком сложен для большинства браузеров. Только спецификация SGML занимает свыше 500 страниц. Поэтому на основе SGML был создан HTML — специализированный, гораздо более простой язык. Язык XHTML был разработан как переходный от HTML к более сложному XML.



HTML есть аббревиатура от слов HyperTextMarkupLanguage – язык разметки страниц. Основное назначение этого языка состоит в размещении на странице таких элементов, как заголовки, абзацы, списки, таблицы, линии, рисунки, гиперссылки. Для каждого такого элемента в HTML предназначен соответствующий тег – команда языка, или группа тегов. Поэтому HTML еще называют языком тегов.

Язык HTML был разработан в 1991–1992 годах Тимом Бернерсом-Ли во время его работы в CERN и поддерживался браузером Mosaic. В 1993 году появилась его версия **1.2**. В ноябре 1995 года организацией Internet Engineering Task Force (IETF) был принят стандарт HTML **2.0**. В начале 1997 года международной организацией The World Wide Web Consortium (**W3C**) [<http://www.w3.org/>] был принят стандарт **3.2** этого языка (версия **3.0** была совместима со второй версией, но ее реализация была сложна для браузеров того времени; версия **3.1** официально никогда не предлагалась), а в конце 1997 года был принят уже новый стандарт **4.0**. В конце 1999 года была утверждена спецификация **4.01**, которая и используется в настоящее время. Основное новшество последнего стандарта состоит в появлении каскадных таблиц стилей (CSS – Cascading Style Sheets). Официальной спецификации HTML **1.0** не было принято. До 1995 года существовало множество неофициальных стандартов HTML. Чтобы стандартная версия отличалась от них, ей присвоили сразу второй номер. Начиная с 2004 года, сообществом **WHATWG** (Web Hypertext Application Technology Working Group – рабочая группа по разработке гипертекстовых приложений для Веба) велась разработка пятой версии HTML. Со временем к ней присоединилась организация W3C. С 2011 года HTML5 утверждается в W3C как рабочий проект (Working Draft). Спецификация получила статус «W3C рекомендована» 28 октября 2014 года.

Теги языка HTML представляют собой служебные слова языка, заключенные в треугольные скобки. Например <BODY>. Многие теги используются парами. Тогда различают открывающий и закрывающий тег. Например, <BODY> </BODY>.

Закрывающий тег начинается с косой черты, за которой идет название открывающего тега. Внутри пары обычно располагается информация и другие теги. Поэтому парные теги называют *контейнерами*.



Контейнеры могут содержать в себе другие контейнеры только целиком по принципу матрешки



<BODY>

<H1> Добро пожаловать ко мне на домашнюю страницу </H1>

<P> Здесь вы узнаете о моих увлечениях и открытиях, о моих друзьях и родных, а также найдете ссылки на интересные места в Интернете </P>

</BODY>

Для изменения некоторых, установленных по умолчанию, значений стилевых элементов предназначены *атрибуты* (параметры) к тегам. Атрибут и его значение пишется внутри треугольных скобок через пробел от основного служебного слова.

<BODY BGCOLOR="green" TEXT="red">

В строгой (strict) версии HTML 4.01 и тем более HTML5 почти все атрибуты, кроме универсальных, таких как ID, CLASS и STYLE, считаются устаревшими. Вместо устаревших атрибутов используется каскадная таблица стилей. Например, те же параметры тега BODY можно заменить на свойства таблиц стилей.

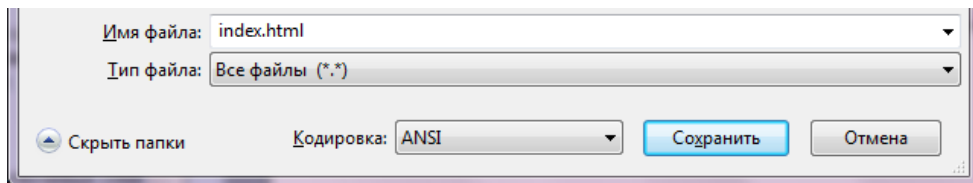
<BODY STYLE="background-color: green; color: red;">

Тем не менее, в книге рассматриваются устаревшие атрибуты с целью достижения понимания сущности описываемой технологии. Устаревшие атрибуты до сих пор поддерживаются большинством популярных браузеров. Создавать новые веб-страницы рекомендуется без устаревших атрибутов и тегов.


СОЗДАНИЕ И ПРОСМОТР HTML-ДОКУМЕНТОВ

Создаются html-документы в любом текстовом редакторе (например, Блокноте) или специализированном html-редакторе, а также в визуальном редакторе веб-сайтов.

Все html-редакторы сохраняют созданный документ с расширением html или htm. Блокнот же по умолчанию дает документу расширение txt, поэтому при сохранении документа надо самому дать расширение html или htm.



Так как html-документ может представлять собой целый набор файлов, то лучше всего поместить их все в отдельную папку. А файл-запуск назвать стандартным именем **index.html**, известным любому браузеру.

После того как документ сохранен, его можно просмотреть в браузере через меню «Вид → Источник». Браузер «понимает» и исполняет html-код. Затем можно снова вернуться к Блокноту с html-кодом, внести изменения, сохранить их в файле. На панели инструментов браузера есть кнопка «ОБНОВИТЬ» , с помощью которой легко увидеть изменения (предварительно сохраненные), внесенные в создаваемый или редактируемый html-документ.

Если в HTML коде содержатся ошибки, то браузер просто проигнорирует их и не выдаст вам или посетителю страницы сообщений об ошибках. Это, конечно, делает процедуру отладки более трудоемкой.

Любой браузер «понимает» язык тегов. Самым популярным является браузер Internet Explorer, так как он бесплатно поставляется с системой Windows и на сегодняшний день является одним из самых мощных браузеров. Еще одним бесплатным, мощным и популярным браузером является Firefox от компании Mozilla. Эти два браузера фактически диктуют стандарты в области разработки веб-страниц.

Веб-страница – это еще одно название html-документа. Оно дается тем документам, которые размещаются в сети Интернет. Набор веб-страниц, объединенных общей темой и гиперсвязями и расположенных на одном сервере, называют **сайтом**. Как правило, сайт содержит гиперссылки на другие сайты. Огромное множество взаимосвязанных веб-сайтов и составляют Всемирную паутину сети Интернет. На сегодняшний день в Сети насчитывается более 10 млрд веб-страниц и каждый день добавляется несколько миллионов.

Еще одно применение html-документы нашли в сфере разработки электронных мультимедийных учебников и обучающих программ.

Среди визуальных редакторов самыми известными являются два: Microsoft Office SharePoint Designer и Adobe Dreamweaver. Первый рассчитан на неопытных создателей веб-сайтов, второй же отличается большей корректностью автоматического генерирования html-кода и более широким набором инструментов для написания кода «вручную».

Вопросы и упражнения

1. С помощью браузера откройте любую веб-страницу из Интернета и просмотрите ее html-код.
2. Найдите ошибку в приведенном фрагменте html-кода.

<BODY>

<H1> Добро пожаловать ко мне на <P> домашнюю страницу </H1>

Здесь вы узнаете о моих увлечениях и открытиях, о моих друзьях и родных, а также найдете ссылки на интересные места в Интернете
</P>

</BODY>

3. Может ли закрывающий тег содержать параметры?
4. Кто создал Всемирную паутину?
5. Что такое тег? Какие бывают теги? Как метод матрешки применим к тегам?
6. Что используется с тегами вместо устаревших атрибутов?

СОХРАНЕНИЕ HTML-ДОКУМЕНТА

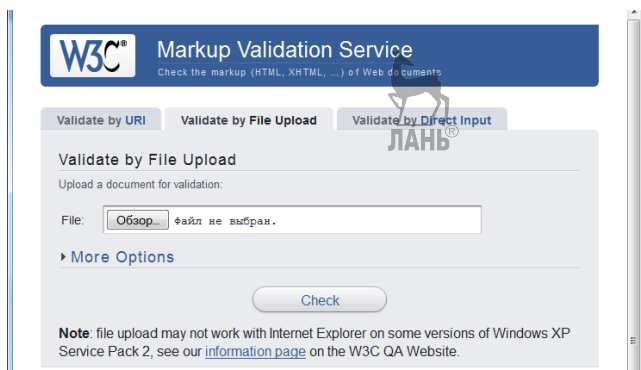
Обычно при работе с теми или иными приложениями создаваемый документ сохраняется в одном файле. Такими документами являются документы текстовых редакторов, табличных процессоров, графических и музыкальных редакторов и т. д. В отличие от них, веб-страница и тем более веб-сайт хранятся в нескольких файлах. К ним относятся: html-код (для каждой страницы узла), рисунки (каждый рисунок хранится в виде отдельного файла), звуковые и видео-файлы, скрипты и т. д. Для простоты написания кода и размещения сайта в Интернете, а также для удобства переноса всех компонент узла с одного компьютера на другой рекомендуется хранить эти компоненты в одной общей папке, которая может включать в себя и другие подпапки, предназначенные под определенные совокупности файлов, объединенных какой-то темой.

ПРОВЕРКА HTML-ДОКУМЕНТА

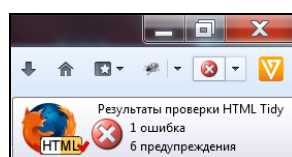
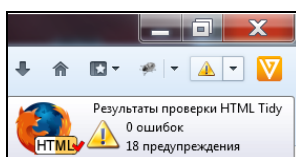
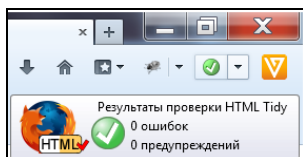
Создаваемая веб-страница должна быть синтаксически корректной и валидной, то есть соответствующей стандартам Всемирной паутины. Синтаксические правила определяются DTD, то есть типом документа. Браузеры при отображении страницы не показывают ошибок, даже если они присутствуют. Они просто их игнорируют. Это хорошо для посетителя веб-страницы, но затрудняет корректировку документа разработчику, ведь ошибки не позволяют достичь желаемого результата.

Для проверки веб-страниц на наличие ошибок и неточностей существует множество путей и способов. Условно они делятся на онлайн-овые и локальные. Онлайн-овые предназначены для проверки страниц с помощью браузера через Интернет, а локальные используются для проверки документов на текущем компьютере. По адресу <http://validator.w3.org> располагается самый распространенный инструмент для проверки отдельных страниц на валидность. Этот сайт предлагает три способа проверки: по URL (адресу), локальному файлу и введенному в форму коду.

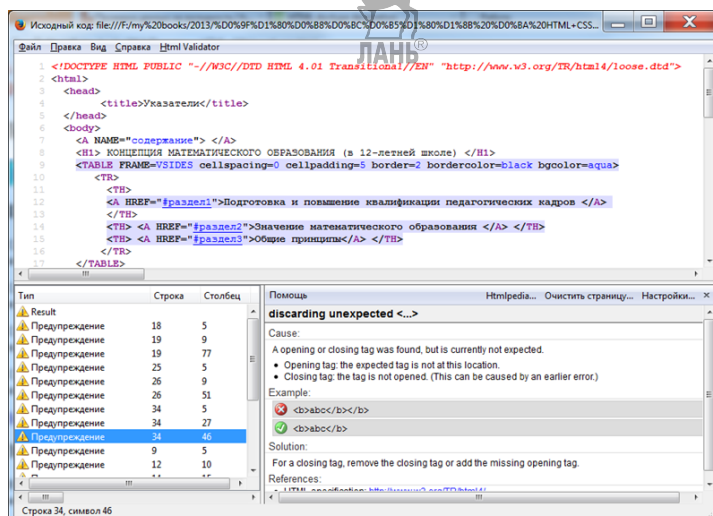
После проверки система выдает число ошибок (errors) и предупреждений (warnings) и список этих ошибок в коде с подробным описанием. Достаточно хоть немного знать английский язык, чтобы понять представленное.



Среди локальных инструментов в помощь разработчику веб-страниц существует программа **HTML Tidy**, написанная Дейвом Рэггеттом из Консорциума Всемирной паутины (W3C). Для браузера Mozilla Firefox существует бесплатное расширение от Марка Гуери **HTML Validator**, которое можно скачать по адресу <http://users.skynet.be/mgueury/mozilla/>. Валидатор использует для проверки документа алгоритм Tidy. После установки в верхней строке браузера появится кнопка с выпадающим меню команд для проверки и отображения результатов. После загрузки браузером страницы валидатор сразу ее проверит, и по изображению кнопки можно понять, содержит она ошибки и предупреждения или нет.



Если имеются ошибки и предупреждения, их можно просмотреть в отдельном окне, просто нажав на кнопку валидатора, или вызвать его через выпадающее меню.



В левой нижней области показывается номер строки и столбца, где содержится ошибка, в нижней правой области – комментарий к этой ошибке и пример ее исправления.

СТРУКТУРА HTML-ДОКУМЕНТА

Любой html-документ начинается с тега DOCTYPE, который является сокращением от слов document type и переводится как «тип документа». Так как браузер способен отображать документы различных типов, то для более корректного отображения важно явно указать ему используемую при разработке документа версию языка. В противном случае браузер будет сам пытаться определить формат документа. Для 5-й версии HTML используется простое и короткое объявление `<!DOCTYPE html>`.

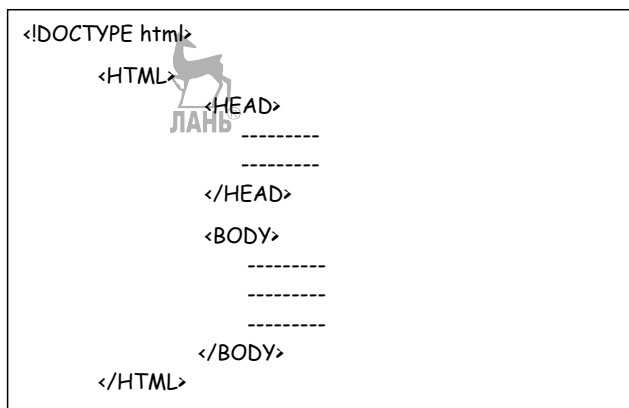


Используйте на каждой html-странице полностью сформированное объявление DOCTYPE



Неправильное объявление типа документа приведет к тому, что страница выведется браузером в нестандартном режиме и будет вести себя непредсказуемо!

Весь html-документ логически делится на две части: заголовочную и основную. Заголовочная часть заключается в теги `<HEAD>` `</HEAD>`. Основная – в контейнер `<BODY>` `</BODY>`. Весь документ – в контейнер `<HTML>` `</HTML>`.



У тега `<BODY>` есть несколько параметров, влияющих на оформление внешнего вида всего документа.

Устаревший атрибут тега <code><BODY></code>	Значение атрибута	Аналогичное свойство из таблицы стилей	Комментарии
TEXT	название или номер цвета	color	определяет цвет текста на всей странице
LEFTMARGIN	число	margin	поля слева и справа
TOPMARGIN	число	margin	поля сверху и снизу

Устаревший атрибут тега <BODY>	Значение атрибута	Аналогичное свойство из таблицы стилей	Комментарии
BGCOLOR	название или номер цвета	background-color	цвет фона
BACKGROUND	имя файла-картинки	background-image	фоновый рисунок
BGPROPERTIES	FIXED	background-attachment	фон не перемещается

О том, как задавать цвет фона или текста и фоновый рисунок, будет сказано дальше.

Существует ряд тегов, которые могут располагаться только в заголовочной части. Одним из них является тег-контейнер <TITLE>, в который заключается предложение, помещаемое браузером в заголовок html-документа или в папку «Избранное» по желанию посетителя.

<HEAD> <TITLE>Технологии будущего</TITLE> </HEAD>

Все остальные рассматриваемые здесь теги будут размещаться в основной части.

В отличие от тега TITLE, который дает информацию обо всем документе и может появиться в документе только один раз, атрибут **title** может содержаться в любом количестве элементов. Этот атрибут предоставляет информацию об элементе, в котором он содержится. Браузеры отображают title как «подсказку» (краткое сообщение), появляющуюся при наведении указательного устройства на объект. Это, например, удобно использовать при наличии аббревиатур в тексте.

<ABBR title="Карманный_Персональный_Компьютер"> КПК </ABBR>

Вопросы и упражнения

1. С помощью браузера откройте любую веб-страницу из Интернета и перейдите в режим просмотра ее html-кода. Узнайте, какую версию html использовали разработчики для написания данной страницы.
2. Создайте шаблон html-документа в текстовом редакторе, содержащий только структурные теги и <DOCTYPE> с параметрами. Сохраните html-документ в папку под именем «шаблон.html».
3. Создайте новый html-документ на основе шаблона. Добавьте в тег <BODY> стиль, меняющий цвет фона веб-страницы на любой, отличный от белого (название цвета см. на форзаце). Сохраните файл под именем index.html и откройте его в браузере. Добавьте тег <TITLE> с текстовой информацией, сохраните измененный документ и обновите его в браузере. Проверьте, появилась ли в заголовке браузера введенная текстовая информация.
4. Сколько голов (тег <HEAD>) и тел (тег <BODY>) может содержать html-документ?

СЕКЦИИ ОСНОВНОГО РАЗДЕЛА HTML-ДОКУМЕНТА

Секционные элементы появились только в HTML5. Эти элементы, так же как и структурные теги `<BODY>` и `<HEAD>`, не размещают никакого контента на веб-странице, но позволяют разработчику разбить страницу на логические разделы. Кроме этого, секционные элементы позволяют лучше проанализировать контент страницы и всего сайта программам, сканирующим Всемирную паутину для составления предметного указателя Интернета (от Yandex, Google и т. д.).

Если рассматривать структуру содержимого веб-страницы сверху вниз, то обычно выделяют три основных блока: заголовочная часть (`<HEADER>`), основная (`<MAIN>`) и заключительная часть (`<FOOTER>`). Все секционные элементы являются контейнерами.

```
<!DOCTYPE html>
  <HTML>
    <HEAD>
      -----
      -----
    </HEAD>
    <BODY>
      <HEADER>
        -----
        -----
      </HEADER>
      <MAIN>
        -----
        -----
      </MAIN>
      <FOOTER>
        -----
        -----
      </FOOTER>
    </BODY>
  </HTML>
```



В заголовочном разделе `<HEADER>` обычно размещается заголовочный баннер с названием сайта. Под ним часто горизонтально располагается навигационная панель, то есть оглавление сайта, которая в html-коде заключается в секционный элемент `<NAV>`.

```
<HEADER> <H1>ШКОЛА XXI века</H1> </HEADER>
<NAV>
  <UL>
    <LI><A HREF="#chapterOne">Глава I</A>
    <LI><A HREF="#chapterTwo">Глава II</A>
    <LI><A HREF="#chapterThree">Глава III</A>
  </UL>
</NAV>
```

Заключительный раздел страницы <FOOTER> используется для представления информации о сайте: автор, копирайт или ссылки на связанные веб-страницы.

Существует еще один секционный элемент <ADDRESS>. Он предназначен для представления контактной информации статьи или веб-страницы. Это может быть, например, ссылка на сайт автора или адрес его электронной почты.

```
<ADDRESS>
  Автор: <A HREF="mailto:matt@example.com">Иванов Д.</A>
</ADDRESS>
```

РАЗМЕТКА ТЕКСТА

Для размещения заголовков на странице предусмотрены теги заголовков, начинающихся буквой H (от слова Header – заголовок) и заканчивающихся номером уровня заголовка. Номеров всего 6. Каждый уровень отличается от предыдущего меньшим размером шрифта.

<H1>Заголовок 1 </H1>	Заголовок 1
<H2>Заголовок 2 </H2>	Заголовок 2
<H3>Заголовок 3 </H3>	Заголовок 3
<H4>Заголовок 4 </H4>	Заголовок 4
<H5>Заголовок 5 </H5>	Заголовок 5
<H6>Заголовок 6 </H6>	Заголовок 6

Перед заголовком и после него заложен определенный отступ. Абзац является **блочным элементом**. Это значит, что он занимает всю ширину страницы и следующий за ним элемент будет размещен на новой строке ниже, а не правее на той же линии.

Для размещения абзацев на странице предусмотрен тег абзаца <P> (от слова Paragraph – абзац). Данный тег является контейнером. К нему прилагается атрибут выравнивания абзаца по горизонтали ALIGN, значениями которого могут быть слова:

LEFT – выравнивание по левому краю
RIGHT – выравнивание по правому краю
CENTER – выравнивание по центру
JUSTIFY – выравнивание по ширине

<P ALIGN=CENTER> На моей домашней странице вы увидите фотоальбом с моим портретом и фото моих друзей, родственников и моей любимой собаки </P>

Однако в HTML5 вместо атрибута ALIGN используется таблица стилей.

<P STYLE="text-align: center;"> На моей домашней странице вы увидите фотоальбом с моим портретом и фото моих друзей, родственников и моей любимой собаки </P>

С помощью тега <P> можно задать пустой абзац, то есть пустую строку. Абзац, так же как и заголовок, является блочным элементом.

Внутри абзаца возможно выделять отдельные фрагменты текста: буквы, слова или предложения. Для этого их нужно заключить в следующие теги или использовать соответствующие свойства из таблиц стилей:

html-код	Вид в браузере	Пояснение	css-свойство
Пример	Пример	полужирный (Bold)	font-weight: bold
<I>Пример</I>	<i>Пример</i>	курсив (Italic)	font-style: italic
<U>Пример</U>	<u>Пример</u>	подчеркнутый (Underline)	text-decoration: underline
<S>Пример</S>	Пример	зачеркнутый (Stroke)	text-decoration: line-through
x²	x ²	верхний индекс (SUPerscript)	vertical-align: super
H₂<SUB>O	H ₂ O	нижний индекс (SUBscript)	vertical-align: sub

Чтобы перейти на новую строку внутри абзаца, используют тег
, BR – сокращение от слова Break – разрыв.
 не является контейнером, поэтому не имеет закрывающего тега.

Если символов для представления текстовой информации на клавиатуре не хватает, то можно воспользоваться специальными символами типа приведенных в таблице ниже*.

* См. подробнее: Википедия: специальные символы [Электронный ресурс]. Режим доступа: http://ru.wikipedia.org/wiki/Википедия:Специальные_символы. Таблица символов Юникода [Электронный ресурс]. Режим доступа: <https://unicode-table.com/ru/>.

§	§	®	®	°	°	—	—
©	©	™	™	±	±	‰	‰
√	√	∫	∫	∑	Σ	∏	Π
×	×	·	·	÷	÷	≅	≅
≈	≈	♠	♠	≡	≡	&neq;	≠
≤	≤	≥	≥	∈	∈	∉	∉
∩	∩	∪	∪	∅	∅	¬	¬
∧	∧	∨	∨	←	←	→	→
↔	↔	¼	1/4	½	½	¾	¾
∞	∞	£	£	¢	¢	€	€
¥	¥	★	★	☭	♣	☂	☂
☎	☎	♫	♪	♞	♠	α	α
β	β	γ	γ	δ	δ	ε	ε
Β	B	Γ	Γ	Δ	Δ	Ε	E

Символ пробел имеет код sp. Бывает необходим, например, для создания таблиц с пустыми ячейками.

Тег-контейнер <BLOCKQUOTE> добавляет поля слева и справа от текста так, что текст сжимается к центру страницы при многократном применении его к одному и тому же фрагменту текста. Так как данный тег не добавляет никакого содержательного элемента на веб-страницу, а служит лишь для внешнего оформления, он признан устаревшим и в строгой версии HTML не используется.

Тег пришел в HTML не сразу. Он не определяет никакого структурного элемента страницы. Он, благодаря своим атрибутам, определяет стилевое оформление текста, как и теги , <I>, <U>. Font переводится как «шрифт».

Атрибут SIZE определяет размер шрифта. Он может принимать значения от 1 до 7. Атрибут FACE задает тип шрифта. Причем можно задать сразу несколько вариантов шрифтов на тот случай, если при просмотре страницы на другом компьютере браузером не будет обнаружен заданный первый шрифт, тогда будет отыскиваться следующий по списку.

Пример: .

Атрибут COLOR определяет цвет шрифта. Его значениями могут быть либо одно шестнадцатеричное число из примерно 16 млн цветов, либо функция *rgb(α, β, γ)*, либо *rgba(α, β, γ, прозрачность)*, либо одно из следующих названий:

black	blue	green	turquoise	maroon	olive
черный	синий	зеленый	бирюзовый	насыщенный красно- коричневый	желтовато- зеленый (оливковый)
teal	gray	silver	lime	yellow	fuchsia
цвет чирка	серый	серебристый	оттенок зеленого	желтый	цвет фуксии
aqua	white	navy	purple	aquamarine	chocolate
цвет морской волны	белый	темно-синий (цвет формы морских офицеров)	пурпурный	аквамарин	шоколадный
darkred	gold	darkgoldenrod	darkkhaki	cadetblue	hotpink
тёмно- красный	золото		тёмный ха- ки	кадет	яркий розо- вый
wheat	tan	darkslateblue	darksal- mon	salmon	deeppink
пшеничный	загар		тёмный оранжево- розовый	оранжево- розовый	густой розо- вый
tomato	red	springgreen	snow	violet	whitesmoke
томатный	красный	весенне-зелёный	снежный	фиолетовый	белый дым

Более полную таблицу цветов см. по адресу https://www.w3schools.com/colors/colors_names.asp

Например: .



Однако с появлением каскадных таблиц стилей (CSS) тег FONT признан полностью устаревшим и не рекомендован к использованию. Вместо него используется таблица стилей, например,



<P style="font-size: 24pt; color: aqua; font-family: Tahoma;">

Если использовать число в качестве номера цвета, то необходимо поставить перед ним знак #, затем написать шесть 16-ричных цифр. Каждая пара цифр задает интенсивность красного, зеленого и синего луча. Например, #00FF00 определяет чистый зеленый цвет.

В Интернете по адресу <http://www.imagitek.com/hex> можно найти таблицу соответствия названия цветов 16-ричным числам, которую составил Calvin Htut. Здесь приведен ее фрагмент.

Значение css-свойства COLOR	Пояснение	Значение css-свойства COLOR	Пояснение
#000000	black	#D2B48C	tan
#000080	navyblue	#D3D3D3	lightgray
#00008B	blue4	#D4D4D4	gray83
#0000CD	blue3	#D6D6D6	gray84
#0000EE	blue2	#D8BFD8	thistle
#0000FF	blue1	#D9D9D9	gray85
#006400	darkgreen	#DA70D6	orchid
#00688B	deepskyblue4	#DAA520	goldenrod
#00868B	turquoise4	#DB7093	palevioletred
#008B00	green4	#DBDBDB	gray86
#008B45	springgreen4	#DC143C	crimson
#008B8B	cyan4	#DCDCDC	gainsboro
#009ACD	deepskyblue3	#DDA0DD	plum
#00B2EE	deepskyblue2	#DEB887	burlywood
#00BFFF	deepskyblue1	#8B8989	snow4
#00C5CD	turquoise3	#8B8B00	yellow4
#00CD00	green3	#8B8B7A	lightyellow4
#00CD66	springgreen3	#8B8B83	ivory4
#00CDCD	cyan3	#8C8C8C	gray55
#00CED1	darkturquoise	#8DB6CD	lightskyblue3
#262626	gray15	#8DEEEE	darkslategray2
#27408B	royalblue4	#8E388E	sgibeet
#282828	sgigray16	#8E8E38	sgiolivedrab
#292929	gray16	#8E8E8E	sgigray56
#2B2B2B	gray17	#8EE5EE	cadetblue2
#2E2E2E	gray18	#8F8F8F	gray56
#2E8B57	seagreen4	#8FBC8F	darkseagreen

Следующий тег выводит на страницу бегущую строку. Этот тег не вошел в стандарт языка, но поддерживается большинством браузеров.

<MARQUEE> Бегущая строка </MARQUEE>

Для управления параметрами бегущей строки существуют следующие атрибуты:

Атрибут	Значение атрибута	Комментарии
ALIGN	TOP / MIDDLE / BOTTOM	вертикальное выравнивание
BEHAVIOR	SCROLL / SLIDE / ALTERNATE	поведение (многократное перемещение / однократное перемещение / вперед-назад)

Атрибут	Значение атрибута	Комментарии
DIRECTION	LEFT / RIGHT/ DOWN / UP	направление движения
HEIGHT	число / %	высота области
WIDTH	число / %	ширина области
BGCOLOR	число / название цвета	цвет фона
LOOP	INFINITE / число	число повторений
SCROLLAMOUNT	число	скорость перемещения (от 1 до 10)
SCROLLDELAY	число	скорость до перерисовки (миллисекунд)
HSPACE	число	поле по вертикали
VSPACE	число	поле по горизонтали

Обычно на веб-странице размещается контактная информация, чтобы посетитель мог связаться с автором или владельцем веб-узла. Для оформления такой информации можно использовать тег <ADDRESS>.

<ADDRESS>e-mail: ivanov@yandex.ru</ADDRESS>

Как и в других языках программирования, в HTML есть тег комментариев.

<!-- - комментарии -->



Вопросы и упражнения

- Добавьте на созданную веб-страницу бегущую строку типа «I love web design» и с помощью параметров измените:
 - ширину и высоту контейнера на 500 px;
 - цвет фона с черного на цветной и цвет текста с белого на цветной;
 - направление перемещения строки на «сверху вниз»;
 - размер шрифта на величину менее 25 pt;
 - поэкспериментируйте с другими параметрами бегущей строки.
- Сколько абзацев (тег <P>) и заголовков (тег <H1>) может содержать html-документ? В каком разделе они должны находиться?

ЛОГИЧЕСКОЕ ФОРМАТИРОВАНИЕ ТЕКСТА

Теги логического форматирования предназначены в основном для отметки логических частей текстовой информации и явно не несут стилового оформления. Но стиливое оформление для них (как и для других тегов) можно закладывать с помощью CSS-каскадных таблиц стилей.

html-тег	Пояснение
<CITE>...</CITE>	от Citation – цитата. Служит для обозначения названия источника информации или автора
<Q> ... </Q>	обозначение цитаты из источника, берет текст в кавычки. У тега q есть атрибут с названием cite, который дает возможность обозначать ссылку на источник цитаты.
<p style="text-align: center;"><p></p> <p>Как сказал <cite>Антуан де Сент-Экзюпери</cite> в своей книге <cite>Маленький принц</cite>, <q>Таким был прежде мой Лис. Он ничем не отличался от ста тысяч других лисиц. Но я с ним подружился, и теперь он — единственный в целом свете.</q></p> <p style="text-align: center;"></p></p>	
<CODE>...</CODE>	вывод кода программы шрифтом фиксированной ширины
...	акцентирование текста от Emphasis – особое внимание, браузер выделяет такой текст курсивом
<KBD>...</KBD>	от Keyboard – клавиатура; вывод текста шрифтом фиксированной ширины, введенного с клавиатуры
<SAMP>...</SAMP>	от Sample – пример, образец; вывод нескольких символов шрифтом фиксированной ширины
<SMALL>...</SMALL>	мелкий текст, обозначающий отступления, предупреждения, ограничения или авторские права
<VAR>...</VAR>	от Variable – переменная
<TT>...</TT>	от Teletype – телетайп; текст, выведенный устройством вывода компьютера
<DFN>...</DFN>	от Definition – формулировка; термин, встречающийся в тексте первый раз
<p><P> <DFN> <ABBR TITLE="HyperText Markup Language">HTML</ABBR> </DFN> это основной язык разметки веб-контента. /P></p>	
<ACRONYM TITLE="Высшее учебное заведение"> ВУЗ </ACRONYM>	акроним; вид аббревиатуры, образованной из начальных букв слов или словосочетаний (например, НИИ, АЭС), произносимой как единое слово, а не по буквам
<ABBR> ... </ABBR>	аббревиатура

СПИСКИ

Список – это перечисление пунктов какой-либо совокупности. На языке HTML возможно создать 3 вида списков. Нумерованный список (Ordered List) открывается тегом . Маркированный список (Unordered List) открывается тегом . Каждый пункт списка начинается с (List Item).

<H4> Направления музыки</H4>

Классическая

Джаз

Рок

Фолк

<H4>Любимые композиторы</H4>

Моцарт

Бах

Вивальди

Направления музыки

1. Классическая

2. Джаз

3. Рок

4. Фолк



ЛАНЬ®

Любимые композиторы

- Вивальди
- Бах
- Моцарт

Список определений (Definition List) открывается тегом <DL>. Каждый определяемый термин начинается с <DT>, а определение с <DD>.

<DL>

<DT>fallacy (фэлеси)</DT>

<DD>заблуждение, ошибочный вывод;</DD>

<DT>fallibility (фэлибилити) </DT>

<DD>погрешимость, подверженность ошибкам/заблуждениям</DD>

</DL>

fallacy (фэлеси)

заблуждение, ошибочный вывод;

fallibility (фэлибилити)

погрешимость, подверженность ошибкам/заблуждениям

Для тега предусмотрены параметры и свойства стилей.

Атрибут	Значение	Пояснение	css-свойство
TYPE	1	нумерация арабскими числами	list-style-type: decimal
TYPE	A	нумерация заглавными латинскими буквами	list-style-type: upper-alpha
TYPE	a	нумерация строчными латинскими буквами	list-style-type: lower-alpha
TYPE	I	нумерация заглавными римскими буквами	list-style-type: upper-roman

Атрибут	Значение	Пояснение	css-свойство
TYPE	i	нумерация строчными римскими буквами	list-style-type: lower-roman
START	число-номер	начальный номер нумерации	
VALUE	число-номер	для тега устанавливает очередную номер.	

У тега тоже есть параметр TYPE (или свойство list-style-type), но вот со значениями не густо. По умолчанию работает значение DISC (круг), при желании можно установить SQUARE (квадрат) или CIRCLE (окружность).

Вопросы и упражнения

1. Разместите на веб-странице маркированный список любимых учебных дисциплин.
2. Измените маркеры списка на строчные римские цифры с помощью стилей.

СПИСКИ В СПИСКАХ (ВЛОЖЕННЫЕ СПИСКИ)

Для вложенных списков не существует специальных тегов. Просто контейнеры внутренних списков вкладываются в контейнер внешнего.

```
<H4>Любимая музыка</H4>
<OL TYPE=I>
<Li>Классическая
<UL> <Li>Моцарт
      <Li>Бетховен
      <Li>Паганини
</UL>
<Li>Авторская песня
<UL> <Li> Высоцкий
      <Li>Дольский
      <Li>Окуджава
      <Li>Визбор
</UL>
</OL>
```

Любимая музыка

I. Классическая

- Моцарт
- Бетховен
- Паганини

II. Авторская песня

- Высоцкий
- Дольский
- Окуджава
- Визбор

ТАБЛИЦЫ

Таблицы в документе используются не только для размещения в них текстовой информации, но и для размещения в них списков и графических рисунков с целью определения более жесткого их расположения относительно друг друга.

<TABLE> </TABLE>	Описание таблицы
<CAPTION> </CAPTION>	Заголовок таблицы
<TH> </TH>	Заголовок столбца
<TR> </TR>	Очередная строка таблицы
<TD> </TD>	Очередная ячейка таблицы

Например, нужно оформить на HTML следующую таблицу:

Прайс-лист

Наименование	Модель	Цена, р.
Процессор	Pentium II	2 800
Память	DIMM, 128 М	1 653
Материнская плата	IWILL VA133	3 500
CD-ROM	Creative, x48	1 125

Код будет выглядеть следующим образом:

```
<TABLE border>
<CAPTION> Прайс-лист </CAPTION>
<TR> <TH>Наименование</TH>
      <TH>Модель</TH>
      <TH>Цена, р.</TH>
</TR>
<TR> <TD>Процессор</TD>
      <TD>AMD Phenom</TD>
      <TD>2 800</TD>
</TR>
<TR> <TD>Память</TD>
      <TD>DIMM, 6 Г</TD>
      <TD>1 653</TD>
</TR>
<TR> <TD>Материнская плата</TD>
      <TD>IWILL, 4HT-S</TD>
      <TD>3 500</TD>
</TR>
<TR> <TD>BR-ROM</TD>
      <TD>Creative</TD>
      <TD>3 125</TD>
</TR>
</TABLE>
```

По умолчанию у таблицы нет внешних и внутренних границ. Чтобы они появились, в теге <TABLE> нужно приписать параметр BORDER. Есть и дру-

гие параметры, влияющие на внешнее представление таблицы и ее содержимого. В приведенной ниже таблице в круглых скобках указывается соответствующее устаревшему параметру свойство стиля.

Атрибут	Значение	Пояснение
ALIGN (margin)	left / right / center	выравнивание таблицы / содержимого ячеек по горизонтали
BACKGROUND (background-image)	путь к файлу-картинке	вставка фонового рисунка
BGCOLOR (background-color)	число / название цвета	цвет фона всей таблицы / строки / ячейки
BORDER (border)	число	толщина линий сетки таблицы
BORDERCOLOR (border или border-color)	число / название цвета	цвет линий сетки таблицы / строки / ячейки
BORDERCOLORDARK (border-style)	число / название цвета	цвет сетки с дополнительным эффектом трехмерности
BORDERCOLORLIGHT (border-style)	число / название цвета	цвет сетки с дополнительным эффектом трехмерности
CELLPADDING (padding)	число пикселей	ширина поля между содержимым ячейки и ее границей
CELLSPACING (border-spacing)	число пикселей	ширина промежутка между ячейками
HEIGHT (height)	число пикселей / число процентов	длина таблицы
VALIGN (vertical-align)	top / middle / bottom	выравнивание содержимого ячейки по вертикали относительно содержимого соседних ячеек строки
WIDTH (width)	число пикселей / число процентов	ширина таблицы

В таблице через знак «/» показаны различные варианты значений параметров и возможность применения параметра либо к тегу <TABLE>, либо к <TR>, либо к <TD>.

При описании таблицы необходимо внимательно следить за тем, чтобы число ячеек в каждой строке было одинаковым. Однако если нужно объединить несколько ячеек, то можно использовать следующие атрибуты для тега <TD> или <TH>:

Атрибут	Значение	Пояснение
ROWSPAN	число ячеек	объединяет ячейки по столбцам
COLSPAN	число ячеек	объединяет ячейки по строкам

<TABLE BORDER>

<TR> <TD ROWSPAN = 2>Процессор</TD>

<TD> Pentium II </TD>

<TD> 2 800 </TD>

</TR>

<TR> <TD> Celeron 500 МГц </TD>

<TD> 2 100 </TD>

</TR>

</TABLE>

Процессор	Pentium II	2
	Celeron 500	2

Для более гибкого оформления таблицы предусмотрены теги структурирования, располагающиеся внутри контейнера <TABLE>.

Теги	Пояснение
<THEAD> </THEAD>	объединяет строки заголовочной части (голова)
<TBODY> </TBODY>	объединяет строки основного содержания (тело)
<TFOOT> </TFOOT>	объединяет строки итоговой части таблицы (ноги)
<COLGROUP> </COLGROUP>	объединяет столбцы в группы для задания единого стиля
<COL> </COL>	задает характеристики одной или нескольких (span) колонок таблицы. Можно использовать совместно с тегом <colgroup>

<table cellpadding="2" rules="groups" border="3">

<colgroup width="90"> </colgroup>

<colgroup span="3" width="90"> </colgroup>

<colgroup span="3" width="90">

<col> <col style="border: 2px dashed red;"> <col> </colgroup>

<colgroup span="3" width="90"> </colgroup>

<tr style="background: black; color: white; font-weight: bold">

<td>Страна</td><td>Население в 1975</td><td>Население в

1976</td><td>Население в 1977</td><td>Женщины в 1975</td>

<td>Женщины в 1976</td><td>Женщины в 1977</td><td>Мужчины в

1975</td><td>Мужчины в 1976</td><td>Мужчины в 1977</td>

```

</tr>
<tr>
  <td>Австрия</td><td>7578903</td><td>7565525</td><td>7568430</td><
  td>4618455</td><td>4644100</td><td>4710300</td><td>4462800</td><t
  d>4517200</td><td>4558800</td>
</tr>
<tr>
  <td>Албания</td><td>2401602</td><td>2455497</td><td>2509909</td><
  td>2550193</td><td>2558469</td><td>2567052</td><td>2506844</td><
  td>2512827</td><td>2519844</td>
</tr>
<tr>
  <td>Андорра</td><td>26558</td><td>28500</td><td>29396</td><td>155
  4000</td><td>1572900</td><td>1621400</td><td>1581100</td><td>16395
  00</td><td>1660800</td>
</tr>
</table>

```

Страна	Население в 1975	Население в 1976	Население в 1977	Женщины в 1975	Женщины в 1976	Женщины в 1977	Мужчины в 1975	Мужчины в 1976	Мужчины в 1977
Австрия	7578903	7565525	7568430	4618455	4644100	4710300	4462800	4517200	4558800
Албания	2401602	2455497	2509909	2550193	2558469	2567052	2506844	2512827	2519844
Андорра	26558	28500	29396	1554000	1572900	1621400	1581100	1639500	1660800

В приведенном примере с помощью тегов `<COLGROUP>` в таблице выделен первый столбец и три группы по три столбца. Во второй группе с помощью тегов `<COL>` выделяется каждый столбец в группе, а средний из них стилем `border` обрамляется пунктирной линией.



Если ячейка не содержит никаких данных, поместите в нее пробел ** **; для корректного отображения в браузере.



Вопросы и упражнения

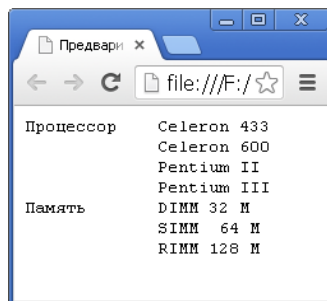
1. Какой параметр к тегу `<TABLE>` необходимо добавить для таблицы «Прайс-лист», чтобы появились граничные линии толщиной 3 пикселя?

2. Какой параметр к тегу <TABLE> необходимо добавить для таблицы «Прайс-лист», чтобы увеличить расстояние (поле) между содержимым ячейки и ее границей?
3. Добавьте пропущенный параметр, объединяющий две ячейки в строке <TR><TD _____ = 2></TD></TR>.
4. Каким свойством стиля можно заменить устаревший параметр BGCOLOR?

ПРЕДВАРИТЕЛЬНОЕ ФОРМАТИРОВАНИЕ

Существует особый контейнер <PRE> </PRE>, из которого заключаемая текстовая информация будет отображаться браузером в том виде, в каком она там хранится (со всеми пробелами). Это означает, например, что возможно расположить информацию в виде таблицы без использования тегов построения таблицы. Внутри контейнера допустимо применять любые теги кроме <BIG>, , <OBJECT>, <SMALL>, <SUB> и <SUP>.

```
<PRE>
Процессор   Celeron 433
              Celeron 600
              Pentium II
              Pentium III
Память      DIMM 32 М
              SIMM 64 М
              RIMM 128 М
</PRE>
```



ГРАФИКА

На веб-странице можно размещать графические рисунки, но только в трёх форматах: GIF, PNG и JPEG (гиф, пнг и джипег). Оба формата используют методы сжатия информации. Весьма желательно, чтобы размер каждой картинки, размещаемой на веб-странице, не превышал 25 килобайт. Размер всей страницы рекомендуют делать в пределах 150–200 килобайт.

JPEG. Формат создан группой экспертов в области фотографии (Joint Photographic Experts Group). Сжатие графики происходит с потерей качества. Коэффициент качества может достигать 1:100. Формат поддерживает 16,7 млн цветов. Используется для фотографических изображений. В качестве недостатка отмечают «дрожание линий», то есть потерю деталей вокруг четких контуров. На рисунках с четкими границами и большими заливочными областями сильно проявляются дефекты сжатия. Особенно характерно появление «грязи» вокруг темных линий на светлом фоне и видимых квадратных областей. Алгоритм сжатия обрабатывает изображение квадратами со стороной 8 пикселей. Формат позволяет определять прогрессивное отображение, когда при загрузке страницы на экране сначала появляется грубое изображение,

которое по мере получения дополнительной информации постепенно улучшается.

GIF – Graphical Interchange Format. В этом формате заложен метод сжатия LZW. Сжатие графики происходит без потерь качества. Есть поддержка прозрачного фона (любой один цвет может быть сделан прозрачным), чересстрочного вывода на экран. Однако формат поддерживает не более 256 цветов. Существует термин «анимированный GIF (гиф)». Им называют картинки, которые «оживают» на веб-странице. Анимированный гиф представляет собой слайд-фильм, который проигрывает браузер на веб-странице. Хранение последовательности кадров в одном файле – еще одна отличительная особенность формата GIF.

PNG – Portable Network Graphics (компактная сетевая графика). Относительно новый формат, конкурирующий с GIF. Он считается единственным форматом в Интернете, способным поддерживать полноцветные изображения с прозрачным фоном в сжатом состоянии без потерь качества. Так же как GIF и JPEG, поддерживает прогрессивную развертку в браузерах (interlased).

Графика на веб-странице служит для того, чтобы сделать страницу более привлекательной и, соответственно, более посещаемой. Все графические изображения можно примерно разделить на следующие группы:

ЛАНЬ® Линии и полосы



Пиктограммы-маркеры в списках

- ☺ Моцарт
- ☺ Вивальди
- ☺ Бетховен
- ☺ Паганини

```
<UL STYLE = "list-style-image: url(smile.gif)">  
<LI>Моцарт  
<LI>Вивальди  
<LI>Бетховен  
<LI>Паганини  
</UL>
```

Пиктограммы новинок



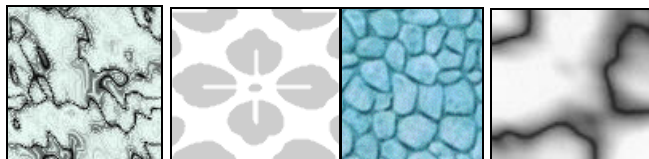
Пиктограммы реконструкции



Пиктограммы перемещения



Фоновые изображения



Фоновые изображения добавляются ко многим элементам веб-страницы и самой странице атрибутом BACKGROUND, где в качестве значения параметра указывается путь к файлу, хранящему образец фона. Браузер, размножая образец в необходимом количестве, «оклеивает» им весь задний план занимаемого элементом пространства. Таблица стилей предлагает аналогичное свойство **background-image**.

```
<MARQUEE STYLE = "background-image: url(bgpict15.jpg); height: 50px; width: 300px">
```

Существует еще один необязательный атрибут BGPROPERTIES для тега <BODY>. Если ему присвоить значение FIXED, то при скроллинге (прокручивании) содержимого страницы фон будет оставаться фиксированным (неподвижным). То есть содержимое будет как бы парить над фоновым рисунком.

```
<BODY BACKGROUND = "bgpict15.jpg" BGPROPERTIES = FIXED>
```

Тот же результат может быть достигнут при использовании вместо параметров аналогичных свойств из таблицы стилей.

```
<BODY STYLE="background-image: url(bgpict15.jpg); background-attachment: fixed;">
```

Картинки и фотографии



Рекламные баннеры и логотипы



Где взять файлы-картинки? **Во-первых**, их можно копировать непосредственно с веб-страниц, которые вы сами посещаете. Просто щелкните правой кнопкой мыши на понравившемся вам изображении и выберите из раскрывшегося списка команду «Сохранить рисунок как». В появившемся после этого диалоговом окне «Сохранение файла» выберите (или создайте) ту папку, в которой будут храниться все компоненты разрабатываемого html-документа.

Однако если просматривается корпоративный сайт, то есть очень большая вероятность того, что вся страница защищена законом от нелегального копирования. Это можно обнаружить по значку © со стоящим рядом именем владельца ресурса (копирайт: права копирования защищены собственником ресурса). Это значит, что прежде чем копировать ту или иную часть этой страницы, необходимо получить официальное разрешение у владельца. Не нарушая прав копирования, можно на своей веб-странице поставить ссылку на рисунок с внешнего сайта и тогда картинка будет отображаться и на вашей странице. Ссылка копируется в зависимости от браузера либо через контекстное меню и команду «Копировать ссылку на изображение» (Mozilla Firefox), либо через пункт «Свойства», где откроется адрес картинки, который тоже можно выделить и скопировать (Internet Explorer). Помещенная в буфер обмена ссылка на изображение затем вставляется в тег IMG в качестве значения параметра SRC.

Во-вторых, в сети Интернет есть специальные сайты, предлагающие множество коллекций графических изображений по различным разделам. Вам предложат как бесплатные ресурсы, так и платные. Вам, как и полагается, будет что выбрать. Одним из таких ресурсов является социальный сервис **Pixabay** [<https://pixabay.com/>].

В-третьих, вы можете воспользоваться любым графическим редактором для того, чтобы создать авторское оригинальное графическое изображение. Но не забудьте сохранить его в одном из сжатых форматов: GIF, PNG или JPEG.

Чтобы добавить любое графическое изображение на страницу, существует тег ****, от слова IMaGe – изображение. У него есть обязательный атрибут SRC, от слова SouRCe – источник.

```
<IMG SRC="путь к файлу-картинке">
```

В качестве значения параметра SRC может выступать:

1. *Абсолютный* путь к файлу (начинается с имени диска). Веб-разработчики никогда не используют этот вариант, так как при переносе сайта на сервер или другой компьютер ссылки работать правильно уже не будут.
2. *Корнезависимый* (начинается с папки, в которой находится html-документ; при этом имя папки не указывается).



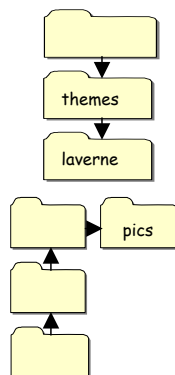
Пример.

```
<img src = "themes/laverne/alvbull1.gif">
```

3. *Документозависимый* (строится относительно веб-страницы, на которой находится ссылка; каждый уровень вверх обозначается "../").

Пример.

```
<img src = "../../pics/home.gif">
```



Необязательные атрибуты

ALT = “текст-пояснение”. Предусматривает текстовое сообщение, которое появится вместо картинки в том случае, если в браузере выключена опция просмотра графики.

Изображение является по умолчанию строчным элементом, в отличие от абзацев и заголовков. Поэтому если мы добавляем несколько изображений подряд, то они размещаются браузером в одной строке слева направо и выравниваются по нижнему краю. Чтобы изменить выравнивание или тип элемента, используется параметр ALIGN или соответствующие свойства из таблицы стилей.



Строгая версия HTML предполагает заключение строчных изображений в контейнеры, например, `<p>` или `<div>`, или `<table>`



Значения атрибута ALIGN приведены в таблице. По умолчанию оно имеет значение baseline.

Значение	Пояснение
TOP	по самому высокому элементу строки (“vertical-align: top”)
TEXTTOP	по самому высокому текстовому элементу строки (“vertical-align: text-top”)
MIDDLE	по середине строки (“vertical-align: middle”)
ABSMIDDLE	по середине текущей строки с учетом нижней части некоторых символов (например: р, д)
BASELINE / BOTTOM	по нижней части строки (“vertical-align: baseline”)
ABSBOTTOM	по самой нижней части строки, с учетом нижней части некоторых символов (например: р, д)
LEFT	изображение располагается слева от текста и не является элементом строки, а как бы плавает (“float: left”)

Значение	Пояснение
RIGHT	изображение располагается справа от текста и не является элементом строки, а как бы плавает ("float: right")

Строка может содержать как текст, так и графические изображения. Базовая линия — нижняя часть строки текста без учета самой нижней части некоторых символов (например: р, д). Остальные пять параметров приведены в таблице ниже.

Параметр	Значение	Пояснение
HSPACE	число	левое и правое поле от изображения ("margin: 0px 120px 0px 120px")
VSPACE	число	верхнее и нижнее поле от изображения ("margin: 120px 0px 120px 0px")
WIDTH	число	устанавливает размер изображения по ширине (width: 200px)
HEIGHT	число	устанавливает размер изображения по длине (height: 150px)
BORDER	число пикселей	выводит рамку вокруг изображения ("border: 1px solid blue ")

При использовании одного из параметров WIDTH или HEIGHT второй параметр автоматически изменяется.



Атрибуты ALIGN, HSPACE, VSPACE, BORDER не рекомендуются спецификацией HTML 4.01 и выше от W3C. Используйте свойства таблицы стилей.

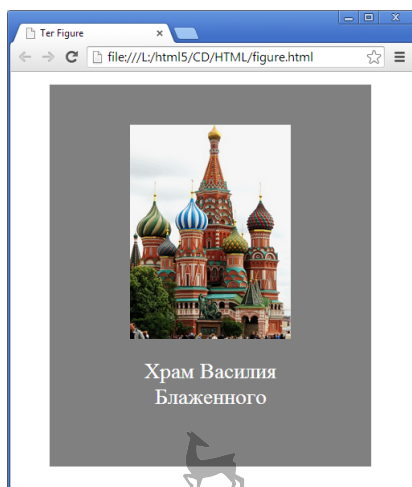


Картинку или фотографию при желании можно заключить в красивую рамку с помощью свойства border из таблицы стилей.

```
<P> <IMG SRC="images/itsme1.jpg" ALT="Моя фотография" STYLE =  
"border: 10px groove NavajoWhite"> </P>
```

В спецификацию HTML5 вошел тег <figure>, который используется для группирования любых элементов, например изображений и подписей к ним. Подпись заключается в теги <figcaption>.

```
<figure>  
    
  <figcaption> Храм Василия Блаженного </figcaption>  
</figure>
```



Для размещения простых разделительных линий в HTML предусмотрен специальный тег **<HR>** (Horizontal Rule) со следующими атрибутами. Все атрибуты не рекомендованы спецификацией HTML 4.01 и тем более HTML5. Вместо атрибутов лучше использовать таблицу стилей.

Атрибут	Значение	Пояснение
SHADE		тень (box-shadow)
NOSHADE		нет тени (box-shadow: none)
WIDTH	число / %	ширина линии (width)
ALIGN	LEFT / RIGHT / CENTER	горизонтальное выравнивание (margin)
SIZE	1...175 пикселей	толщина линии (height)
COLOR	название или номер цвета	цвет линии (color)

Пр и м е р.

<HR NOSHADE WIDTH=50% SIZE=6>



<HR WIDTH=70% SIZE=3 NOSHADE>



Адаптивные изображения

В связи с появлением огромного количества разнообразных устройств, обеспечивающих доступ ко Всемирной паутине и имеющих экраны с разрешением от 320px до 7680px, возникает необходимость в технологии изменения

макета веб-страницы в зависимости от устройства, на котором она отображается. В HTML5 появился в связи с этим новый тег <PICTURE> с медиазапросами для загрузки картинок разных размеров в зависимости от разрешения экрана.

Тег <PICTURE> является контейнером, в который помещаются теги <SOURCE> с медиазапросами для каждой картинке. Ниже приводится пример, который проверяет: если окно просмотра меньше 768 пикселей, то в этом случае загружается изображение меньшего размера (small.jpg), в противном случае – исходное изображение (default.jpg). В контейнере PICTURE включен и тег для совместимости со старыми версиями браузеров.

<PICTURE>

<SOURCE SRCSET="small.jpg" MEDIA="(max-width: 768px)">

<SOURCE SRCSET="default.jpg">

</PICTURE>

Для проверки можно создавать запросы для max-width, min-width, max-height, min-height и orientation. Свойство orientation может иметь одно из двух значений: portrait (книжная) или landscape (альбомная). Медиазапросы могут быть сложными и включать в себя логические союзы AND или OR.

<PICTURE>

<SOURCE SRCSET="smallLandscape.jpg" MEDIA="(max-width: 40em)
and (orientation: landscape)">

<SOURCE SRCSET="smallPortrait.jpg" MEDIA="(max-width: 40em)
and (orientation: portrait)">

<SOURCE SRCSET="defaultLandscape.jpg" MEDIA="(min-width: 40em)
and (orientation: landscape)">

<SOURCE SRCSET="defaultPortrait.jpg" MEDIA="(min-width: 40em)
and (orientation: portrait)">

</PICTURE>

Вопросы и упражнения

1. Чем отличаются форматы веб-графики?
2. Как можно добавить на свою страницу понравившуюся картинку с опубликованной в Интернете веб-страницы, не нарушая права копирования?
3. Почему рекомендуется заключать тег картинки в контейнер <DIV> или <P>?
4. Какие новые контейнеры появились в спецификации HTML5 для графических изображений и чем они отличаются?
5. Какой новый тег появился в спецификации HTML5 для подписи к графическим изображениям?

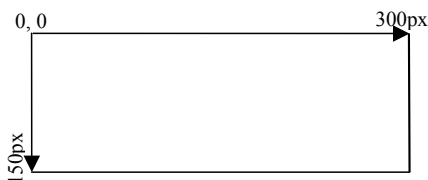
-
6. Каким образом браузер по умолчанию заполняет фоновую область элемента графическим рисунком?
7. Какое свойство фиксирует фоновый рисунок на экране при прокрутке содержимого страницы?

Графическое полотно <CANVAS>

Элемент <CANVAS> появился в 5-й версии HTML. Это очень мощный элемент, представляющий собой прямоугольное полотно или холст, на котором можно строить изображения. При этом изображения могут строиться на холсте динамически, после загрузки веб-страницы браузером.

`<CANVAS ID="холст1"> </CANVAS>`

По умолчанию полотно ничего не содержит и имеет размер 300×150px.



Все полотно покрыто координатной сеткой. Левый верхний угол – точка с координатами (0, 0). Правый верхний угол – (300, 0). Левый нижний угол – (0, 150). Таким образом имеем только положительные значения координат.

Размер полотна можно изменить параметрами WIDTH и HEIGHT или соответствующими стилями.

`<CANVAS ID="холст1" WIDTH="200" HEIGHT="100"> </CANVAS>`

Для того чтобы увидеть полотно на странице, можно задать цвет фона или фоновый рисунок и границу.

`<CANVAS ID="холст1" WIDTH="200" HEIGHT="100"
STYLE="border: 1px solid black; background-color: pink;">
</CANVAS>`

Для рисования на холсте или вывода текста используется язык программирования JavaScript. Скрипт заключается в теги <SCRIPT>.

Пример.

```
<SCRIPT TYPE="text/javascript">  
  var canvas = document.getElementById("холст1");  
  var ctx = canvas.getContext("2d");  
  ctx.fillStyle = "red";  
  ctx.fillRect(0,0,150,75);  
</SCRIPT>
```

Первая строка скрипта – это создание ссылки на полотно с именем «холст1». Вторая строка – задание графического объекта со встроенными свойствами и методами для рисования. Третья строка – задание красного цвета для заливки фигуры. Последняя строка скрипта – вывод прямоугольника по заданным параметрам.

На полотне с помощью встроенных методов можно по координатам рисовать цветные линии, прямоугольники, окружности, выводить текст. При этом есть разделение на закрашенные фигуры или буквы и контурные.

Линии

Рисовать линию на холсте можно по двум точкам. Метод `lineTo` рисует линию от точки, оставленной последним графическим оператором полотна. Первую точку линии на полотне можно также задать методом `moveTo`. Так как линия стрится по точкам, то она является составной фигурой и поэтому заключается в команды `beginPath` и `stroke`, что означает начало рисования фигуры и конец соответственно.

`moveTo` (координата по оси x, координата по оси y)

`lineTo` (координата по оси x, координата по оси y)

Пр и м е р.

```
ctx.beginPath();
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```

Пр и м е р. Нарисуем букву Г

```
ctx.lineWidth=10;
ctx.beginPath();
ctx.moveTo(10,170);
ctx.lineTo(10,120);
ctx.lineTo(50,120);
ctx.stroke();
```

У линий можно изменить некоторые параметры, например толщину линии. Все параметры собраны в следующей таблице.

Свойство линии	Значение	Пояснение	Пример
<code>lineWidth</code>	число	устанавливает новое значение толщины линии	<code>x.lineWidth=3</code>
<code>lineCap</code>	round square butt	устанавливает форму концов линий: круглые или квадратные	<code>x.lineCap='round'</code>

Свойство линии	Значение	Пояснение	Пример
lineJoin	round miter	round – сглаживает стыки двух линий; miter – не сглаживает стыки	x.lineJoin='round'
miterLimit	число большее 1	задает максимально допустимое расстояние для дорисовки концов соединенных линий	x.lineJoin = "miter" x.miterLimit = 5
strokeStyle	название rgb() gradient() pattern()	устанавливает новый цвет линии	ctx.strokeStyle=pat

Пр и м е р. Нарисуем линию с закругленным кончиком зеленого цвета и толщиной 10 пикселей.

```
ctx.beginPath();
ctx.strokeStyle = "green";
ctx.moveTo(50,50);
ctx.lineWidth=10;
ctx.lineCap='round';
ctx.lineTo(200,10);
ctx.stroke();
```

Пр и м е р. Нарисуем линию с рисунком вместо цвета. В разделе <BODY> разместим картинку с параметом ID="picLine".

```
var img=document.getElementById("picLine");
var pat=ctx.createPattern(img,"repeat");
ctx.lineWidth=30;
ctx.strokeStyle=pat;
ctx.moveTo(70,150);
ctx.lineTo(240,150);
ctx.stroke();
```



Прямоугольники

Прямоугольник на холсте рисуется по координатам левого верхнего угла прямоугольника, к которым добавляется длина и ширина. Полотно поддерживает два типа рисования прямоугольников – закрашенные методом fillRect() и незакрашенные (контурные) методом strokeRect(). Цвет заливки прямоуголь-

ника задается методом `fillStyle`, а цвет контура методом `strokeStyle`, при этом можно задать толщину контура методом `lineWidth`.

```
strokeRect (x, y, длина, ширина)
```

```
fillRect (x, y, длина, ширина)
```

```
clearRect (x, y, длина, ширина)
```

Метод `clearRect` очищает указанную прямоугольную область на холсте.

Пример. Нарисуем два прямоугольника: первый залит красным цветом, второй без заливки, с толщиной линий 10 пикселей синего цвета.

```
ctx.fillStyle = "red";  
ctx.fillRect(0,0,150,75);  
ctx.strokeStyle = "blue";  
ctx.lineWidth=10;  
ctx.strokeRect(10,10,150,75);
```

Заливка прямоугольника может быть не просто цветом, но и градиентом, и рисунком, и полупрозрачным цветом.

Окружности и дуги

Полотно поддерживает метод `arc()`, который позволяет рисовать окружности и их части, заданные в радианах.

```
arc (x, y, радиус, начальный угол, конечный угол)
```

Чтобы нарисовать окружность, необходимо задать все параметры метода `arc`. Начальный угол u окружности равен 0 радиан, конечный – 2π радиан (соответствует 360 градусам). Число π равно примерно 3,14. В JavaScript 2π задается выражением `2*Math.PI`, где `Math` – это объект JavaScript, отвечающий за многие математические константы и функции.

Пример. Нарисуем окружность с центром в точке (150, 100), радиусом 50 пикселей и толщиной линии 10 пикселей.

```
ctx.beginPath();  
ctx.lineWidth=10;  
ctx.arc (150,100,50,0,2*Math.PI);  
ctx.stroke();
```

Пример. Нарисуем круг (закрашенную окружность) с центром в точке (100, 70), радиусом 50 пикселей.

```
ctx.beginPath();  
ctx.arc (100,70,50,0,2*Math.PI);  
ctx.fill();
```

Пр и м е р. Нарисуем полуокружность с углами от 0 до π радиан.

```
ctx.beginPath();
ctx.lineWidth=10;
ctx.arc
(50,150,45,0,Math.PI);
ctx.stroke();
```

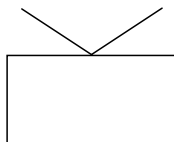


Составные фигуры

Сложные фигуры состоят из комбинации простых фигур – линий, прямоугольников и окружностей. В следующем примере показано, как можно задавать координаты через переменные величины, что позволяет впоследствии более гибко редактировать рисунок.

Пр и м е р. Нарисуем телевизор с антенной.

```
ctx.clearRect(0,0,300,200)
ctx.beginPath();
w=5
ctx.lineWidth=w;
x=100
y=60
ctx.moveTo(x,y);
ctx.strokeRect(x,y,100,75);
ctx.moveTo(x+50,y-w);
ctx.lineTo(x+5,y-30);
ctx.moveTo(x+50,y-w);
ctx.lineTo(x+95,y-30);
ctx.stroke();
```



Для рисования замкнутых составных фигур удобно использовать метод `closePath()`.

`closePath ()`

Пр и м е р. Нарисуем закрашенный треугольник.

```
ctx.clearRect(0,0,300,200)
ctx.beginPath();
ctx.fillStyle="#db7093";
ctx.moveTo(20, 20);
ctx.lineTo(20, 100);
ctx.lineTo(70, 150);
ctx.closePath();
ctx.fill();
```



Изображения

На холст можно нанести изображение, хранящееся в файле и размещенное на веб-странице, с помощью метода `drawImage`.

```
drawImage (img, x, y)
```

```
drawImage (img, x, y, длина, ширина)
```

```
drawImage (img, x1, y1, длина1, ширина1, x2, y2, длина2, ширина2)
```

Здесь *img* – это идентификатор изображения, *x* и *y* – координата левого верхнего угла изображения на холсте, *длина* и *ширина* – размеры изображения на холсте.

Второй вариант метода используется для масштабирования исходного изображения. Если *длина* и *ширина* исходного изображения больше *длина* и *ширина* указанных в методе `drawImage`, то браузер изменит масштаб исходного изображения для помещения его целиком на холст в указанное место.

Третий вариант метода `drawImage` используется для вырезки части изображения и помещения фрагмента на холст в указанное место. Здесь *x1*, *y1* – координата левого верхнего угла фрагмента из исходного изображения; *длина1*, *ширина1* – размеры фрагмента исходного изображения; Здесь *x2*, *y2* – координаты левого верхнего угла фрагмента на холсте; *длина2*, *ширина2* – размер вырезанного фрагмента на холсте.

Благодаря новому атрибуту HTML5 **hidden** можно скрыть размещенное вне холста изображение.

```

```

Пр и м е р. Добавим в составное изображение телевизора из предыдущего примера картинку.

```
var canvas = document.getElementById("холст1");  
var ctx = canvas.getContext("2d");  
var img2=document.getElementById("tv");  
width=100; height=75;  
ctx.drawImage(img2,x+w,y+w,width-2*w,height-2*w);
```



Пр и м е р. Вырежем из исходного изображения фрагмент и поместим на холст в телевизор.

```
ctx.drawImage(img2,300,170,320,200,x+w,y+w,width-2*w,height-2*w);
```

Кроме рассмотренных фигур (линии, прямоугольники, окружности, составные фигуры) на холст можно выводить еще и кривые Безье методами `quadraticCurveTo()` и `bezierCurveTo()`, о которых можно прочитать в дополнительной литературе.



Свойства фигур холста

Выше уже рассматривались свойства `fillStyle` и `strokeStyle`, предназначенные для заливки фигур цветом или рисунком или заливки контура фигуры соответственно. К этим свойствам можно добавить градиентную заливку и создание тени.

Градиентная заливка

Градиент – это плавный переход цвета от светлых тонов к темным или наоборот, при использовании нескольких цветов – плавный переход от одного цвета к другому. Градиентом заливают фигуры или граничные линии (`fillStyle` или `strokeStyle`). Различают линейный и радиальный градиенты. Создать линейный градиент для холста можно методом `createLinearGradient`.

```
createLinearGradient(x0,y0,x1,y1)
```

Здесь `x0`, `y0` – координаты начальной точки градиента; `x1`, `y1` – координаты конечной точки градиента.

Координаты в методе задают направление градиента: слева направо или справа налево, сверху вниз или снизу вверх, от верхнего левого угла к правому нижнему и т. д.

Цвет текущей точки градиента добавляет метод `addColorStop`.

```
градиент.addColorStop(стоп, цвет)
```

Здесь `стоп` – число в диапазоне от 0.0 до 1.0, которое представляет позицию между началом и концом градиента; `цвет` – один из способов задания цвета.

Пример. Нарисуем залитый градиентом прямоугольник.

```
var grd=ctx.createLinearGradient(0,0,100,0);
grd.addColorStop(0,"black");
grd.addColorStop("0.3","magenta");
grd.addColorStop("0.5","blue");
grd.addColorStop("0.6","green");
grd.addColorStop("0.8","yellow");
grd.addColorStop(1,"red");
ctx.fillStyle=grd;
ctx.fillRect(20,20,250,150);
```



Радиальный градиент для холста создается методом `createRadialGradient`.

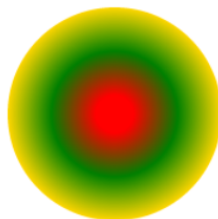
```
градиент.createRadialGradient(x0,y0,r0,x1,y1,r1)
```

Здесь x_0, y_0 – координата центра внутреннего круга распространения радиального градиента; r_0 – радиус внутреннего круга распространения градиента; x_1, y_1 – координата центра внешнего круга распространения радиального градиента; r_1 – радиус внешнего круга распространения градиента.

Задать цвет радиального градиента можно тем же методом `addColorStop`, что и для линейного градиента.

П р и м е р. Нарисуем залитый градиентом круг.

```
var rgrd =
ctx.createRadialGradient
(150,100,15,150,100,85);
rgrd.addColorStop(0,"red");
rgrd.addColorStop(0.5,"green");
rgrd.addColorStop(1,"gold");
ctx.fillStyle=rgrd;
ctx.beginPath()
ctx.arc(150,100,85,0,Math.PI*2)
ctx.fill()
```



Тень от фигуры

Тень представляет собой копию фигуры, отстоящую от оригинала с некоторым смещением по горизонтали и вертикали. Для создания тени на холсте предусмотрены специальные свойства.

Свойство тени	Значение	Пояснение	Пример
<code>shadowOffsetX</code>	число пикселей	смещение тени от объекта по горизонтали	<code>x.shadowOffsetX=3</code>
<code>shadowOffsetY</code>	число пикселей	смещение тени от объекта по вертикали	<code>x.shadowOffsetY=5</code>
<code>shadowBlur</code>	число пикселей	размытость тени	<code>x.shadowBlur=5</code>

Свойство тени	Значение	Пояснение	Пример
shadowColor	цвет	цвет тени	x.shadowColor='red'

Пр и м е р. Нарисуем прямоугольник с тенью

```
ctx.shadowOffsetX=3;
ctx.shadowOffsetY=5;
ctx.shadowBlur=8;
ctx.shadowColor='black';
ctx.fillStyle='#ffaa00';
ctx.fillRect(50,40,55,55);
```



Прозрачность

Прозрачность цвета можно задавать через известную ранее функцию `rgba()`. Однако для холста есть еще так называемое свойство «Глобальная прозрачность». После установки значения глобальной прозрачности все выводимые фигуры будут ее иметь.

Свойство	Значение	Пояснение	Пример
globalAlpha	число от 0 до 1	устанавливает или возвращает глобальную прозрачность	ctx.globalAlpha=0.2

Текст

На холст можно выводить текст также двумя стилями, как и фигуры – контурные буквы текста или сплошной заливки. Соответственно, существуют два метода – **strokeText** и **fillText** для вывода текста. Еще один метод **measureText** предназначен для измерения ширины текстового сообщения. Например, прежде чем вывести сообщение, можно проверить уместается ли оно на холст.

fillText (текст, x-координата, y-координата, максимальная ширина)
strokeText (текст, x-координата, y-координата, максимальная ширина)
measureText(текст).width

Для стилизации текста предусмотрены ряд свойств со множеством значений.



Свойство текста	Значение	Пояснение	Пример
font	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size</i> <i>font-family</i>	изменяет или возвращает свойства шрифта	<code>ctx.font="30px Arial";</code> <code>ctx.font="italic small-caps bold 12px arial"</code>
textAlign	start end left right center	изменяет или возвращает тип выравнивания текста по горизонтали относительно вертикальной линии, проходящей через заданную координату	<code>ctx.textAlign = "center";</code> <code>ctx.fillText("Мой текст",150, 120);</code>
textBaseline	top bottom middle alphabetic hanging	изменяет или возвращает тип выравнивания текста по вертикали относительно горизонтальной линии, проходящей через заданную координату	<code>ctx.textBaseline="top";</code> <code>ctx.fillText("Мой текст",5,100);</code>

Пр и м е р. Напечатаем на холсте текст буквами сплошной заливки и контурной заливки.

```
var c=document.getElementById('myCanvas');
var ctx=c.getContext('2d');
ctx.font='30px Arial';
ctx.fillText('Hello World',20,50,200);
ctx.strokeText("Hello World!",20,90,200);
```

Пр и м е р. Напечатаем текст на холсте с градиентной заливкой: сплошной и контурной.

```
ctx.font="30px Verdana";
// Создаем градиент
var gradient=ctx.createLinearGradient(0,0,c.width,0);
gradient.addColorStop("0","magenta");
gradient.addColorStop("0.5","blue");
```



```
gradient.addColorStop("1.0","red");
// Выводим текст
ctx.fillStyle=gradient;
ctx.fillText("Красивый текст",20,130);
ctx.strokeStyle=gradient;
ctx.strokeText("Красивый текст",20,170);
```

Пр и м е р. Напечатаем текст на всю ширину холста с заливкой рисунком.

```
var img=document.getElementById("picLine");
var pat=ctx.createPattern(img,"repeat");
ctx.font="50px Impact";
ctx.fillStyle=pat;
txt="Очень красивый текст";
//Изменяем ширину холста под ширину текста
canvas.width=ctx.measureText(txt).width;
//Снова задаем параметры текста
ctx.font="50px Impact";
ctx.fillStyle=pat;
ctx.fillText(txt,0,130);
```

Кроме рассмотренных методов по рисованию фигур, холст поддерживает следующие методы:

clip()	определяет область для рисования, вне которой ничего выводиться не будет
bezierCurveTo()	выводит на холст кривую Безье
quadraticCurveTo()	выводит на холст кривую Безье второго порядка
arcTo()	рисует дугу, заключенную между двумя отрезками
isPointInPath()	возвращает true, если указанная точка находится внутри контура, и false — если вне его
globalCompositeOperation	определяет, как будут себя вести перекрывающиеся фигуры

Трансформация

К фигурам, размещенным на холсте, можно применить методы трансформации: изменение масштаба scale, поворот rotate, сдвиг translate и матрицу трансформаций transform и setTransform.

scale(масштаб по ширине, масштаб по длине)

rotate(угол поворота)

ranslate(сдвиг по x, сдвиг по y)

transform(масштаб по x, скос по x, скос по y, масштаб по y, сдвиг по x, сдвиг по y)

setTransform(масштаб по x, скос по x, скос по y, масштаб по y, сдвиг по x, сдвиг по y)

Масштаб задается числами: 1 = 100%, 0,5 = 50%, 2 = 200% и так далее. Например, scale (2,2).

Поворот фигуры осуществляется вокруг начала координат, точки с координатами (0,0). Угол поворота задается в радианах от 0 до 2π радиан (2 π соответствует 360 градусам). Число π равно примерно 3,14. Для перевода градусов в радианы можно использовать формулу

$$\text{угол поворота} * \text{Math.PI} / 180.$$

Например, чтобы повернуть фигуру на 30 градусов по часовой стрелке, задаем выражение $30 * \text{Math.PI} / 180$.

Сдвиг фигуры задается в пикселях по оси x и по оси y. Например, translate(70,70). Возможно использовать отрицательные числа для сдвига в противоположную сторону.

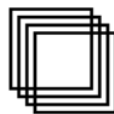


После задания метода трансформации, все применяемые методы рисования будут учитывать эту трансформацию.



Пр и м е р. Нарисуем квадрат и его копии со сдвигом по x и по y.

```
var canvas = document.getElementById("canvas16");
var ctx = canvas.getContext("2d");
ctx.strokeRect(150, 100, 50, 50);
ctx.translate(5,5);
ctx.strokeRect(150, 100, 50, 50);
ctx.translate(5,5);
ctx.strokeRect(150, 100, 50, 50);
ctx.translate(5,5);
ctx.strokeRect(150, 100, 50, 50);
```



Метод **transform**() заменяет текущую матрицу трансформации элемента холста. С помощью этого метода можно изменить одновременно масштаб, сдвиг, поворот и скос элемента.

Метод **setTransform**() действует точно также, но в отличие от transform каждый раз обнуляет предыдущую трансформацию, а метод transform() накладывается поверх нее. Этот метод удобен при работе с несколькими объектами при условии наложения трансформации автономно на каждый объект.

Параметры обоих методов задаются положительными и отрицательными целыми и дробными числами.

При значениях `transform(1,0,0,1,0,0)` трансформации не происходит, так как 1 означает отсутствие изменений масштаба.

Пример. Разместим на холсте рисунок и преобразуем его с эффектом «скос» по оси у.


```
var canvas = document.getElementById("canvas18");  
var ctx = canvas.getContext("2d");  
var img=document.getElementById("hru");  
ctx.transform(1,0,0.4,1,0,0);  
ctx.drawImage (img,30,30,150,100)
```



Вопросы и упражнения

1. Какие размеры по умолчанию имеет графическое полотно `canvas`?
2. В каком месте полотна находится точка с координатами (0,0)?
3. Какие фигуры рисуют графические методы полотна?
4. Сколько радиан соответствует 180 градусам?
5. Чем отличается метод `transform()` от `setTransform()`?
6. Каким образом можно разместить текст на рисунке полотна?
7. В каком диапазоне изменяется значение свойства «Прозрачность»?
8. С помощью какого метода полотна можно сделать фигуру скошенной?
9. Как работает глобальная прозрачность?

ГИПЕРССЫЛКИ

Просматривая веб-страницы во Всемирной паутине, обязательно встретить фрагменты текста или графические изображения, при наведении на которые курсора мыши он превращается в кисть руки с вытянутым указательным пальцем . В этот момент в статусной строке браузера появляется адрес (URL) страницы или иного ресурса, который будет загружен в том случае, если щелкнуть левой кнопкой мыши.

Текст, который содержит ссылки на ресурсы Интернета, называется *гипертекстом* (термин ввел Тед Нельсон). Документы, на которые ссылается такой текст, могут находиться как на удаленном компьютере в сети Интернет, так и на том же компьютере, где находится просматриваемая страница. Сами указатели называют *гиперссылками*.

Гиперсвязи между различными веб-документами одинаковой тематики могут выглядеть так, как показано на рисунке 1.

Тег <A> является контейнером. Между открывающимся и закрывающимся тегами размещается текст, который становится при просмотре в браузере гипертекстом.

 Ростелеком

Гипертекстовые ссылки в браузере по умолчанию выделяются синим цветом и подчеркиванием (Link). При щелчке мышью ссылка становится активной (Active Link) и меняет свой цвет. Если вернуться на эту же страницу после посещения ссылки (Visited Link), то ее цвет поменяется на третий вариант. Таким образом, различают три состояния ссылки, и каждому из них соответствует свой цвет, который может быть изменен разработчиком страницы с помощью стилей. Например, можно следующим образом добавить стили в заголовочный раздел html-документа:

<HEAD>

<STYLE TYPE = "text/css">

A:link {color: yellow}

A:visited {color: aqua}

A:active {color: lime}

</STYLE>

</HEAD>



Группу ссылок можно оформить в виде списка. Например, в виде списка поисковых систем Интернета.

<H2> Наиболее популярные каталоги Интернета </H2>

 Свободный каталог

 Желтые страницы INTERNET

 Diamond Team - каталог русскоязычных ресурсов Internet

 LINKMAKER! - Каталог INTERNET ресурсов

 Каталог ресурсов URL

 Улитка

 Со-кровища Интернета. Россия и ближнее зарубежье

 Карта русскоязычных ресурсов Интернета



В виде гиперссылки можно оформить и любое графическое изображение.

```
<A HREF="http://www.yandex.ru"> <IMG SRC="logo2.gif"> </A>
```

ВНУТРЕННИЕ ССЫЛКИ И УКАЗАТЕЛИ

В том случае, если ссылка осуществляется на внутреннюю страницу сайта, в качестве URL указывается просто путь к файлу, без указания протокола связи.

```
<A HREF = "fotogalary.html"> Галерея фотографий </A>
```

```
<A HREF = "../..index.html"> На главную </A>
```

В случае, когда html-документ очень вытянут в длину (не умещается на нескольких экранах), удобно составить небольшое оглавление в начале страницы в виде списка ссылок, которые будут перемещать выбранные щелчком разделы документа сразу на экран, чтобы их можно было не искать на странице. В конце каждого такого раздела размещается гиперссылка, направляющая обратно на начало страницы, где находится оглавление.

Чтобы создать такой указатель, необходимо в тексте html-документа расставить именные якоря в начале каждого раздела, на который затем будет осуществляться ссылка. Для этого предназначен тот же тег <A>, но с атрибутом NAME, значение которому дается разработчиком веб-документа. Это значение является уникальным именем текущего раздела.



Гиперссылки относятся к строчным элементам, поэтому в соответствии со спецификацией версии strict должны заключаться в блочный контейнер, например <P>.



```
<HTML>
```

```
<BODY>
```

```
<H1> <A NAME="содержание"> КОНЦЕПЦИЯ МАТЕМАТИЧЕСКОГО  
ОБРАЗОВАНИЯ </A> </H1>
```

```
<P> <A HREF="#раздел1">Подготовка и повышение квалификации  
педагогических кадров </A>
```

```
<A HREF="#раздел2">Значение математического образования </A>
```

```
<A HREF="#раздел3">Общие принципы</A>
```

```
</P>
```

```
<H2> <A NAME="раздел1"> Подготовка и повышение квалификации  
педагогических кадров </A> </H2>
```

```
<P>Сложившаяся система воспроизводства педагогических кадров со-  
храняет традиционную приверженность к «знаниевой» парадигме пе-  
дагогического образования. Между тем, природа современной про-  
фессиональной деятельности требует учителя, ценностными установ-  
ками которого являются приоритет личностного развития школьни-
```

ков, его способность свободно ориентироваться в сложных социо-культурных обстоятельствах.</P>

<P> Вернуться в начало </P>

<H2>

Значение математического образования </H2>

<P>В течение многих столетий математика является неотъемлемым элементом системы общего образования всех стран мира. Объясняется это уникальностью роли учебного предмета «Математика» в формировании личности. </P>

<P>Универсальный элемент мышления — логика. Полноценное развитие мышления современного человека, осуществляемое в ходе самопознания и общения с другими людьми, в ходе рассуждений и знакомства с образцами мышления, невозможно без формирования известной логической культуры. </P>

<P>Вернуться в начало</P>

<H2> Общие принципы </H2>

<P>Главный принцип концепции математического образования в 12-летней школе, направленный на осуществление этих идей, состоит в реальном осуществлении в методической системе обучения математике двух генеральных функций школьного математического образования, определяемых глобальным совпадением и локальными различиями общественных и личных интересов в математических знаниях и математической культуре:</P>

<OL style="list-style-type: lower-alpha">

 образование с помощью математики,

 собственно математическое образование.

<P> Вернуться в начало </P>

</BODY>

</HTML>

Из примера видно, что оформлено гипертекстовое оглавление разделов страницы в виде одной строки таблицы с вертикальными границами. Здесь надо обратить внимание в коде HTML на то, что URL внутренней ссылки начинается не с протокола связи, а со знака #, после которого идет имя раздела, данное тегом .



КОНЦЕПЦИЯ МАТЕМАТИЧЕСКОГО ОБРАЗОВАНИЯ

Подготовка и повышение
валификации педагогических
кадров

Значение математи-
ческого
образования

Общие
принципы

Подготовка и повышение квалификации педагогических кадров

Сложившаяся система воспроизводства педагогических кадров сохраняет традиционную приверженность к «знаниевой» парадигме педагогического образования. Между тем, природа современной профессиональной деятельности требует учителя, ценностными установками которого являются приоритет личностного развития школьников, его способность свободно ориентироваться в сложных социокультурных обстоятельствах, готовность обслуживать не только инновационные процессы, но и процессы творчества в широком смысле.

Вернуться в начало.

Значение математического образования

В течение многих столетий математика является неотъемлемым элементом системы общего образования всех стран мира. Объясняется это уникальностью роли учебного предмета «Математика» в формировании личности. Образовательный, развивающий потенциал математики огромен.

Универсальный элемент мышления – логика. Полноценное развитие мышления современного человека, осуществляемое в ходе самопознания и общения с другими людьми.

Вернуться в начало.

Общие принципы

Главный принцип концепции математического образования в 12-летней школе, направленный на осуществление этих идей, состоит в реальном осуществлении в методической системе обучения математике *двух генеральных функций* школьного математического образования, определяемых глобальным совпадением и локальными различиями общественных и личных интересов в математических знаниях и математической культуре:

- a. *образование с помощью математики,*
- b. *собственно математическое образование.*

Вернуться в начало

Существует реальная возможность делать ссылку и на именованную таким же образом часть веб-страницы, расположенную на удаленном компьютере в сети Интернет. Такая ссылка будет выглядеть примерно так:

Профессия Энди Шафрана – автора книги «Создание Web-страниц»

Детали

В спецификации HTML5 появился тег <details>, который может показать или скрыть дополнительную информацию по требованию посетителя. Основной контент размещается в контейнере <summary>.

<details>

<summary>Александр Сергеевич Пушкин</summary>

<p> 26 мая 1799, Москва — 29 января 1837, Санкт-Петербург</p>

<p>русский поэт, драматург и прозаик, заложивший основы русского реалистического направления, критик и теоретик литературы, историк, публицист; один из самых авторитетных литературных деятелей первой трети XIX века. </p>

</details>

► Александр Сергеевич Пушкин

Браузеры по умолчанию выводят основной контент, дополнительный показывается и скрывается по щелчку указателя мыши. Чтобы посетитель догадался о наличии дополнительной информации, слева от основной выводится черный треугольник, одной из вершин развернутый к тексту, после щелчка и вывода дополнительного контента, вершина треугольника поворачивается к нему.

Вопросы и упражнения

1. Что такое гиперссылка?
2. Может ли быть гиперссылкой не только текст, но и графическое изображение?
3. Как расшифровывается и что такое URL?
4. В каких состояниях может находиться гиперссылка?
5. Каким образом можно изменить заданные по умолчанию цвета различных состояний гиперссылки?
6. На какие виды делятся гиперссылки?
7. Является ли гиперссылка блочным элементом?
8. Какими тегами можно расставить именные якоря в html-документе?
9. Как организовать гиперссылку на отдельный кадр презентации?
10. Кто ввел термин гипертекст?
11. Разместите на веб-странице список гиперссылок на пять любимых веб-сайтов Интернета.
12. Разместите на веб-странице три термина со скрытой поясняющей информацией.

ИЗОБРАЖЕНИЯ-КАРТЫ

Изображения-карты – это такие графические изображения, отдельные части которых являются гиперссылками. Удобны они из-за того, что наглядно показывают содержание имеющейся ссылки и объединяют группу ссылок в единое целое.



Практически любое изображение можно превратить в изображение-карту. Каждому фрагменту изображения соответствуют собственные координаты, выражаемые в пикселях. Левый верхний угол карты имеет координату (0,0).

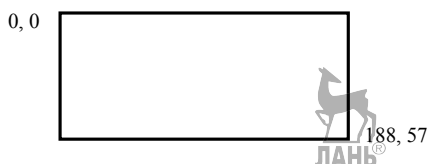
В html-документе необходимо открыть именной раздел-контейнер для описания изображения-карты.

<MAP NAME="Карта 1"> </MAP>

В этом разделе для каждой области карты, которая будет служить ссылкой, размещается тег **<AREA>** с параметрами. Существует возможность описать области, имеющие форму прямоугольников, многоугольников и кругов.

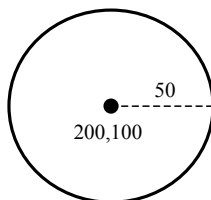
ПРЯМОУГОЛЬНИК (RECTangle)

<AREA SHAPE=RECT COORDS="0, 0, 188, 57" HREF = "page1.html">



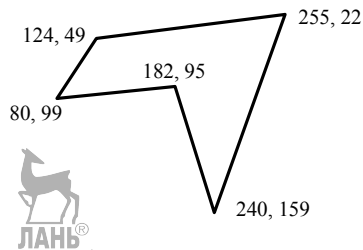
КРУГ (CIRCLE)

<AREA SHAPE=CIRCLE COORDS="200, 100, 50" HREF = "page2.html">



МНОГОУГОЛЬНИК (POLYgon)

<AREA SHAPE=POLY COORDS="80,99, 124,49, 255,22, 240,159, 182,95, 80,99" HREF = "page3.html">



Если фрагменты имеют область перекрытия, то срабатывает ссылка того фрагмента рисунка, который стоит первым в описании областей.

ССЫЛКА ПО УМОЛЧАНИЮ

При желании можно установить общую ссылку для всех тех частей изображения-карты, где не были определены ссылки. Такую ссылку и называют ссылкой по умолчанию. Она описывается последней в контейнере <MAP> и должна перекрывать формой все изображение-карту.

После того как все области описаны и задана, при желании, ссылка по умолчанию, в нужное место html-документа вставляется непосредственно само изображение-карта тегом с параметром USEMAP, значением которого является название изображения, данное параметром NAME тега <MAP>.

Для облегчения создания изображения-карты существуют специальные визуальные инструменты в редакторах создания веб-сайтов и соответствующие интернет-сервисы. Сначала загружается изображение в формате GIF или JPEG, а затем визуальным образом, с помощью необходимых инструментов, размечается изображение на области и вносится в диалоговое окно адреса гиперссылок. Редактор автоматически генерирует соответствующий html-код.

Вопросы и упражнения

1. Какие изображения не подойдут для разметки в виде карты?
2. Какими геометрическими фигурами можно размечать графическое изображение?
3. Что такое «ссылка по умолчанию» в разметке карты?

МУЛЬТИМЕДИА

Звукозаписи, презентации, анимацию и видеофильмы в html-документе размещают разными способами. Первый способ состоит в использовании уже известного тега <A> – гиперссылки.

** Прослушайте пение моей домашней птички **

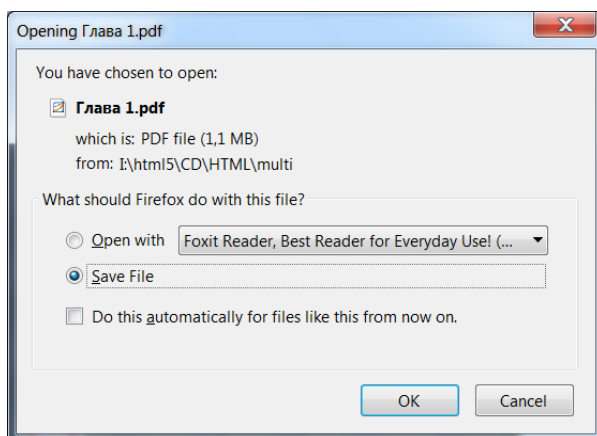
** Посмотрите фильм о моей собачке (1,5 М) **

Подобным образом можно размещать на веб-странице ссылки на любые типы документов (файлы MS Word, презентации и так далее).

`Презентация о теореме Пифагора `

На веб-странице появляется текст гиперссылки на документ, при щелчке на который браузер либо открывает документ сам, либо, если встречает незнакомый ему формат, предлагает посетителю открыть документ другим приложением или просто сохранить его в папке загрузки. В HTML5 появился новый атрибут **download** для тега гиперссылки, который заставляет браузер независимо от формата документа всегда предлагать сохранить документ на локальном компьютере или открыть его локальным приложением. Значением атрибута **download** является имя скачиваемого файла. Оно может не совпадать с исходным именем, указанным как значение параметра **href**.

` Скачай 1 главу книги по компьютерным сетям `



Еще один способ внедрения аудио- и видеоинформации заключается в применении специальных тегов-контейнеров `<AUDIO>` и `<VIDEO>`, которые загружают и проигрывают звуковой или видеофайл вместе со страницей, а также показывают консоль проигрывателя для управления воспроизведением посетителем. Спецификация HTML5 и большинство браузеров поддерживают форматы видео **MP4**, **OGG** и **WebM**.

`<VIDEO WIDTH="700" HEIGHT="500" controls>
 <SOURCE SRC="movie.ogg" TYPE="video/ogg">
 <SOURCE SRC="movie.mp4" TYPE="video/mp4">
</VIDEO>`

Параметр **controls** добавляет к видеоролику панель инструментов для управления воспроизведением. Каждый тег `<SOURCE>` ссылается на различный тип видеофайла. Чем больше типов мы включим в тег `<VIDEO>`, тем

больше браузеров будут отображать данный видеоролик. Браузер будет проигрывать первый формат из списка, который он «понимает».

Тег <VIDEO> поддерживает еще один параметр **autoplay**, который запускает воспроизведение фильма сразу после загрузки страницы.

Добавить на веб-страницу звукозапись, хранящуюся в файле можно тегом-контейнером <AUDIO>. Поддерживаемые форматы **WAV**, **MP3** и **OGG**.

<AUDIO controls>

<SOURCE SRC="song.ogg" TYPE="audio/ogg">

<SOURCE SRC="song.mp3" TYPE="audio/mpeg">

</AUDIO>

Существует еще один способ размещения звука. Он заключается в использовании звука в качестве фона (работает только в Internet Explorer и Opera). Тег <BG SOUND> следует размещать в заголовочном разделе (<HEAD>). Данный тег не требует закрывающего тега.

<BG SOUND SRC = "sound.wav" LOOP = infinite>

Необязательный параметр LOOP задает число повторений (infinite – бесконечность). Параметр VOLUME задает громкость звучания (значения находятся в интервале от –10 000 до 0). Параметр BALANCE задает баланс между правым и левым динамиком (значения лежат от –10 000 до 10 000).

Социальные медиасервисы

Во Всемирной паутине существует множество мест, куда выкладывают материал люди, желающие сделать его всеобщим достоянием, то есть они делятся своими цифровыми накоплениями с другими людьми. Медиасервисы, хранящие выложенный посетителями материал, предоставляют возможность, не копируя материал, встраивать его в свой сайт. Просмотр встроенного ресурса осуществляется, таким образом, как бы в режиме трансляции. Таким образом, интегрированный ресурс называют виджетом.

Виджет (от англ. widget – штукавина, штучка) – любое полезное приспособление. В информационных технологиях виджет – небольшое приложение (миниприложение) для решения отдельных задач или быстрого получения информации из Интернета, интегрированное в какой-либо документ (например веб-страницу) или на рабочий стол операционной системы. Внешне виджеты представляют собой небольшие формы-окна, предоставляющие информацию, например, прогноз погоды или курс валют, видеоролик или диаграмма.

Социальные сети Всемирной паутины, предоставляющие возможность встраивания виджета, хранят огромное количество цифровых материалов и поэтому имеют собственную поисковую строку (и соответствующую систему поиска), через которую посетитель может по ключевым словам найти интересующий его материал. Как правило, отбирается несколько или несколько десятков объектов и посетитель сам решает, какой материал из найденного удовлетворяет его запросам.

Для генерации гиперссылки англоязычные сервисы, как правило, используют опцию SHARE→LINK, а для генерации html-кода вставки виджета SHARE→EMBED.

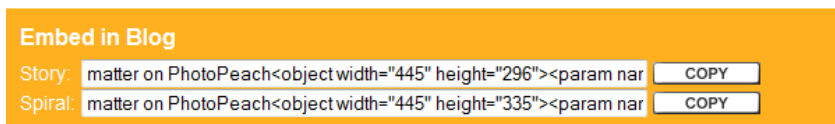
Слайд-шоу или презентации



Англоязычный сервис **PhotoPeach** [<http://photopeach.com/>] позволяет создавать из фотографий музыкальное слайд-шоу за секунды и поделиться им с другими.

Чтобы добавить просматриваемое слайд-шоу на сайт или блог, сначала необходимо щелкнуть по графической ссылке с изображением карандаша.

Над роликом появятся два текстовых поля: Story (традиционное музыкальное слайд-шоу) и Spiral (спиралевидное музыкальное слайд-шоу). Нажав кнопку Copy, сервис скопирует код соответствующего текстового поля в буфер обмена, после чего его можно вставить в свой веб-ресурс и сразу же посмотреть.



Slideshare [<http://www.slideshare.net/>] (с англ. – поделись слайдами) – служба, предназначенная для публикации презентаций, разработанных, например, в приложении PowerPoint, и просмотра их посетителями данного интернет-ресурса. Сервис Slideshare открылся в 2006 году. С его помощью появилась возможность быстрого обмена и распространения разработанных презентаций. Слайд-хостинг преобразует загруженные презентации в формат Flash, что позволяет просматривать видеоклипы в режиме реального времени по мере скачивания, уменьшить размер исходного файла и не запускать программу разработки презентаций.

После загрузки и конвертации появится опция, по которой можно скопировать html-код и добавить его либо в свой блог, либо на свою веб-страницу или любой другой разрабатываемый веб-документ с тем, чтобы внедрить в него презентацию вместе с пультом управления.

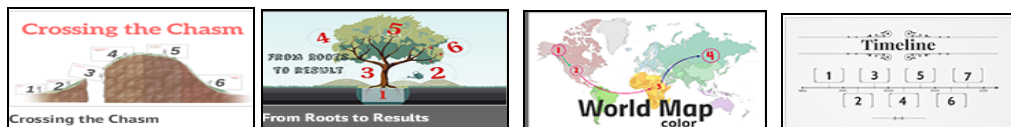
Похожими сервисами для создания и обмена презентациями являются русскоязычный MyShared [<http://www.myshared.ru/>] и англоязычный Slideboom [<http://www.slideboom.com/>], Prezi [<http://prezi.com/>], документы Google с частично русским интерфейсом [<http://docs.google.com/>]. Prezi имеет собственную программу создания презентаций с интересными готовыми шаблонами и эффектами, отличными от PowerPoint.



Prezi

Сервис предлагает собственный инструментарий, отличный от известного PowerPoint компании Microsoft, поэтому созданные в Prezi [<http://prezi.com/>] работы получаются оригинальными и необычными по представлению информации. Технология создания и показа презентаций основана на Flash-технологии. Презентации создаются и сохраняются на сервере, но при этом существует возможность их локализации.

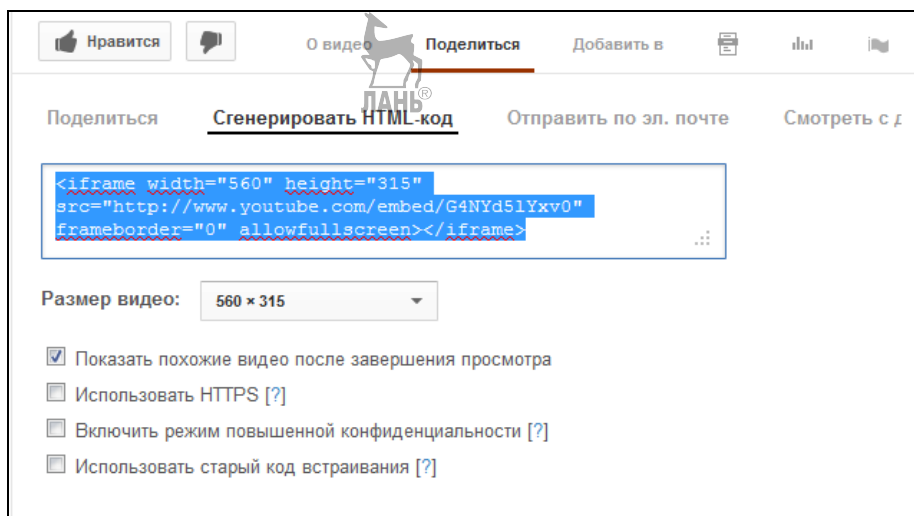
После регистрации на сервисе выбираем ссылку New Prezi для создания новой презентации. Далее появится набор красивых шаблонов (template) презентаций. В отличие от презентаций PowerPoint, презентация Prezi представляет собой единое полотно с графическим фоном, на котором в виде какого-либо достаточно сложного изображения (дерево, Вселенная, айсберг и т. д.) размещается направленная последовательность слайдов. При разработке презентации и во время ее демонстрации можно всегда увидеть не только отдельный слайд, а целую совокупность кадров или всю презентацию одновременно, что позволяет больше акцентировать внимание на взаимосвязи отдельных частей целого.



Как и другие социальные медиасети, Prezi позволяет через кнопку Share скопировать гиперссылку или html-код для внедрения презентации в блог или веб-сайт, а также позволяет сохранить ее на флэшке в виде PDF-файла или программы показа презентации в автономном режиме, то есть на локальном компьютере даже без подключения к Интернету.

Видеоролики

YouTube (с англ. – твоя труба/камера) [<http://www.youtube.com/>] – сервис, предоставляющий возможность бесплатного размещения и просмотра любительского видео. Другими словами, это хранилище нелицензионных видеоматериалов. Проект запущен в начале 2005 года Стивеном Ченом и Чедом Хёрли. Служба использует технологию Flash Video (flv), позволяющую получить хорошее качество записи при небольшом объеме передаваемых данных. Аналогичным сервисом являются Vimeo [<http://vimeo.com/>] или DailyMotion [<http://www.dailymotion.com/>].





Issuu [<http://issuu.com/>] – англоязычный социальный сервис для создания (цифровых книг, брошюр, журналов, газет и т.д.) и обмена публикациями – электронными книгами. После загрузки на сервер документов происходит их конвертация в электронную книгу (для просмотра требуется плагин Flash-плеер). После конвертации, так же как и YouTube и SlideShare, сервис предоставляет html-код для внедрения в блог или любой другой веб-сайт. Issuu имеет достаточно удобный просмотр. Книгу в буквальном смысле можно «пролистать». Существует удобное цифровое увеличение и перемещение страницы по экрану. Бесплатные ресурсы сопровождаются рекламой. Похожим сервисом является Yumpu [<https://www.yumpu.com/>].

В сервисах Storybird [<http://storybird.com/>] и StoryJumper [<http://www.storyjumper.com/>] можно создавать книгу непосредственно на веб-сервере и там же хранить ее.

StoryJumper – удобная площадка, которая позволяет иллюстрировать рассказы, применяя набор интуитивно понятных встроенных инструментов. Реализована возможность использовать не только имеющиеся у автора фотографии, но и графическую коллекцию из базы данных StoryJumper.

Сервис бесплатен для создания и чтения книг в режиме онлайн. Книгами можно бесплатно делиться и встраивать в свой блог или веб-сайт. Кроме того, за небольшую плату можно купить свою или книги других авторов в цифровом или печатном виде. Права копирования оставляет за собой компания StoryJumper.

Книга имеет альбомную ориентацию страниц, выделяются страницы обложки и форзаца. На переднем форзаце проставлены права копирования, которые изменить нельзя. Страницы можно добавлять или удалять. Как правило, на развороте сделанных книг можно увидеть с одной стороны картинку, с другой – текстовую информацию. Профессиональные дизайнеры настроили параметры текстовой информации для наилучшего восприятия читателями, поэтому разработчику книги остается лишь небольшое поле для изменения шрифта, его размера, цвета, начертания и выравнивания. Существует правило: текст не должен превышать размер блока, иначе читатель его не увидит. Поэтому сервис всегда в таком случае выдает предупреждение о том, что нужно сократить текст. Для красоты текст можно обрамить стильными рамками из коллекции данного веб-приложения.

В коллекции сервиса хранится довольно большое число графических изображений. Они разбиты на три категории: Картинки (Props), Фоновые рисунки (Scenes) и Фотографии (Photos). В каждой категории есть поисковая строка для быстрого поиска и расширенных возможностей. Если картинка не найдена, то сервис позволяет загрузить свою фотографию.

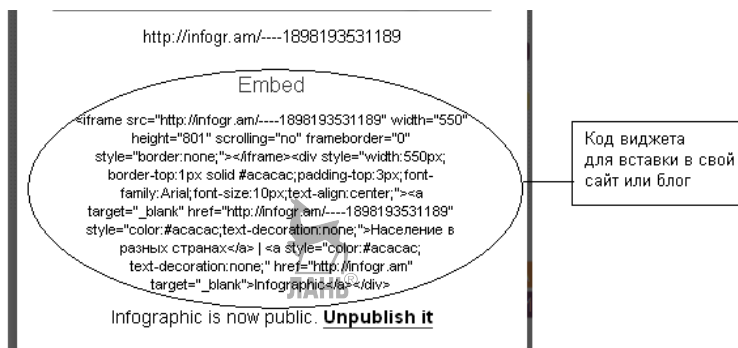
Работать над книгой можно даже сообща. Через кнопку Invite высылаются приглашения для совместной работы. Готовая книга получается очень красивой и вызывает восхищение само динамическое перелистывание страниц.

Диаграммы



Интерактивная инфографика [<https://infogram.com/>] – социальная сеть для создания и хранения диаграмм. В отличие от диаграмм, созданных в электронных таблицах, инфографика обладает свойством графического динамизма и интерактивности. Если подвести курсор мыши к какой-либо части диаграммы, то она начнет реагировать.

Сервис поддерживает следующие типы диаграмм: столбиковая диаграмма, круговая диаграмма, график, диаграмма с областями, точечная, пузырьковая, табличная. Среди необычных типов диаграмм присутствуют графическая, облако слов, карта дерева и измерительный прибор.



В приведенном выше примере видно, что документ опубликован как веб-страница по адресу <https://infogram.com/----1898193531189/>.

Мы рассмотрели создание простой диаграммы (второй пункт главного графического меню). Если выбрать первый пункт – создание инфографики, то в документ, содержащий диаграмму, можно добавлять дополнительную информацию.

Каждое добавление появляется в виде отдельного блока, который можно перемещать по документу или удалить.

Облака слов

WordArt [<https://wordart.com/>] – веб-сервис, позволяющий создавать облако слов из текста пользователя или текста размещенной в Интернете веб-странички. Облако может иметь различную форму и цветовое решение. Каждое слово облака представляет собой интерактивную гиперссылку для поиска в Google или на любую другую веб-страницу, включая дочерние страницы собственного веб-сайта. Размер слова в облаке зависит от числа его повторений в исходном тексте.

После создания аккаунта для создания облака выбираем ссылку **Create New Cloud** и в появившейся форме (Рисунок 2) совершаем нужные настройки. В текстовое поле **Cloud name** вписываем имя создаваемого облака. Текст вводится через вкладку **Words** и опцию **import words** или просто вписывается в текущую таблицу слов. Если слова для облака берем с веб-страницы, то

вводим ее URL-адрес в соответствующее поле. Если облако строится не для веб-страницы, то просто вводим слова в текстовую область. В любом случае слова расположатся в исходной таблице, где в соседнем столбце рядом со словом будет стоять число его повторений в тексте. Этот счетчик можно «подкручивать» в любую сторону, настраивая тем самым визуальную важность терминов.

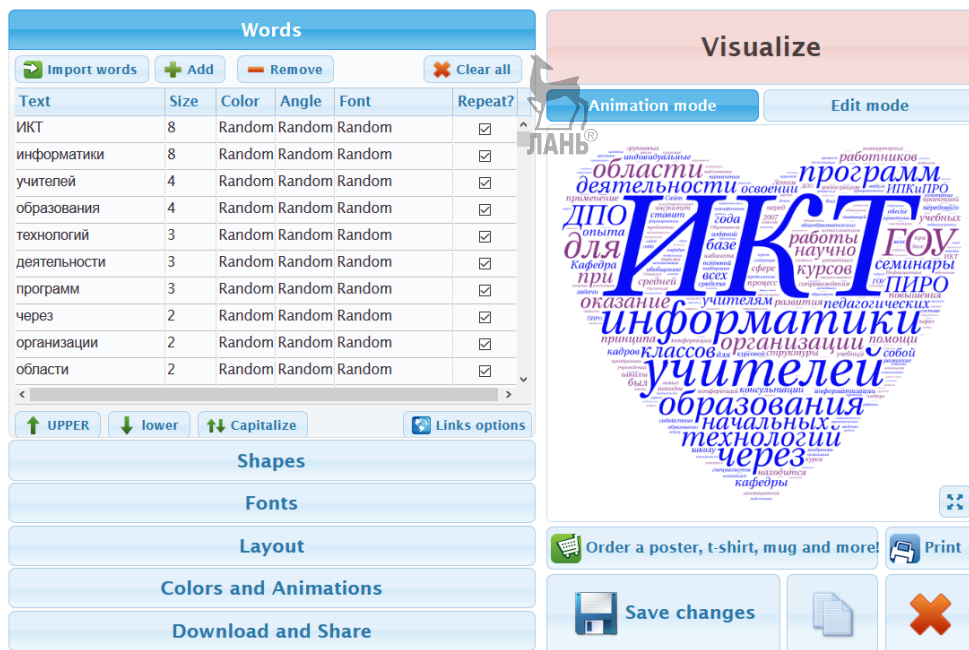


Рисунок 2. Панель инструментов сервиса WordArt

После каждого действия по заполнению формы можно щелкать по кнопке Visualize!, чтобы в правой части формы увидеть текущий результат, то есть облако слов. На вкладке Words формы находится уже упомянутая выше таблица, где сервис отображает количество вхождений каждого слова в исходный текст и где есть инструменты для коррекции всех параметров таблицы. В нижней части формы есть вкладка Fonts, которая раскроет набор доступных шрифтов. По умолчанию в сервисе установлен нерусифицированный шрифт, поэтому необходимо сменить его на, например, Neuristica. По умолчанию сервис построит из слов графическую форму облака, но в разделе Shapes имеется возможность изменить форму на другую, в том числе пользовательскую. Рисунок пользователя при загрузке на сервис можно настроить несколькими инструментами Tagul, то есть произвести его коррекцию с целью более стильного преобразования в облако слов. Сервис заполнит все отличные от белого цветовые области рисунка словами различного размера (Рисунок 3).

Важно заметить, что для организации корректной работы ссылок на поисковую систему Google, необходимо заменить исходную строку в поле Tags link pattern на [https://www.google.ru/#newwindow=1&q=\\$tag](https://www.google.ru/#newwindow=1&q=$tag) (Рисунок 4).

Сервис позволяет назначить каждому слову в облаке индивидуальную гиперссылку. Для этого нужно поставить галку в опции Show URL column in Words table, чтобы в таблице слов появился дополнительный столбец с названием URL.



Рисунок 3. Примеры облаков слов

Рисунок 4. Форма настройки гиперссылки

В разделе Download and share находятся ссылка на облако, которую можно скопировать и вставить в блог или на сайт, а также код виджета для внедрения облака на веб-страницу. Здесь же можно просто скопировать облако в виде рисунка (графического файла) на флешку, учитывая при этом, что оно потеряет свою интерактивность. И наконец, чтобы не потерять свое творение, используем кнопку Save changes для сохранения результата работы на веб-сервисе WordArt.

При внедрении облака на веб-страницу важно подобрать фон рисунка в разделе Colors and Animations, но обычно веб-дизайнеры оставляют фон рисунка прозрачным. Для этого в этом разделе нужно отметить опцию Transparent. В этом же разделе находятся инструменты по корректировке цветовой схемы облака.

Сервисом с похожим набором инструментов и возможностей является Tagxedo [<http://www.tagxedo.com/>].

Wordle

Еще одним веб-сервисом по автоматическому созданию облака слов является Wordle [<http://www.wordle.net/>]. Отличием от аналогичных сервисов

является использование технологии Java вместо Flash. Поэтому для просмотра и создания облаков необходимо иметь на компьютере установленную платформу языка Java и соответствующий плагин в браузере.

Слова облака не являются интерактивными, но есть возможности сохранения облака в виде графического рисунка невысокого разрешения в формате PNG на локальный компьютер и печати картинки на принтере. Сервис не требует регистрации и авторизации.

Чтобы создать свое облако, выбираем ссылку Create. Откроется текстовая область, куда необходимо ввести или скопировать текст, из слов которого будет состоять облако. Далее нажимаем кнопку Go, и графическое воплощение текста будет сгенерировано веб-приложением Wordle в текущем окне, где будут кнопки и меню в качестве инструментов настройки полученного произведения. В левом нижнем углу появится кнопка Open in Window – Открыть в новом окне (Рисунок 5).

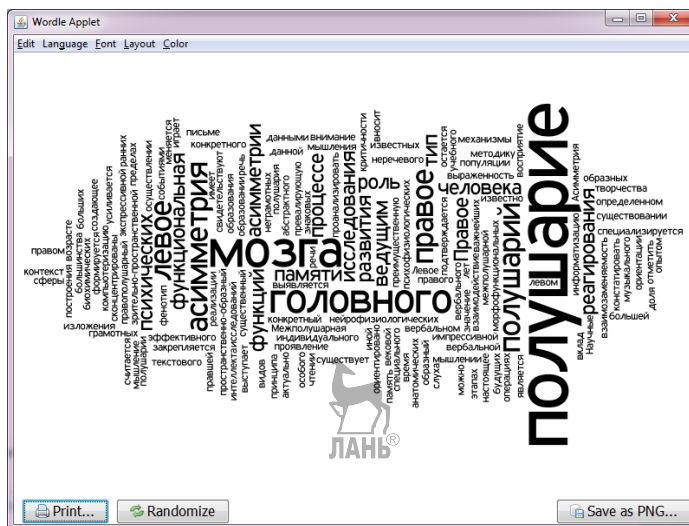


Рисунок 5. Пример облака слов в сервисе Wordle

Набор инструментов довольно простой: палитра цветов, шрифты, положение слов, можно задать общее количество слов и просмотреть число повторений каждого слова текста, удалить распространенные слова русского и других национальных языков, перевести все слова в верхний или нижний регистр, отменить или вернуть последнее действие над облаком и так далее.

Большим недостатком сервиса является невозможность сохранения облака на сервисе и, как следствие, публичного обмена им.

AnswerGarden

Уникальным среди сервисов по созданию облака слов является AnswerGarden [<https://answergarden.ch/>] – Сад Ответов. Этот сервис формирует облако из ответов посетителей на поставленный разработчиком вопрос. Посетитель Сада может либо выбрать (присоединиться) к существующему в Саду ответу,

тем самым увеличивая его рост, либо добавить новый ответ и начать его «выращивать» (Рисунок 6).

Сад Ответов предназначен для использования в образовании как минималистичный набор инструментов для организации быстрой обратной связи или как инструмент для креативного мозгового штурма, для определения уровня знаний класса по определенной теме, на конференциях и семинарах, или чтобы растопить лед в аудитории в веселой и интерактивной форме.

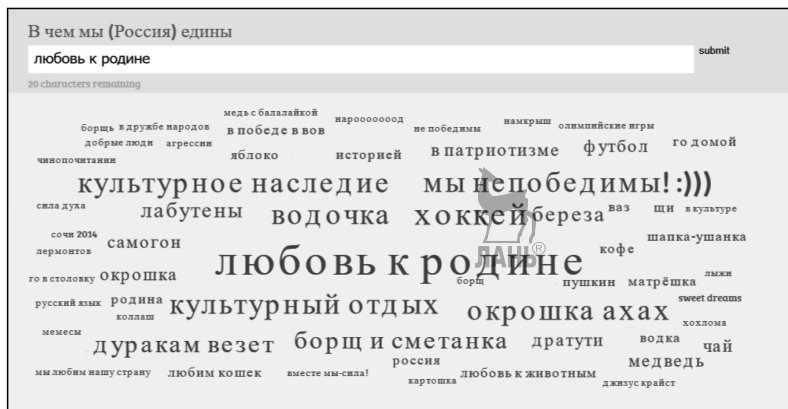


Рисунок 6. Пример облака слов в сервисе AnswerGarden (<https://answergarden.ch/share/308478>)

С помощью данного сервиса нельзя организовать тест из нескольких вопросов разного типа, но можно организовать быстрый опрос. Вопрос необходимо сформулировать таким образом, чтобы ответы были короткими и респондент мог бы ответить многократно, то есть дать несколько разных ответов.

Сервис поддерживает четыре различных режима. В режиме Classroom (класс по умолчанию) респонденты могут дать неограниченное количество ответов, но представить каждый ответ только один раз. В режиме Brainstorm (мозговой штурм) респонденты могут дать неограниченное количество ответов, в том числе неограниченное количество копий одного и того же ответа. Этот режим полезен, если вы доверяете своей аудитории. В режиме Locked (блокировка) AnswerGarden закрыт, поэтому никакие новые ответы не принимаются. Для разблокировки нужно просто открыть его (установить его в другой режим). В режиме Moderator (модератор) каждый ответ перенаправляется в AntiGarden (теневую сторону вашего AnswerGarden). С помощью панели администратора можно отбирать только те ответы, которые вы найдете соответствующими. При создании нового AnswerGarden лучше оставить режим по умолчанию Class Mode и задать пароль администратора.

При создании опроса необходимо зарегистрировать пароль администратора, чтобы в дальнейшем можно было удалять нежелательные введенные респондентами слова. Удаленные слова помещаются системой в АнтиСад (Anti-Garden) и будут автоматически блокированы при их повторном вводе. Если же слово попало в АнтиСад случайно, то его можно вернуть обратно в Сад. Командой Wipe! можно полностью очистить Сад и АнтиСад от слов.

Сервис для публичного доступа генерирует не только гиперссылку и html-код, но и QR-код (Рисунок 7).



Рисунок 7.

QR-код для публичного доступа к облаку слов AnswerGarden

Замечательной опцией сервиса из группы Share (поделиться) и Export (экспорт) является возможность трансфера облака в другие аналогичные сервисы, а именно в Taxedo и Wordle, и социальные сети, такие как Twitter, Facebook, «ВКонтакте», «Одноклассники» и т. д. Сервис позволяет ко всему прочему добавить в другие службы вместо ссылки и самого облака так называемый микроСад (Рисунок 8).

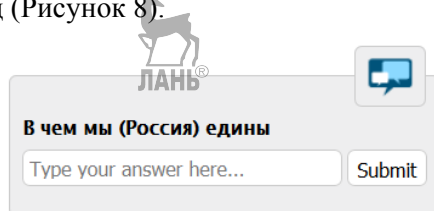


Рисунок 8.

Компактная форма для заполнения облака слов

microGarden (микроСад) – это компактная форма с текстовым полем для проведения опроса, которую можно разместить на своем сайте или блоге. В этой форме не показывается облако ответов. Но ответы помещаются в Сад ответов. Одним из удобств компактной формы является возможность на одной странице разместить несколько вопросов.

Word It Out

Word It Out [<https://worditout.com/>] — еще один сервис для создания облака слов. Он полностью бесплатный и не требует регистрации и авторизации. Сервис позволяет создать облако не только с копируемого текста или ссылки, но и из таблицы. Разработчик сам решает, как фильтровать этот текст, какие слова отобразить, какие удалить, также можно вручную настроить важность слов. Еще можно выбирать, как будет в генерируемом облаке представлен текст: с автоматической обработкой (Normal text), где наиболее часто встречающиеся слова выделяются цветом и размером шрифта, накладывается фильтр на распространенные слова; либо текст представляется в том виде, как был введен (Unprocessed), без обработки. Цвет фона, текста и размер слов, фильтрация легко настраиваются самим разработчиком в отличие от Wordle.

Готовое облако слов можно сохранить на сервисе в публичном или частном режиме. Для этого сервис запросит e-mail, на который вышлет письмо со ссылкой на облако и инструменты по его редактированию. Возможно созда-

вать новые облака и сохранять их через тот же e-mail. Это похоже на аккаунт, только без регистрации.

Облака файлов

Dropbox [<https://www.drobox.com/>] – русскоязычный сервис для хранения пользовательских файлов в Интернете, обмена файлами и совместной работы с файлами. Ссылки хранящихся в архиве Dropbox файлов можно отправить друзьям и коллегам, разместить на форумах, блогах или сайтах. Сервис доступен после регистрации. Для работы с ним требуется установка бесплатной программы.

Аналогичными сервисами являются **OneDrive** от компании Microsoft [<https://onedrive.com>] и **Yandex.Диск**. Сервисы позволяют хранить до 15 Гб информации в упорядоченном виде. Для изображений предусмотрен предпросмотр в виде эскизов, а также возможность просмотра их в виде слайдов. OneDrive позволяет внедрять виджет папки с файлами для общего просмотра в блог или сайт.



Звукозаписи

AudioPal (звук). С помощью сервиса AudioPal [<http://www.audiopal.com/>] можно бесплатно, указав только адрес своей электронной почты, создать звуковой файл, который и будет выслан на указанный адрес.

AudioPal поддерживает следующие способы создания звукозаписи:

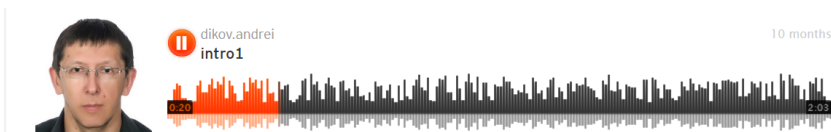
- через телефонный звонок;
- через микрофон;
- напечатать текст, который система преобразует в звук (есть возможность выбора языка и голоса для прочтения текста);
- загрузить готовый аудиофайл в формате mp3.

Всего за 3 шага можно подготовить и отправить на свой емейл информацию о созданном звуковом файле. В высланном системой письме будет содержаться ссылка на сгенерированный виджет в следующем тексте: Click Here to pickup your AudioPal code (Щелкни здесь, чтобы взять код). В открывшейся веб-странице можно увидеть изображение виджета и прослушать через него звукозапись. Красная кнопка Copy положит в буфер обмена код для встраивания виджета в блог или веб-страницу.



Похожим сервисом является **SoundCloud** [<http://soundcloud.com>]. Сервис открылся в 2008 году. На сайт бесплатно можно загрузить записи совокупной продолжительностью 120 минут. Размер файла совершенно не важен, главное – учитывать время его звучания. Выкладывать можно не только музыку, а любой звуковой файл, детские записи, речь президента, лекции и многое другое. Для загрузки или записи аудио нужно перейти в раздел Upload&Share. Появятся две большие кнопки для загрузки или непосредственной звукозаписи через браузер.

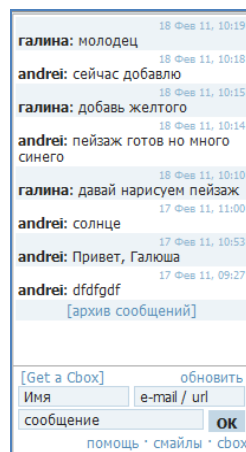
На SoundCloud все звуковые дорожки изображены графически в виде шкалы, что дает возможность пользователям комментировать конкретные части трека. К шкале слева можно добавить фотографию или рисунок.



Сервис сгенерирует код виджета для размещения звукозаписи на любом блоге или веб-сайте.

Чат

Cbox (chat box) [<http://cbox.ws/>] – чат для социальной сети, имеющий ряд возможностей: историю сообщений, хорошо настраивается под сайт, имеет элементы управления чатом. Бесплатная версия имеет меньше возможностей по сравнению с платной. Для получения кода виджета необходимо зарегистрироваться. Для просмотра новых сообщений необходимо щелкать по ссылке «Обновить».



Все приведенные ссылки на ресурсы Интернета находятся на веб-странице <http://dikandr.ru/resources/webtwo.htm>.

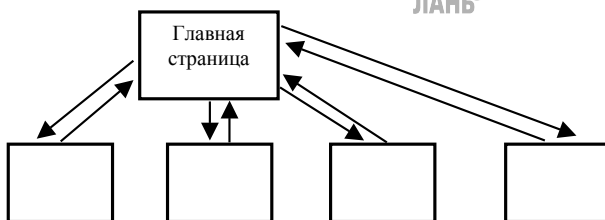
Вопросы и упражнения

1. Что такое виджет?
2. Какие существуют способы размещения мультимедиа на веб-страницу?
3. Какие интернет-сервисы предназначены для обмена презентациями?
4. С помощью какого сервиса можно разместить на веб-странице опрос?
5. В каком сервисе при просмотре слайдов используется эффект приближения и удаления?
6. Какой сервис может генерировать облака текстовых гиперссылок?

СТРУКТУРА САЙТА

В тот момент, когда информации на одной странице становится настолько много, что становится трудно ее находить, самое время подумать о том, как разбить одну страницу на несколько связанных гиперссылками страниц, называемых узлом (сайтом). Существуют различные схемы взаимосвязи группы страниц. В книге Э. Шафрана приведены следующие схемы:

Стандартная структура узла

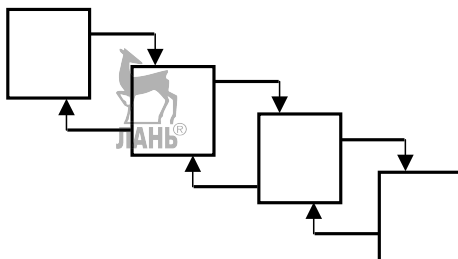


Стандартная структура характеризуется тем, что навигацию по сайту можно осуществлять с главной страницы на любую дочернюю и обратно, но нельзя перейти с одной дочерней на другую, не посетив главную. Каждая страница должна быть сохранена в файле формата html. Главная страница должна иметь имя index.html. Начальная страница сайта (главная страница) является для посетителя входом на территорию сайта. На этой странице обычно размещают название и аннотацию веб-ресурса, информацию о разработчиках, логотипы, сведения о тех, кому принадлежит авторское право и право копирования, дату создания сайта, счетчик посещений, контактную информацию, навигационную панель и тому подобное. Главная страница занимает обычно один экран. Поэтому часто информацию на ней располагают в несколько колонок. На дочерних (внутренних) страницах можно выделить несколько структурных областей: шапку, основную область и подвал. Шапка содержит заголовок текущей страницы и, возможно, навигационную панель. Подвал также может содержать навигационную панель или ее элементы. В основной части находится информация по теме. Все страницы сайта должны быть выполнены в едином стиле (одинаковый фон, одинаковое оформление заголовков и так далее).

Навигационная панель сайта представляет собой совокупность гиперссылок на дочерние страницы первого уровня (или более) иерархической структуры узла, выполненную в текстовом (список, таблица) или графическом (изображение-карта) виде. Панель, как правило, присутствует на всех страницах сайта.

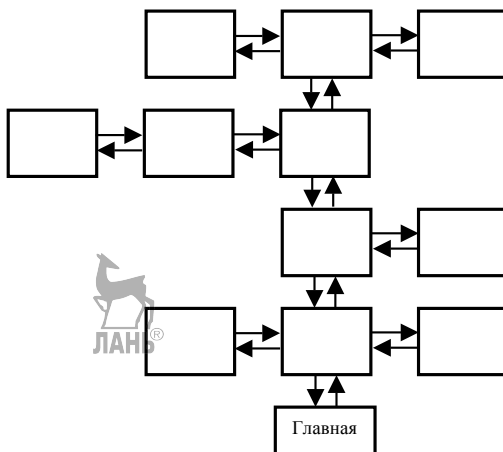
Каскад

Посетители сайта, построенного по схеме «Каскад», смогут перемещаться последовательно шаг за шагом в одном направлении, как вода в каскаде водопадов. С текущей страницы можно попасть на следующую или предыдущую, но нельзя перепрыгнуть через несколько страниц сразу. Такую схему трудно назвать демократической.



Небоскреб

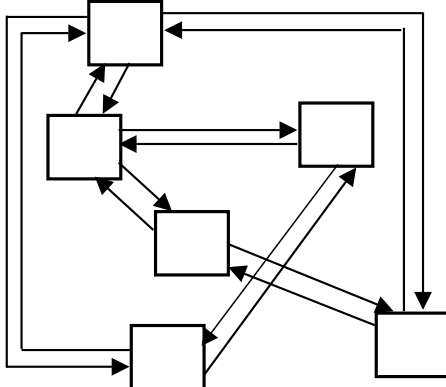
Структура «Небоскреб» является многоуровневой (многоэтажной). На каждом уровне (этаже) находится страница (комната), с которой можно перейти по гиперссылке на другой уровень. Такая структура расположения материала удобна учителю, когда он, например, группирует учебный материал по уровню сложности или степени детализации. Неудобство заключается в навигации. Посетителю трудно сразу сориентироваться, в каком месте сайта он находится в данный момент и как перейти на другой этаж.



Паутина

Структура «Паутина» характеризуется хаотичностью гиперсвязей. Все страницы сайта соединяются без какой-либо логической закономерности. Такие узлы часто превращаются в подобие самой паутины. В них трудно ориентироваться.

Для лучшего ориентирования на просторах веб-сайта разработчики строят *карту сайта*. Она представляет собой схему расположения и взаимосвязей страниц, выполненную в текстовом или графическом варианте. Каждая страница представлена гиперссылкой, что позволяет посетителю со схемы сразу перейти на выбранную страницу сайта.



Вопросы и упражнения

1. На какие типы делятся веб-страницы сайта?
2. Какое имя файла должна иметь главная страница сайта?
3. Что такое навигационная панель сайта?
4. Что такое карта сайта?
5. Чем отличается структура «Каскад» от всех остальных?
6. Чем отличается структура «Небоскреб» от всех остальных?

ФОРМЫ

До сих пор все средства языка HTML позволяли создавать страницы, где посетитель мог взаимодействовать с ними только щелкая мышкой на гиперссылках. С помощью форм возможности взаимодействия расширяются. Форма похожа на бланк документа, в котором заполняют отдельные области, или на анкету, где письменно отвечают на вопросы.

Создается форма контейнером `<FORM>`, внутри которого размещаются интерактивные элементы формы. Каждый элемент формы (и сам тег `<FORM>`) может содержать атрибут `NAME`, присваивающий элементу формы уникальное имя. Через это имя в скрипте, обрабатывающем форму, можно обращаться к конкретному элементу.

Тег `<FORM>` может содержать два параметра:

ACTION	задает адрес ресурса Интернета, который примет для обработки данные, введенные в форму. Если параметр опущен, то данные отправляются на ту же страницу, где находится сама форма (обязательный атрибут)
METHOD	задает способ отправки данных. Метод <code>POST</code> посылает данные отдельно от <code>URL</code> , способ <code>GET</code> присоединяет данные к адресу

`<FORM NAME="firstForm" METHOD="POST" ACTION = "">`

Тег `<BUTTON>` `</BUTTON>` позволяет создавать кнопки на веб-странице практически с любым содержанием. Элемент является строчным, поэтому требуется заключать его в блочный контейнер.

`<P>`

`<BUTTON>`

``

`
` Начальная страница

`</BUTTON>`

`</P>`

Тег <INPUT> с параметром TYPE создает различные элементы для ввода информации посетителем. Все элементы <INPUT>, так же как и <BUTTON>, являются строчными и должны заключаться в блочный контейнер, чтобы соответствовать последней спецификации HTML.

TYPE=TEXT создает поле для ввода или вывода текста. Для определения максимального количества символов в текстовом поле есть параметр MAXLENGTH=число. Для отображения максимального количества символов параметр SIZE=число. Для задания начального значения VALUE=текст. Чтобы использовать текстовое поле только для вывода информации, необходимо добавить параметр *readonly*. Например, `<p><input name="field1" type="text"></p>`.

Сюда вводится текст

TYPE=PASSWORD создает текстовое поле, где все вводимые символы отображаются символом *.

TYPE=CHECKBOX создает поле для установки флажка. Параметр VALUE=строка определяет значение, которое будет передано серверу, если эта кнопка будет выбрана посетителем. Параметр CHECKED делает поле выбранным по умолчанию, то есть ставит в него флажок. Чекбоксы (флажки) используют для предоставления выбора нескольких вариантов из многих.

Пр и м е р.

К тригонометрическим функциям относятся

- | | |
|---|---|
| <input type="checkbox"/> парабола | <input checked="" type="checkbox"/> косинусоида |
| <input type="checkbox"/> гипербола | <input type="checkbox"/> гипотенуза |
| <input checked="" type="checkbox"/> синусоида | <input checked="" type="checkbox"/> тангенсоида |

```
<P> К тригонометрическим функциям относятся <BR>
<INPUT NAME="C2" TYPE="CHECKBOX" > парабола
<INPUT NAME="C2" TYPE="CHECKBOX" > косинусоида<BR>
<INPUT NAME="C2" TYPE="CHECKBOX" > гипербола
<INPUT NAME="C2" TYPE="CHECKBOX" > гипотенуза<BR>
<INPUT NAME="C2" TYPE="CHECKBOX" > синусоида
<INPUT NAME="C2" TYPE="CHECKBOX" > тангенсоида<BR>
</P>
```

TYPE=RADIO создает очередной переключатель в группе переключателей. Чтобы создать группу элементов, необходимо использовать параметр NAME с одинаковым значением для каждого элемента группы. Здесь так же используются параметры VALUE и CHECKED. Например, варианты ответа в тесте могут быть оформлены так:

☒ скорее нравится, чем не нравится

☐ определенно нравится

☐ очень нравится



<P>

<INPUT TYPE = RADIO NAME = "a1" CHECKED>

скорее нравится, чем не нравится

<INPUT TYPE = RADIO NAME = "a1">

определенно нравится:

<INPUT TYPE = RADIO NAME = "a1">

очень нравится:
 </P>

TYPE=RESET создает кнопку сброса всех последних изменений, восстанавливая значения по умолчанию. Значение VALUE дает название кнопке.

Очистить

<P> <INPUT TYPE = "RESET"
VALUE =
"Очистить"> </P>

TYPE=SUBMIT создает кнопку отправки введенных в форму данных для обработки их скриптом, расположенном на сервере по адресу, указанному параметром ACTION тега <FORM>. Способ отправки задает параметр METHOD тега <FORM>.

Отправить

<P> <INPUT TYPE =
"SUBMIT" VALUE =
"Отправить"> </P>

TYPE=BUTTON создает кнопку, щелчок по которой может запускать функцию обработки данного события, написанную на каком-либо скриптовом языке (например JavaScript).

Проверить

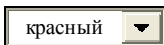
<P> <INPUT TYPE = BUTTON
VALUE = "Проверить"> </P>

TYPE=FILE создает кнопку с текстовым полем для вызова стандартного окна операционной системы «Выбор файла для загрузки». Имя выбранного файла запоминается в свойстве VALUE и может быть только прочитано с помощью скриптового языка, но не может быть изменено. В зависимости от браузера, в свойство value записывается не только имя файла, но и путь к нему.

<P> <INPUT TYPE = "FILE" NAME = "загрузить_изо"> </P>



TYPE=HIDDEN создает невидимый элемент, предназначенный, как правило, для отправки дополнительной информации обрабатываемому скрипту. Тег <SELECT> </SELECT> позволяет размещать на странице свернутый список элементов. Очередной элемент добавляется тегом <OPTION>. Параметр SELECTED делает элемент выбранным по умолчанию. В скрипте этот параметр возвращает логическое значение, зависящее от сделанного посетителем выбора. Параметр VALUE задает передающееся скрипту для обработки значение выбранного элемента списка.



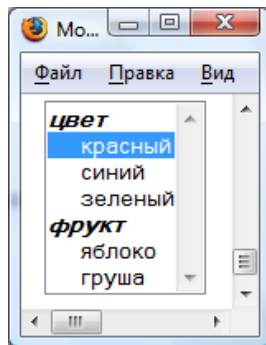
```
<P>
<SELECT NAME = "list1" SIZE = 1>
  <OPTION SELECTED VALUE = 1> Красный
  <OPTION VALUE = 2> Синий
  <OPTION VALUE = 3> Зеленый
</SELECT>
</P>
```

У тега <SELECT> имеются необязательные параметры.

NAME	строка	определяет имя списка
SIZE	1	поле со списком
	больше 1	развернутый список
MULTIPLE		одновременный выбор нескольких элементов

С помощью тега OPTGROUP и параметра LABEL можно объединять в группы пункты списка. Например:

```
<P> <SELECT SIZE = 5>
  <OPTGROUP LABEL = цвет>
    <OPTION> красный
    <OPTION> синий
    <OPTION> зеленый
  </OPTGROUP>
  <OPTGROUP LABEL = фрукт>
    <OPTION> яблоко
    <OPTION> груша
  </OPTGROUP>
</SELECT> </P>
```



`<TEXTAREA>` `</TEXTAREA>` создает поле для многострочного текста.

Здесь пишется
объемная
текстовая
информация

Параметры:

NAME	текст	определяет имя текстовой области
ROWS	число	определяет число строк в области
COLS	число	определяет число столбцов в области

Блочный контейнер `<FIELDSET>` `</FIELDSET>` создает прямоугольную рамку вокруг элементов формы, группируя их таким образом. Для вывода названия рамки существует тег `<LEGEND>` `</LEGEND>`.

Легенда


Однако чтобы иметь практическую пользу от применения элементов формы, необходимо знать язык JavaScript, на котором можно написать процедуру обработки введенной информации и выдать ответ посетителю страницы.

Дополнительные типы полей для ввода данных

В HTML5 были введены новые типы полей. Они выступают помощниками ввода и проверки введенных значений без написания JavaScript-кода. Среди новых элементов есть, например, указатель ресурса `<url>` и электронная почта `<email>`. Они дают возможность писать семантически правильный код, облегчить заполнение формы на мобильных устройствах путем отображения специальных кнопок (вроде кнопок `@` или `.com`) в зависимости от типа ввода данных и обеспечивают валидацию введенных значений браузером.

Тип	Описание
email	ввод email-адреса
<code><input type="email" placeholder="name@host.ru"></code>	
tel	ввод телефонного номера – нет строгого синтаксиса, но разрывы строк будут удалены
url	ввод адреса ресурса Интернета URL
search	поле поиска с разрывами строк автоматически удаляется



Тип	Описание
number	число с плавающей точкой
<pre><input type="number" name="points" min="0" max="10" step="3"></pre>	
range	элемент управления слайдер для ввода числового значения из заданного диапазона
<pre><input type="range" name="slider" value="5" min="-10" max="10" step="2" onInput="txt.value=slider.value"></pre> 	
date	ввод дня, месяца и года
datetime	ввод дня, месяца, года, часа, минуты, секунды и микросекунды относительно текущей временной зоны
datetime-local	ввод даты и времени без временной зоны
month	ввод месяца и года без временной зоны
week	ввод номера недели без временной зоны
time	ввод времени без временной зоны
color	выбор цвета

Атрибуты форм и их элементов

В спецификации HTML5 появились новые атрибуты у многих элементов формы. Это связано, в первую очередь, с проблемой повышения производительности разработки интерактивных веб-страниц. На практике разработчикам приходилось писать много кода для проверки (верификации) вводимых данных. Как правило, код пишется в виде скриптов на языке программирования JavaScript, что требует процедуры отладки, тем самым существенно удлиняя процесс разработки готового продукта. Теперь многие типичные сценарии верификации на JavaScript заменены атрибутами тегов.

Атрибут	Значение	Пояснение
placeholder	текст	устанавливает образец заполнения в текстовое поле
<p> Фамилия <input type="text" placeholder="Иванов"> </p>		
autofocus		фокусирует курсор в данном поле после загрузки страницы
<input type="text" name="firstName" autofocus>		
autocomplete	on/off	заполняет поле формы на основе предыдущего введения данных
<input type="text" name="trackingCode" autocomplete="off">		
required		требует от пользователя ввести данные в поле до отправки формы
<input type="text" name="givenName" required>		
form	имя формы®	соединяет элементы input, select или textarea с формой, даже если они находятся за пределами формы
<input type="button" name="sortLH" form="sort">		
formaction, formenctype, formmethod, formtarget		заменяет соответствующие атрибуты формы: formaction определяет файл или приложение, которое будет отправлять форма; formenctype подробно описывает, как данные формы кодируются методом POST; formmethod определяет, который из методов HTTP (GET, POST, PUT, DELETE) будет применяться для отправки данных формы; formtarget определяет окно для результатов формы®
<input type="submit" value="Submit" formmethod="POST">		
disabled		отменяет проверку и отправку данных данного поля, поле ввода затеняется серым и отключается возможность с ним взаимодействовать
<input type="reset" disabled>		
pattern	регулярное выражение	определяет шаблон на основе регулярных выражений, используемый для проверки поля ввода

Атрибут	Значение	Пояснение
<code><input type="text" name="countryCode" pattern="[A-z]{3}" title="Трехбуквенный код страны"></code>		
novalidate		форма или поле ввода не проверяются при отправке формы
<code><form action="demo_form.php" novalidate="novalidate"></code>		

Автозаполнение с помощью списка данных

При помощи элемента `datalist` HTML5 дает возможность создавать список данных для автозаполнения поля ввода.

```
<input type="text" list="фрукты" >
<datalist id="фрукты">
  <option value="арбузы">
  <option value="ананасы">
  <option value="абрикосы">
  <option value="апельсины">
  <option value="бананы">
</datalist>
```

Если необходимо сопроводить текстовое поле или любой другой элемент формы подсказкой, то применяют тег `<LABEL>` с атрибутом `FOR`. Этот тег создает связь подсказки с элементом, и посетитель может щелкать мышкой не только на элементе, но и на подсказке.

Вопросы и упражнения

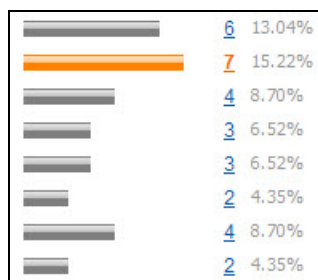
1. Какую роль играют формы на веб-страницах?
2. Почему в группе радиокнопок каждая радиокнопка должна иметь одно и то же имя?
3. Для чего может пригодиться скрытый элемент на веб-странице?
4. Какое значение нужно присвоить параметру `SIZE`, чтобы список превратился в поле со списком?
5. Если не указать явно адрес страницы для обработки данных формы, то куда передадутся и кто примет данные для обработки?
6. Какая специальная кнопка предназначена для отправки введенной в форму информации?
7. Какая специальная кнопка предназначена для сброса введенной в форму информации к первоначальному виду?
8. Почему каждый элемент формы заключается в тег `<P>`?
9. Каким тегом можно добавить в форму графическую кнопку?

Вывод числовой информации в графической форме

Выше упоминалось, как можно с помощью текстовой строки или области выводить информацию на веб-странице только для чтения. Для этого в соответствующие теги добавляется атрибут **readonly**. В HTML5 добавились два новых элемента для стильного отображения числовых данных в графическом виде – **<METER>** и **<PROGRESS>**.

Элемент **<PROGRESS>** служит для визуализации процесса выполнения какой-либо задачи. Другими словами – это прогресс-бар. Элемент **<METER>** очень похож на **progress**, но служит для графического отображения каких-либо числовых значений, не меняющихся непрерывно по времени. Для того чтобы на веб-странице после ее загрузки непрерывно менялись значения атрибутов элемента **<PROGRESS>**, необходимо писать соответствующий скрипт.

Типичным примером использования элемента **<METER>** может служить вывод результатов голосования или визуализация количества заполненной информации «О себе» на личной странице «ВКонтакте», результаты экзамена или коэффициент загрузки диска. Типичным примером использования элемента **<PROGRESS>** является процент загрузки файла либо процент завершенности работ.



Атрибуты элемента **<METER>**:

Атрибут	Значение	Пояснение
value	число	текущее значение
min	0/число	минимальное значение (по умолчанию 0)
max	1/число	максимальное значение (по умолчанию 1)
low	число	от min до low значение считается ниже нормы, полоса окрашивается в другой цвет
optimum	число	оптимальное значение
high	число	от high до max значение считается выше нормы, полоса окрашивается в другой цвет
<pre><h2>Уровень воды в чайнике:</h2> <meter min="0.4" max="2.2" low="0.7" optimum="1" high="2" value="0.54">0.54 литра</meter> <p>Долейте воды!</p></pre>		

Атрибуты элемента <PROGRESS>:

Атрибут	Значение	Пояснение
value	число	текущее значение
max	1/число	максимальное значение (по умолчанию 1)
<progress value="50" max="100">50%</progress>		

У элемента <PROGRESS> всего два атрибута: value (для определения состояния выполнения) и max (для обозначения максимально достижимого значения). Если значение max не установлено, предполагается диапазон от 0 до 1 и текущее значение value может колебаться между ними. Если значение max установлено, то value меняется от 0 до установленного в max значения.

На графической полосе цветом будет показан процент текущего значения value от максимальной величины, то есть от всей длины полоски.

Оба элемента являются контейнерами, внутри которых помещается информация для браузеров, не поддерживающих эти теги.

МЕТАТЕГИ

Метатеги предназначены для помещения в код веб-страницы информации о ней самой. Эта информация используется поисковыми системами Интернета, браузерами и редакторами веб-страниц. Она никак не отображается в окне веб-обозревателя.

Все метатеги помещаются в секции html-заголовка (контейнер <HEAD></HEAD>) веб-страницы. Большинство из них создается с помощью одинарного тега <META> и трех его атрибутов. Тип метатега задается атрибутом NAME или HTTP-EQUIV, а сами данные – атрибутом CONTENT.

Определение кодировки алфавита

Нередко, посещая веб-страницы, можно увидеть вместо привычных букв русского алфавита всякую белиберду. Это происходит оттого, что разработчик данной страницы не прописал метатег кодировки алфавита. Поэтому браузер пытается определить кодировку самостоятельно, но у него это не всегда получается правильно. Кириллица для системы Windows задается кодировкой windows-1251 или более современной utf-8.

<META charset = "utf-8">

Обновление и перенаправление на другую веб-страницу

Если на странице данные меняются очень часто (например, курс акций на торговой бирже), то загруженная на компьютер клиента веб-страница должна с коротким промежутком времени обновлять данные, непрерывно изменяемые на сервере. Следующий пример показывает определение операции обновления. Refresh в переводе с английского – обновление. Параметр content задает число секунд перед обновлением.

<META HTTP-EQUIV = "refresh" CONTENT = "100">

Следующий метатег определяет операцию перенаправления браузера с текущей страницы на ту, адрес которой задан в параметре content. Здесь же задается число секунд до выполнения этого действия.

<META HTTP-EQUIV = "refresh" CONTENT = "10; URL =
www.sura.ru/dikov">

Базовый интернет-адрес

Метатег, задающий базовый интернет-адрес, может быть полезен, если страницы сайта разбросаны по разным веб-узлам. Вместо того, чтобы в гиперссылках указывать полные интернет-адреса, можно указать относительные, а в секции заголовка страницы поместить метатег базового адреса. В этом случае полные интернет-адреса будут вычисляться относительно него.

<BASE HREF = "www.narod.ru">

Контроль кэширования

<META HTTP-EQUIV = "Pragma" CONTENT="no-cache">

Данный метатег отменяет сохранение браузером данной веб-страницы в кэше.

Тема веб-страницы

<META HTTP-EQUIV = "subject" CONTENT="Фракталы вокруг нас">

Специальные метатеги

Задав тип метатега NAME, можно ввести собственные данные. Значениями атрибута NAME могут быть слова: keywords, description, author, generator, abstract, expiration и другие. Сами же данные вводятся в область CONTENT.

<META NAME="generator" CONTENT="Microsoft FrontPage">

<META NAME="Microsoft Border" CONTENT="none, default">

<META NAME = "Author" CONTENT = "Диков Андрей">

Определение набора ключевых слов для поисковой машины

<META NAME="keywords" CONTENT="математика, Пифагор, школа">

Значение параметра CONTENT должны составлять слова, которые потенциальные посетители будут задавать поисковой машине.

Описание веб-страницы

Информация из области CONTENT с большой вероятностью будет выдана поисковой машиной как аннотация найденного веб-ресурса.

<META NAME="description" CONTENT="Страница о математической школе Пифагора">

Определение времени переиндексации для поисковой машины

<META NAME="Document-state" CONTENT="Dynamic">

<META NAME="Revizit-after" CONTENT="10 days">

В примере задан период переиндексации каждые 10 дней. Если обновление страницы не планируется, то пишем

<META NAME="Document-state" CONTENT="Static">

Определение круга посетителей

<META NAME="Classification" CONTENT="Учителя математики">

Авторские права

<META NAME="Copyright" CONTENT="ПГПУ им. В.Г. Белинского">

Управление работой поисковых машин

<META NAME="Robots" CONTENT="α">,

где α может принимать следующие значения:

index – индексировать страницу,

follow – индексировать страницы, на которые указывают ссылки,

noindex – не индексировать страницу, но отработать ссылки,

nofollow – индексировать страницу, но не отрабатывать ссылки,

none – равносильно двум предыдущим значениям вместе взятым.

Значения могут быть перечислены через запятую.

Практическая работа

Задание 1. Первая веб-страница

Создайте в текстовом редакторе «Блокнот» html-документ, являющийся вашей домашней веб-страничкой.

1. Разместите на ней заголовок-приглашение на страницу посетителей.
2. Поместите бегущую строку с какой-либо рекламой.
3. Добавьте абзац текста, рассказывающий о том, где и когда вы родились, в какой школе учились и т.п.
4. Добавьте еще один абзац текста, рассказывающего о том, где вы учитесь и по какой специальности, на каком курсе, где работаете.
5. Добавьте еще абзац текста, рассказывающий об учебном заведении, в котором вы учитесь.
6. Залейте фон страницы каким-либо цветом или «оклейте» его обоями с графическим рисунком.
7. Подберите цвет текста к выбранному фону так, чтобы он легко читался.
8. В конце страницы по центру разместите слово «ДАЛЕЕ...».

Задание 2. Вторая веб-страница

Создайте в текстовом редакторе «Блокнот» вторую веб-страницу, являющуюся продолжением первой.

1. Разместите на ней многоуровневый список своих увлечений.
2. Постройте таблицу музыкальных предпочтений.
3. Фон таблицы залейте цветом или «оклейте» обоями.

4. Добавьте несколько картинок, в том числе анимированных, на обе веб-страницы.
5. Фон страницы сделайте таким же, как и на предыдущей странице.
6. В конце каждой страницы поставьте разделительную линию, после которой расположите гиперссылки, связывающие обе страницы и дату создания ресурса.

Задание 3. Таблицы

Создайте в текстовом редакторе «Блокнот» третью веб-страницу, на которой разместите:

1. таблицу, подобную по структуре таблице «Площадь круга» с большими полями вокруг содержимого ячеек;
2. таблицу «Линия времени»;
3. таблицу размером 10 строк \times 10 столбцов с пробелом в ячейках, где выделите различным цветом фона заголовочную часть (несколько первых строк), нижнюю часть (несколько нижних строк) и основную часть (строки между верхней и нижней частями). Используйте теги структурирования;
4. таблицу размером 10 строк \times 10 столбцов с пробелом в ячейках, где организуйте столбцы в несколько групп и придайте им общие свойства (фон, граница, размер).

Площадь круга									

Линия времени

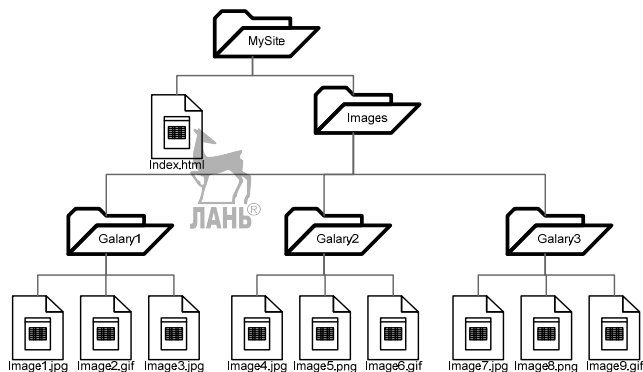
1987	1988	1989	1990	1991	1992	1993	1994	1995
------	------	------	------	------	------	------	------	------

Триколор

Задание 4. Галереи

Организируйте файловую структуру для хранения веб-сайта примерно как на приведенной схеме.

В папку Images положите изображения для фотогалерей. С целью удобного структурирования информации создайте в папке Images еще три папки с названием фотогалерей и распределите соответственно графические файлы. Не забудьте в html-коде указывать имена папок при задании значения параметра SRC.



На веб-странице разместите картинки в соответствии с темой и с различными вариантами расположения, как показано ниже. Картинки первой галереи выровнены по верхнему краю, второй галереи – расположены в шахматном порядке, третьей галереи – заключены в фотопленку. Каждую галерею сопроводите заголовком.

Галерея 1



Шахматный порядок второй галереи достигается за счет расположения картинок и текста в соответствующих ячейках таблицы. Таблицу не видно за счет отсутствия граничных линий.

Галерея 2



Крокодил
«Гена»

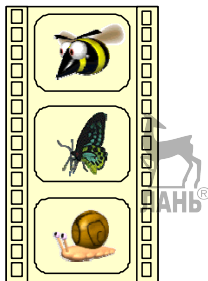
Слон
«Кеша»



Корова
«Мурка»

Фото пленка делается из таблицы, состоящей из трех столбцов и трех строк. Ячейки первого и последнего столбцов объединены (параметр ROWSPAN=3), и для них устанавливается ширина и фоновый рисунок. Фоновый рисунок – это изображение одного квадрата. Его можно создать в графическом редакторе Paint. Браузер размножит квадрат на весь столбец.

Галерея 3



Задание 5. Постановление

Скопируйте с веб-страницы <http://dikandr.ru/mybooks/stuff.htm> текстовый файл «Постановление.txt». Преобразуйте его в html-документ. Создайте в начале html-документа гипертекстовое оглавление на основные разделы документа.

Задание 6. Изображение-карта

Подберите из имеющихся рисунков изображение-карту или создайте собственное. Разместите его на веб-странице и используйте по назначению. Для разметки карты воспользуйтесь бесплатным приложением GeoHTML или онлайн-генератором карты.

Задание 7. Мультимедиа

На веб-странице разместите аудио- и видеoinформацию различными способами. Добавьте ссылки на файлы презентаций и текстовые документы. Разместите не менее трех виджетов, например: видеоролик с YouTube, звукозапись с SoundCloud и презентацию с SlideShare.

Задание 8. Canvas. Пизанская башня

Разместите графическое полотно на веб-странице и нарисуйте графическими методами Пизанскую башню из квадратов.



Задание 9. Canvas. Yota

Разместите графическое полотно на веб-странице и нарисуйте графическими методами логотип компании Yota из окружностей и дуг. Примените методы трансформации для создания копий.

Задание 10. Canvas. Галерея

Разместите графическое полотно на веб-странице. С помощью стилей сделайте у него графический фон и разместите на этом фоне несколько фотографий с эффектом поворота на небольшой градус в разные стороны.



Задание 11. Тест

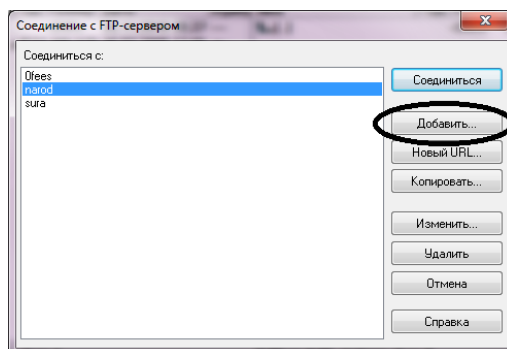
Придумайте или возьмите готовый тест и, используя интерактивные элементы формы, разработайте его на веб-странице. Скрипт для обработки теста напишите после изучения языка JavaScript.

Дополнительная литература

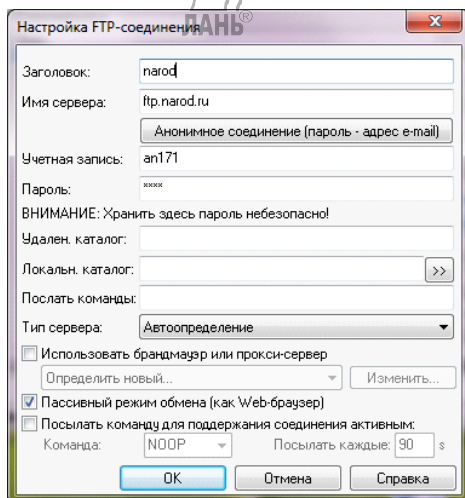
1. HTML Current Status — W3C [Electronic resource]. — Mode of access : http://www.w3.org/standards/techs/html#w3c_all
2. Mozilla Developer Network [Electronic resource]. — Mode of access : https://developer.mozilla.org/ru/docs/Обучение_canvas.
3. W3School Online Web Tutorial. HTML Tutorial [Electronic resource]. — Mode of access : <http://www.w3schools.com/html>
4. *Квинт И.* Создаем сайты с помощью HTML, XHTML и CSS. — СПб. : Питер, 2014.
5. Основы и секреты front-end разработчика [Электронный ресурс]. — Режим доступа : <http://xiper.net/>
6. Теги HTML [Электронный ресурс]. — Режим доступа : <http://htmlbook.ru/>
7. Учебники для веб-разработчиков [Электронный ресурс]. — Режим доступа : <http://www.wisdomweb.ru/HTML5/>
8. *Фримен Э.* Изучаем HTML, XHTML и CSS. — СПб. : Питер, 2014.

Приложение. Публикация сайта по ftp

В приложении Total Commander выбираем пункт меню Сеть→Соединиться с ftp-сервером. В открывшемся диалоговом окне нажимаем кнопку Добавить.



В следующем окне заполняем параметры соединения, которые при необходимости можно уточнить в справке того веб-сервера, к которому налаживаем доступ.



После соединения с сервером на панели устройств появится устройство со знаком O. Это и есть удаленный сервер.



СТИЛИЗАЦИЯ ВЕБ-СТРАНИЦ СРЕДСТВАМИ CSS

ВВЕДЕНИЕ

Каскадные таблицы стилей предназначены в первую очередь для внешнего представления содержимого веб-страницы или всего сайта. Они пришли на смену параметрам html-тегов и оформились как самостоятельная технология, способная работать не только с HTML, но и с другими языками разметки.

Каскадные таблицы стилей позволяют задать уже известным тегам авторский стиль оформления. Например, возможно всем заголовкам первого уровня придать зеленый цвет и больший размер или у всех элементов маркированного списка изменить маркер на красивое графическое изображение.

Таким образом, таблицы стилей снимают ограничения старых версий HTML на набор свойств тех или иных элементов документа.

Первая версия CSS (CSS – Cascading Style Sheets) **CSS 1** была рекомендована консорциумом по веб-технологиям W3C в 1996 году. Она содержит базовые свойства элементов, такие как шрифт, цвет, ширина полос, свойства блоков.

Вторая версия CSS (**CSS 2**) получила рекомендацию консорциума в 1998 году. В нее добавлено позиционирование элементов, селекторы различных типов (дочерние, смежные и универсальные). С 2002 года была начата работа над версией **CSS 2.1**. В ней исправлены ошибки предыдущей версии и представлено более аккуратное описание реализации таблиц стилей в браузерах. Рекомендацию CSS 2.1 получила в 2011 году.

Третья версия CSS (**CSS 3**) находится в стадии разработки и еще не получила рекомендации от организации World Wide Web Consortium. Она будет включать средства многоколоной верстки, различные скругления, тени, множественные фоны и прочее.

СЕЛЕКТОРЫ

Применять стили к элементам можно, только если мы обозначим браузеру сами элементы и новые значения их свойств. Выделение html-элементов происходит с помощью селекторов (от англ. select – выбор) или атрибута STYLE.



Селектор – это инструкция в CSS, которая указывает браузеру, какой элемент (элементы) выбрать для применения стиля.



Элементы можно выбрать на основе их имени, атрибутов, связи и других аспектов.

1. Применение стиля без селектора

Стиль задается непосредственно для одного какого-либо элемента в основном разделе документа с помощью параметра STYLE.

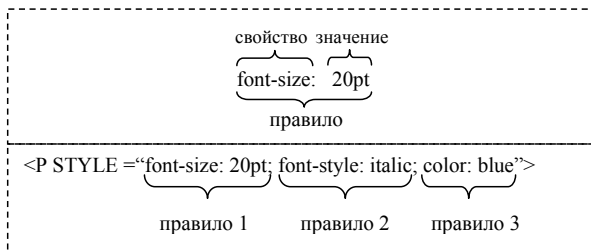
Пример.

`<P STYLE="font-size: 20pt; font-style: italic; color: blue">`

Здесь слово STYLE выступает в роли атрибута тега. Из примера видно, что значение параметра STYLE представляет собой последовательность свойств и их значений, разделенных точкой с запятой. Очередное свойство и его значение в этой последовательности разделяются двоеточием.



Каждая пара **свойство: значение**; называется **правилом** стиля



2. Селекторы элементов и селекторы потомков

Атрибут STYLE в предыдущем способе задает свойства только одного элемента, а с помощью тега `<STYLE>` можно задать свойства всех одноименных элементов. Контейнер `<STYLE>` может размещаться только в заголовке `<HEAD></HEAD>`.

```
<STYLE TYPE="text/css">
  селектор {правило1, правило2; ...}
</STYLE>
```

Гораздо удобнее располагать каждое правило на отдельной строке.

Пример.

`<STYLE TYPE="text/css">`

```
H1 {
  border-width: 1;
  border: groove;
  text-align: center;
  color: green
}
H2 {
  color: maroon;
  font-style: italic
}
```

```

P {
    text-indent: 10;
    color: blue;
    font-size: 12pt
}
B {
    color: Lime;
    font-size: 20pt
}
IMG {
    display: block
}
P, MARQUEE, TR, TH {
    color: rgb(102,102,0)
}
</STYLE>

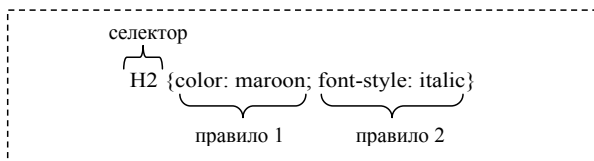
```



Таблица стилей может состоять из одного или нескольких селекторов и описаний стиля, определяющих внешний вид соответствующих элементов. Каждый селектор сопровождается описанием стиля. В приведенной таблице для первого селектора **H1, border-width: 1; border: groove; text-align: center; color: green** – описание стиля. В общем виде это выглядит следующим образом:



[html-элемент {описание стиля}]



В примере последняя строка тега **STYLE** показывает, как можно одно описание свойства применить сразу к нескольким элементам, перечисленным через запятую.

Следующий пример показывает, как можно выделить элементы, находящиеся внутри родительских. Такой селектор называют **селектором потомка** (descendant selector).

```

P B {color: Aqua}

```

Стиль для вложенных элементов *P B* будет использоваться браузером только тогда, когда элемент *B* окажется внутри элемента *P*. При этом свойство *font-size* будет унаследовано от предыдущего описания стиля тега *B*.

[элемент-родитель элемент-потомок {правила CSS}]

Для дополнительного чтения

Селектор дочернего элемента (*child selector*)

Отличие селектора дочернего элемента от селектора потомка в том, что если селектор потомка выбирает все потомки, независимо от уровня вложенности, то селектор дочернего элемента выбирает потомков только первого уровня.

BODY>P {line-height: 1.3}

DIV OL>LI P {color: pink}

Последний пример – это комбинация дочернего селектора и селектора потомка. OL должен быть потомком элемента DIV, LI должен быть потомком только первого уровня элемента OL, вложенного в DIV. P должен быть потомком элемента LI указанного списка OL. Перед и после треугольной скобки не должно быть пробелов!

[элемент-родитель>дочерний элемент {правила CSS}]

Селектор сестринского элемента (*sibling selector*)

Выбирает элемент, расположенный непосредственно за заданным элементом и имеющий с ним общего родителя.

H1 + H2 {margin-top: -5mm}

[элемент-сестра1 + элемент-сестра2 {правила CSS}]

3. Селекторы классов

Предыдущий способ позволяет присвоить любому элементу только один стиль. Иногда удобнее назначать для стилей имена и указывать их вместе с элементами или без элементов. Такие универсальные стили называют классами.

<STYLE TYPE = "text/css">

H1.red {color: red; text-align: center}

H1.green {color: green; text-align: left}

H1.yellow {color: yellow; text-align: justify}

A.copyright:hover {color: aqua}

.k1 {color: khaki; background: Purple;}

```
.слайдер {width: 300px; height: 500px; float: right}  
</STYLE>
```

В основном разделе html-документа можно в нужных местах указывать подходящий стиль для заголовка первого уровня.

```
<H1 CLASS = "red"> текст заголовка </H1>
```

или

```
<H1 CLASS = "yellow"> текст заголовка </H1>
```

При желании можно создать любое количество классов, каждый из которых будет иметь свой вариант форматирования для одного элемента.

В последнем примере (.k1) описан универсальный класс k1, не привязанный изначально ни к какому элементу. Применяется также через параметр CLASS.

```
<DIV CLASS = "слайдер"> </DIV>
```

4. Селекторы идентификаторов (ID) и контекстные селекторы ID

Этим способом можно создавать классы, связанные как со стандартными, так и с нестандартными html-элементами, то есть элементами веб-разработчика.

```
<STYLE TYPE = "text/css">
```

```
#списокСсылок {  
    color: maroon;  
    font-weight: bold;  
    text-align: center;  
}
```

```
</STYLE>
```

```
<UL ID = "списокСсылок"> ..... </UL>
```

С помощью идентификатора можно выделить какой-либо стандартный html-элемент для придания уникального стиля. Это придаст больше смысла html-коду и, соответственно, улучшит процесс разработки html-документа.

С атрибутом ID тесно связан структурный элемент **<DIV>** (от англ. division – часть, раздел). Тег **<DIV>** не описывает базовых элементов html-документа, таких как заголовок, абзац, список, таблица и т. д. Он предназначен для описания какой-либо структуры, созданной самим дизайнером. Типичным примером такого элемента является навигационная панель. Имя такой структуре дает тоже разработчик через атрибут ID. Таблица стилей используется для придания внешнего вида структурному элементу. Следующий пример кода создает стиль *Примечание*, которым оформляется структурный элемент с данным уникальным именем. В результате на веб-странице будут размещены два абзаца, обрамленные красной рамкой.

```
<STYLE TYPE = "text/css">
```

```
#Примечание {
```


border: medium double red;

}

</STYLE>

.....

<DIV ID = "Примечание">



<P> С помощью псевдоэлементов :before и :after разработчики задают стиль и местоположение генерируемого содержимого. Как следует из названия, псевдоэлементы :before и :after определяют его местоположение перед и после содержимого элемента, принадлежащего дереву документа, соответственно. В сочетании с этими псевдоэлементами свойство 'content' определяет содержимое вставляемого объекта. </P>

<P> Например, следующее правило вставляет строку "Примечание: " перед содержимым каждого элемента P, значение атрибута "class" которого равно "note": </P>

</DIV>



Существует важное правило: через атрибут ID можно присвоить элементу уникальное имя, которое нельзя использовать в пределах html-документа больше чем один раз.



DIV следует применять только тогда, когда нельзя использовать стандартный html-элемент.



Тег , в отличие от <DIV>, предназначен для определения внутренних элементов какой-либо структуры или стандартного элемента. С помощью тега можно не только выделить часть информации внутри других тегов, но и установить для нее свой стиль. Например, внутри абзаца (тега <P>) можно изменить цвет и размер первой буквы, если заключить ее в контейнер и определить соответствующий стиль.

Стиль нового структурного элемента можно сделать зависимым от окружения.

<STYLE TYPE = "text/css">

#navBar1 p {font-size: 24pt; color: silver}

#navBar1 ul {list-style-image: url(z9.jpg)}

#navBar1 table {width: 65%; font-size: 20pt}

table#work {width: 65%; font-size: 12pt;}

</STYLE>

Этот пример (первые три правила) показывает, как задать внешний вид всем абзацам, маркированным спискам и таблицам, входящим в структурный элемент **DIV** с **ID = "navBar1"**. Последнее правило показывает, как можно задать стиль отдельной таблице, имеющей **ID = "work"**.

5. Селекторы псевдоклассов

Псевдокласс используется, чтобы определить особое состояние элемента, такое как, например, движение мышки над элементом или посещенная гиперссылка.

Селекторами псевдоклассов выделяют элементы, различные состояния которых будут зависеть от действий посетителя или состояния браузера.

Синтаксис применения псевдоклассов:

[Селектор:Псевдокласс {Описание правил стиля}]

Условно все псевдоклассы делятся на три группы:

1. Определяющие состояние элементов.
2. Имеющие отношение к дереву элементов.
3. Указывающие язык текста.

Каскадные таблицы стилей поддерживают следующие псевдоклассы гиперссылок:

:link (непосещенная ссылка)

:visited (посещенная ссылка)

:active (активная, то есть выбранная ссылка)

Для многих объектов веб-страницы, включая гиперссылки, определены **динамические псевдоклассы**:

:hover (мышинный курсор нависает над элементом)

:focus (элемент в фокусе, то есть тот, на котором позиционируется курсор)

Пр и м е р.

<STYLE TYPE = "text/css">

A:link {color: red;}

A:visited {color: blue;}

A:hover {color: yellow; background: gold;}

A:active {color: lime;}

</STYLE>

Важно заметить порядок следования в описании данных стилей для гиперссылок. Он должен быть как в приведенном примере. Данный способ описания стилей изменяет оформление всех гиперссылок на веб-странице. Следующий пример показывает применение псевдоклассов к классам гиперссылок.

Пр и м е р.

<STYLE TYPE="text/css">

A:link.внутренняяСсылка {color: silver}

```

A:hover.внутренняяСсылка {
    color: yellow;
    background-color: gold
}
A:link.внешняяСсылка {color: red}
A:hover.внешняяСсылка {
    color: pink;
    background-color: forestGreen
}
</STYLE>
...
<BODY>
<P>
  <A CLASS="внутренняяСсылка" HREF="page1.htm"> страница 1
</A> <BR>
  <A CLASS="внешняяСсылка" HREF="http://ya.ru/"> страница 2
</A> <BR>
  <A HREF="page30.htm"> страница 3 </A> <BR>
  <A CLASS="внешняяСсылка" HREF="http://aport.ru/"> страни-
  ца 4 </A>
</P>
</BODY>

```



Псевдоклассы, имеющие отношение к дереву документа

Псевдокласс `:first-child` выбирает первый элемент среди группы дочерних по отношению к селектору.

:first-child (применяется к первому дочернему элементу родителя)

Псевдокласс `:first-child` удобнее всего использовать в тех случаях, когда требуется задать разный стиль для первого и остальных однотипных элементов.

Пр и м е р.

```
LI:first-child {color: red}
```

Данный пример показывает изменение цвета содержимого первого пункта каждого размещенного на веб-странице списка.

Псевдокласс, задающий язык текста

`:lang`

В качестве языка могут выступать следующие значения: ru — русский; en — английский; de — немецкий; fr — французский; it — итальянский. Общий вид конструкции

Элемент:lang(язык) { ... }

6. Универсальный селектор

Универсальный селектор обозначается символом * и предназначен для выбора всех элементов веб-страницы. Часто применяется для придания одинакового стиля всем элементам с целью одинакового отображения всеми браузерами. Например, следующее правило отменяет вывод по умолчанию внутренних и внешних полосок для каждого элемента:

* {padding: 0px; margin: 0px}

Для дополнительного чтения

7. Селекторы атрибутов

Выбирает элементы по наличию или значению заданного атрибута. Общий вид:

Элемент[Имя атрибута]

Элемент[Имя атрибута=Значение атрибута]

Отсутствие *Элемент* обозначает, что в этом месте стоит универсальный селектор, который можно опустить.

[TITLE] {color: blue}

SPAN[CLASS=example] {color: blue}

SPAN[HELLO="Cleveland"][goodbye="Columbus"] {color: blue}

A[Href="http://dikandr.ru/"]:hover {background-color: gold}

A[REL~="copyright"] {text-decoration: line-through}

Знак ~ означает примерное сходство. Атрибут REL определяет отношения между текущим документом и файлом, на который делается ссылка. Это необходимо, чтобы браузер знал, как использовать подключаемый документ.

*[LANG=fr] {display: none}

Все элементы на французском языке будут скрыты.

Вопросы и упражнения

1. Что такое селектор и для чего он применяется?
2. Каким образом можно изменить стиль элемента без использования селектора?
3. Из чего состоит css-правило?

4. Какими бывают селекторы?
5. Чем отличается селектор класса от селектора идентификатора?
6. Чем удобны в применении контекстные селекторы?
7. На какие элементы распространяет правила стиля универсальный селектор?
8. Какие псевдоклассы вносят динамические эффекты на веб-страницу?
9. Каким тегом можно выделить часть элемента?
10. Каким тегом можно объединить в одно целое несколько элементов?
11. Сколько элементов с одинаковым идентификатором может содержать html-документ?

СПОСОБЫ ПРИМЕНЕНИЯ СТИЛЕЙ

1. Универсальный атрибут STYLE в html-теге

Данный способ подробно описан в разделе Селекторы. Общий вид данного способа:

$\langle \alpha \text{ STYLE} = \beta \rangle$,

где α – тип html-элемента, β – правила CSS.

2. Тег STYLE

Данный способ также подробно описан в разделе Селекторы. Тег STYLE располагается в заголовочном разделе html-документа и содержит различные селекторы.

`<HEAD>`

`<STYLE TYPE = "text/css">`

`...`

`...`

`</STYLE>`

`</HEAD>`



3. Связывание с внешней таблицей стилей (тег <LINK>)

Таблицы стилей могут сохраняться в отдельном файле и использоваться несколькими html-документами. Более того, один и тот же документ может обращаться к нескольким таблицам стилей одновременно. Именно поэтому таблицы стилей называли **каскадными**. Стили накладываются друг на друга, и более поздние определения отменяют принятые ранее установки.

`<LINK HREF = "table_of_style.css" REL="stylesheet" TYPE = "text/css">`

Здесь *table_of_style.css* есть имя файла, где описываются таблицы стилей. Тег <LINK> помещается в заголовочный раздел html-документа.

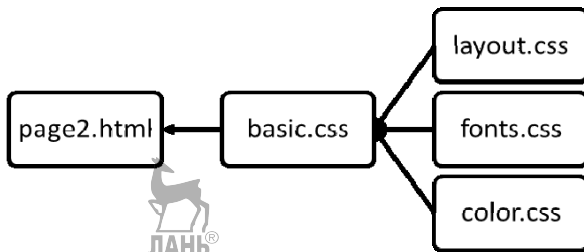
4. Импортирование

Правила импорта должны стоять первыми в таблице стилей. Нужно иметь в виду, что правила в связанных таблицах (LINK) будут всегда переопределять

правила импортированных таблиц. Устаревшие браузеры не поддерживают импорт.

```
<STYLE TYPE = "text/css">  
  @import (url(layout.css))  
  LI:first-child {color: red}  
  DT:lang(en) {color: green}  
</STYLE>
```

Импортировать таблицу можно не только в html-документ, но и в другую таблицу стилей. При этом правила импорта должны идти первыми в исходной таблице. Использовать более двух уровней вложенности таблиц не рекомендуется, так как браузеры плохо поддерживают такой способ.



Некоторые браузеры выводят html-страницу сначала без импортированных стилей и только после их полной загрузки применяют к выведенному содержимому. Этот эффект называют **FOUC** (Flash Of Unstyled Content) – мелькание нестилизованного содержимого. С помощью размещения сценария или элемента ссылки в заголовочном разделе этот эффект можно предотвратить.

Вопросы и упражнения

Используя CSS, создайте:

1. разноцветные гиперссылки на одной веб-странице;
2. гиперссылки, меняющие свой цвет при наведении на них курсора мыши;
3. любой элемент (элемент таблицы, текстовая область, абзац), меняющий цвет фона при наведении на него курсора мыши;
4. заголовок размером шрифта 25 пт и цветом, отличным от чёрного.

КОММЕНТАРИИ

Для оставления заметок разработчику предоставляется возможность комментировать код. Символы комментариев такие же, как и в языке программирования C и JavaScript. Комментарии могут быть как однострочными, так и многострочными.

```
/* Комментарий в CSS */  
/* -----
```

Многострочный комментарий CSS

*/-----

Пример.

```
<STYLE TYPE = "text/css">
```

```
.menu:hover {left: 0;}
```

```
/*-----
```

* Красный квадрат

* =====*/

```
.эффект2 {
```

```
width: 100px;
```

```
height: 100px;
```

```
background: red;
```

```
transition: width 2s;
```

```
}
```

```
.эффект2:hover {width: 300px;}
```

```
</STYLE>
```



НАСЛЕДОВАНИЕ

Свойства элемента наследуются его потомками, то есть вложенными в него элементами. Например, если в стилях для селектора TABLE задать цвет текста, то он автоматически устанавливается для содержимого ячеек. Однако не все стилевые атрибуты наследуются. Чтобы определить, наследуется значение стилевого атрибута или нет, требуется заглянуть в справочник по свойствам CSS.



Стиль, присвоенный элементу всегда переопределяет унаследованный стиль.



В названии описываемой технологии присутствует слово «каскад», которое отражает возможность наложения нескольких стилевых правил на один и тот же элемент. При этом возможны конфликты. Чтобы избежать конфликтов, введены приоритеты и правила применения стилей браузером, которые отражены в понятии «специфичность».

Специфичность – это способ, с помощью которого браузеры определяют, какие значения свойств CSS будут применены к элементу.

Специфичность представляет собой вес, придаваемый правилу CSS. Вес правила определяется количеством каждого из типов селекторов в данном правиле. Если у нескольких правил специфичность одинакова, то к элементу применяется последнее по порядку следования в коде правило CSS. Специфичность имеет значение только в том случае, если одному элементу соответ-

ствуется несколько правил. Согласно спецификации CSS, правило, непосредственно предназначенное элементу, всегда будет иметь больший приоритет, чем правила, унаследованные от предка.

В следующем списке типы селекторов расположены по возрастанию специфичности:

1. Селекторы типов элементов и псевдоэлементов.
2. Селекторы классов, селекторы атрибутов и псевдоклассов.
3. Селекторы идентификаторов.

Стили, объявленные в элементе (например, `style="font-size: 24pt"`), всегда переопределяют любые правила, и их специфичность можно считать наивысшей.

При объявлении стиля можно применять модификатор **!important**, тогда объявление получает наивысший приоритет среди всех прочих объявлений. Модификатор **!important** влияет на специфичность. Поскольку **!important** усложняет отладку, нарушая естественное каскадирование стилей, он **не рекомендуется**, а значит, следует избегать его использования.

Специфичность — это некоторая условная вычисляемая величина. За каждый идентификатор (id) начисляется 100 единиц, за каждый класс (class) и псевдокласс начисляется 10, за каждый селектор элемента и псевдоэлемент начисляется 1. Складывая указанные значения, получаем значение специфичности для данного селектора. Встроенный стиль, добавляемый к тегу через атрибут `style`, имеет специфичность 1000.

Например, специфичность селектора `#navB ul li {color: gold;}` будет равна 102 единицы (100+1+1).

Вопросы и упражнения

1. Подсчитайте специфичность селектора:
 - a. `ul.list li:hover p`
 - b. `#banner .block1 .block2 img`

СЕМАНТИЧЕСКАЯ ВЕРСТКА

Развитие веб-технологий идет по пути создания инструментов осуществления **семантической разметки веб-документов**, тогда как в настоящее время все еще используется смешанная разметка: табличная и блочная (семантическая). В связи с направлением развития в сторону семантики, то есть смысла, веб-разработчику следует там, где это возможно, использовать элементы по их прямому назначению.

Семантическая разметка ≈ смысловые элементы

Из стандартных html-элементов можно делать дополнительные смысловые элементы, добавляя параметры ID или CLASS. Идентификатор может быть полезным как для создания целой структуры (например, навигационная панель или баннер), так и для выделения отдельного чем-либо примечательного элемента, например отдельной гиперссылки или поля формы.

В именах для ID и CLASS общепринято использовать символы верхнего регистра только для первых букв второго и последующего слов. Например:

основноеСодержание
главнаяНавигационнаяПанель
верхнийБаннер

Имена для параметров ID и CLASS лучше давать, исходя из их смыслового назначения, нежели из их расположения на странице или формата, так как со временем расположение и формат могут измениться. Например, имя **верхнийБаннер** не является удачным, так как впоследствии элемент может переместиться и имя не будет соответствовать действительности. Лучше подойдет имя **заголовочныйБаннер** или **рекламныйБаннер**.

Добавляя один и тот же идентификатор (ID) на каждую страницу сайта, мы тем самым добиваемся единого оформления, то есть как бы единого начертания. Такой способ называют «**сигнатурой сайта**».

Пр и м е р.

```
body #www_dikandr_ru {  
    font-size: 20pt;  
    background-image: url(jfhgt.jpg); color: white  
}
```

Такой подход позволит посетителям сайта переопределять таблицы стилей разработчика на свои собственные.

Существует микроформат **XFN** (XHTMLFriendsNetwork), который представляет собой простой способ показать связи между людьми с помощью гиперссылок, разработанный Global Multimedia Protocols Group. XFN позволяет выразить отношение к человеку, добавив соответствующее значение атрибута **REL** к ссылке на его сайт, например:

```
<A HREF = "http://jeff.example.org" REL="friendmet">.
```

Значения атрибута rel в XFN

| | |
|--------------|--|
| friend | друг |
| acquaintance | знакомый |
| contact | кто-либо, кого вы знаете, как найти (обычно тоже знает, как найти вас) |
| met | встречались лично |
| co-worker | вместе работаете |
| colleague | коллега (общие интересы) |
| co-resident | живете на одной улице |

| | |
|------------|------------------------------------|
| child | ребёнок |
| parent | родитель |
| sibling | брат или сестра |
| spouse | супруг или супруга |
| kin | родственник |
| muse | вдохновляет вас |
| date | кто-либо, с кем вы
встречаетесь |
| sweetheart | любимый человек |
| me | ваш сайт |

Все вышеперечисленные значения являются необязательными и могут быть объединены. Например:

REL = "acquaintance colleague date"


Чтобы браузеры и поисковые системы знали о том, что сайт поддерживает XFN, необходимо в теге <HEAD> добавить атрибут PROFILE со значением <http://gmpg.org/xfn/11>.

<HEAD PROFILE="http://gmpg.org/xfn/11">

Используя селекторы атрибутов, можно придать определенные стили гиперссылкам, составляющим группы микроформата XFN.

A[REL="colleague "] {background: url(colleaguelink.gif)

no-repeat right top; padding-right: 30px}

 Вопросы и упражнения

1. Назовите три типа разметки веб-страниц. Чем они отличаются друг от друга?
2. Какая разметка веб-документа считается перспективной?
3. Как можно добавить смысл в обычные html-элементы?
4. По какому правилу обычно конструируются смысловые имена html-элементов?
5. Что такое сигнатура сайта?
6. **Новый раздел.** Используя тег <DIV>, создайте нестандартный элемент на веб-странице. Например: навигационную панель, информационный баннер; или оформите как единый элемент картинку с текстом на едином фоне и с общей пунктирной или двойной границей. Примените CSS для единого оформления созданного элемента (общий фон, граница и т. д.).



ЦВЕТОВОЙ КРУГ

Цветовой круг – это способ представления непрерывности цветовых переходов, а также модели HSB (англ. Hue, Saturation, Brightness – оттенок, насыщенность, яркость). Сектора круга окрашены в различные цветовые тона, размещенные в порядке расположения спектральных цветов, при этом пурпурный цвет связывает крайние (красный и фиолетовый) цвета.

Первая попытка привести видимые цвета в систему принадлежала Исааку Ньютону (ученый, работавший со светом). Он разработал теорию, что все цвета – это смеси красного, зеленого и синего света (**RGB**). В результате в 1666 году появился первый известный цветовой круг. Цветовая система Ньютона – цветовой круг, составленный из семи секторов (цвета радуги): красного, оранжевого, желтого, зеленого, голубого, синего и фиолетового (каждый охотник желает знать, где сидит фазан). Цветовой круг RGB представляет, как с помощью трех световых лучей генерируются цвета на экране телевизора или мониторе компьютера. На мониторе, на телевизоре и на телефоне все цветовые оттенки получаются высветлением темного экрана красными, зелеными и синими лучами.

Для цветной печати существует свой цветовой круг **CMY(K)**. Первичные цвета Cyan, Magenta и Yellow плюс Black. Зеленовато-синий, пурпурный и желтый плюс черный в цветовой модели CMYK являются основными для цветных принтеров. Цветовая модель CMYK имеет широкое применение в полиграфии.

Основным инструментом для подбора цветовых сочетаний различных элементов на веб-странице или проверки гармоничности подобранных цветов является цветовой круг **RYB**. Этот цветовой круг используют художники,

артисты и дизайнеры для смешивания пигментов и получения пигментарных цветовых смесей.

Круг RYB состоит из 12 частей. В его основе лежат три основных цвета (красный, желтый, синий – RYB), смешивая которые, мы получаем три дополнительных (составных) – зеленый, оранжевый и фиолетовый.

Red + Yellow = Orange

Yellow + Blue = Green

Blue + Red = Purple



Смешивая три основных цвета с тремя составными вторичными цветами, получим 6 третичных цветов: Red-Orange, Yellow-Orange, Yellow-Green, Blue-Green, Blue-Purple, Red-Purple. Складывая три первичных, три вторичных и шесть третичных цветов, в сумме получаем круг из 12 цветов.

| | | | | | |
|-------------------------|-----------------------|--------------------------|-----------------------|--------------------------|-----------------------|
| RED
#FE2712 | R-O
#FC600A | ORANGE
#FB9902 | Y-O
#FCCC1A | YELLOW
#FEFE33 | Y-G
#B2D732 |
| GREEN
#66B032 | B-G
#347C98 | BLUE
#0247FE | B-P
#4424D6 | PURPLE
#8601AF | R-P
#C21460 |

Одна из современных классификаций, используемых дизайнерами, подразумевает следующее деление цветов. Светлые и пастельные тона, которые визуально увеличивают пространство. Темные и интенсивные цвета, которые создают уют, но одновременно уменьшают помещение. Холодные оттенки (синий, зеленый, пурпурный), которые отвечают за создание спокойной и расслабляющей атмосферы. Теплые тона: к ним относят желтые, красные и оранжевые оттенки; они приносят в интерьер солнечность и нотки веселья. – Читайте подробнее на FB.ru: <http://fb.ru/article/303488/tsvetovoy-krug-nyutona-organizatsiya-tsvetovyyih-ottenkov>.

Самым лучшим кругом для подбора удачных цветовых сочетаний является круг RYB.

Цветовые схемы

Для подбора цвета различным элементам веб-страницы используют цветовые схемы, взятые из круга. Цветовые схемы представляют собой

логические комбинации цветов на цветовом колесе. Цель цветовой схемы – создать эстетическое чувство стиля и привлекательности.

Ахроматические цветковые схемы

Черно-белая ахроматическая цветковая схема используется по умолчанию для веб-страницы. Однако белый цвет на черном фоне в ахроматической цветовой гамме часто предпочтительнее для небольших устройств. Белый на черном требует меньше энергии (улучшает срок службы батареи) на большинстве устройств отображения.

БЕЛЫЙ ТЕКСТ
НА ЧЕРНОМ ФОНЕ

ЧЕРНЫЙ ТЕКСТ
НА БЕЛОМ ФОНЕ

Монохроматические цветковые схемы

Монохроматические цветковые схемы легко создавать, так как они используют только один цвет. Монохроматические схемы используют разные тона с одного угла на цветовом колесе (один и тот же оттенок).



Монохроматические цвета цветковой схемы гарантированно подходят друг другу, потому что все они родом из одной семьи.

#FFDEDB

#FE8176

#FE2712

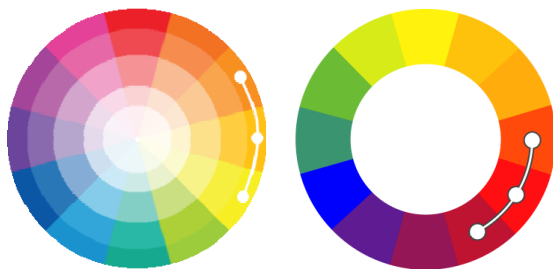
#A70F01

#340D09

Подобрать монохроматическую цветковую схему в онлайн можно с помощью генератора на странице® https://www.w3schools.com/colors/colors_monochromatic.asp.

Аналоговые цветковые схемы

Аналоговые цветковые схемы также легко создавать. Аналоговые цветковые схемы создаются с помощью цветов, расположенных рядом друг с другом на цветовом круге. Аналоговая триада – мягкое и приятное сочетание трех цветов, часто встречающееся в природе. Можно включать в схему и пять рядом стоящих цветов. Пример сочетания аналоговых приглушенных цветов: желто-оранжевый, желтый, желто-зеленый, зеленый, сине-зеленый.



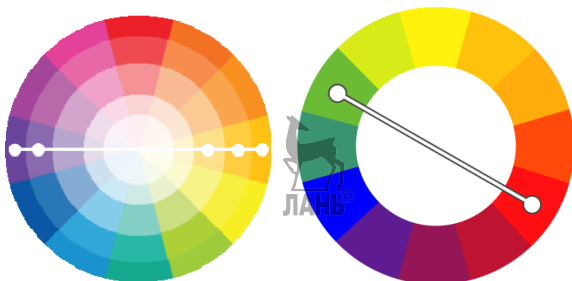
Если аналоговая цветовая схема не имеет достаточного контраста между цветами (для приятного дизайна сайта), то ее можно отрегулировать, чтобы придать ей больше разнообразия.



Подобрать аналоговую цветовую схему в онлайн можно с помощью генератора на странице https://www.w3schools.com/colors/colors_analogous.asp.

Комплементарные (взаимодополняющие) цветовые схемы

Комплементарные схемы создаются путем объединения цветов с противоположных сторон цветового колеса. Комплементарные цвета – контрастные цвета. Очень удачно можно использовать контрастные сочетания для выделения деталей. Выглядят сочетания очень живо и энергично, особенно при максимальной насыщенности цвета.



Комплементарные цветовые схемы в основном состоят из двух дополнительных цветов, расширенных серыми тонами, оттенками и затемнениями.

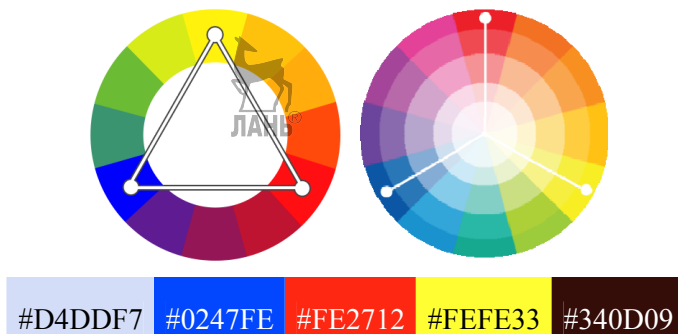


Подобрать комплементарную цветовую схему в онлайн можно с помощью генератора на странице https://www.w3schools.com/colors/colors_complementary.asp.

Триада

Триадные схемы состоят из оттенков, одинаково расположенных вокруг цветного колеса. В такой схеме рекомендуется выбирать один главный цвет, а два других – для расстановки акцентов. Триада обеспечивает высокую контра-

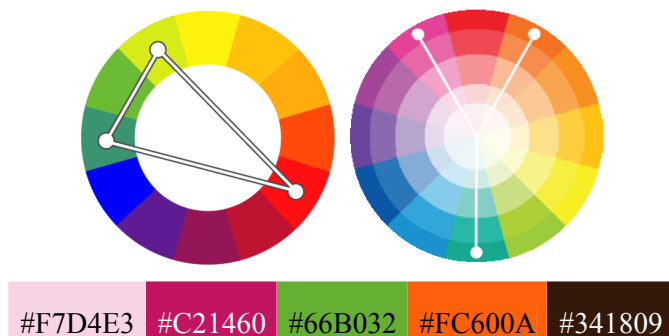
стность при сохранении гармонии. Такая композиция выглядит достаточно живой даже при использовании бледных и ненасыщенных цветов.



Подобрать аналоговую цветовую схему в онлайн можно с помощью генератора на странице https://www.w3schools.com/colors/colors_triadic.asp.

Составная комплементарная цветовая схема

Составные схемы практически совпадают с дополнительными схемами. Вместо использования цветов, которые являются противоположными, используются цвета с обеих сторон противоположного оттенка. Сочетание включает основной цвет и два дополнительных. Выглядит эта схема почти настолько же контрастно, как комплементарная, но не настолько напряженно.



Подобрать аналоговую цветовую схему в онлайн можно с помощью генератора на странице https://www.w3schools.com/colors/colors_compound.asp

МОДЕЛЬ КОНТЕЙНЕРОВ

Каждый html-элемент окружен рядом областей, свойствами которых можно управлять через таблицы стилей.





Вокруг элемента располагается внутренняя пустая область, которой можно придать некоторый цвет и изменить размер через соответствующие свойства стилей. За ней следует граница, то есть рамка, стиль которой может быть определен через свойство **border**. И наконец завершает слои внешняя пустая полоска, которую можно назвать внешним полем. Свойства внешней полосы настраиваются через свойство **margin**. У внешней полосы отсутствует фон, поэтому она является невидимой. Но у нее имеется настраиваемый размер, что определяет расстояние между соседними элементами на странице.

Браузеры имеют встроенные таблицы стилей, в которых хранятся значения стилей по умолчанию. Эти значения в различных браузерах могут отличаться друг от друга. Поэтому существует хороший способ сбрасывания в ноль значений внутренних и внешних полосок всех элементов документа с помощью универсального селектора.

* **{padding: 0px; margin: 0px}**

Свойства **width** и **height** определяют размер элемента без полосок и границы. Тогда в соответствии со спецификацией* ширина всего контейнера вычисляется следующим образом:

| | | | | | | | | |
|---------------------------|---|--------------------|---|--|---|-------------------|---|------------------------------|
| ширина
контей-
нера | = | ширина
элемента | + | ширина
внутрен-
ней по-
лоски | + | ширина
границы | + | ширина
внешней
полоски |
|---------------------------|---|--------------------|---|--|---|-------------------|---|------------------------------|

Для изменения алгоритма расчета ширины и высоты элемента применяется свойство стилей **box-sizing**, которое позволяет сделать так, чтобы свойства **width** и **height** задавали размеры не контента, а размеры блока (контейнера). Свойство **box-sizing** входит в спецификацию CSS 3, поэтому может не восприниматься устаревшими версиями браузеров.

* 10 Visual formatting model details [Electronic resource]. — Mode of Access : <http://www.w3.org/TR/CSS21/visudet.html#blockwidth>.

CSS-live [Электронный ресурс]. — Режим доступа : <http://css-live.ru/tajny-css2-1/raznica-mezhdu-widthauto-i-width100.html>

Значения свойства box-sizing

| | |
|-------------|--|
| content-box | Основывается на стандартах CSS, при которых свойства width и height задают ширину и высоту контента и не включают в себя значения отступов, полей и границ |
| border-box | Свойства width и height включают в себя значения внутренних полей и границ, но не отступов (margin) |
| padding-box | Свойства width и height включают в себя значения внутренних полей, но не отступов (margin) и границ (border) |
| inherit | Наследует значение родителя |

В современных браузерах в пакет инструментов веб-разработчика включена визуальная модель контейнера, выбранного на странице элемента. Вызывается окно клавишей F12. Аналогичные окна есть и в других браузерах. В Firefox такое окно вызывается из контекстного меню пунктом «Исследовать элемент».

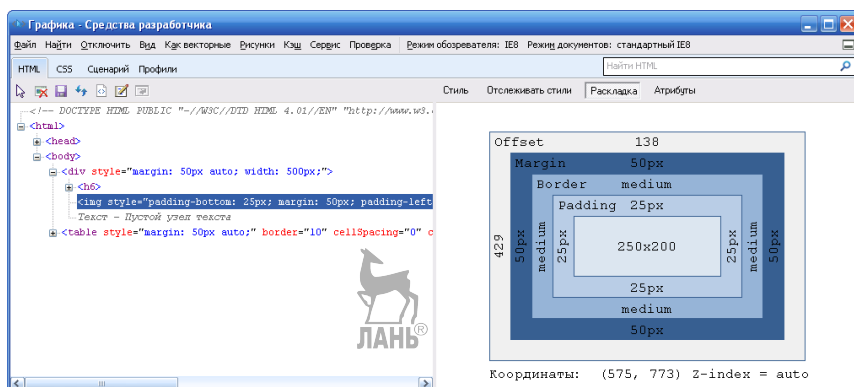
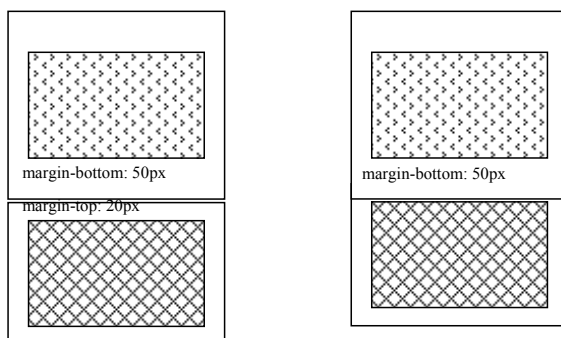


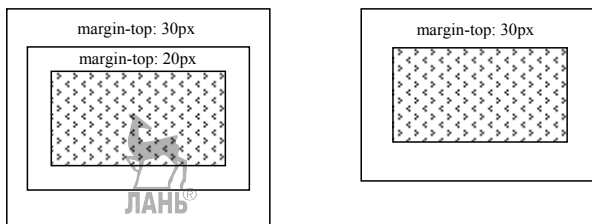
Рисунок 9. Окно средства разработчика браузера IE10

Слияние внешних полосок

При соприкосновении вертикальных внешних полосок они сливаются, образуя одну полосу. При этом ширина результирующей полосы равна большей из двух взаимодействующих.



Примерно то же самое происходит, если элемент находится внутри родительского контейнера и не имеет рамок и внутренних полосок, разделяющих внешние полоски.



Если на веб-странице находится пустой элемент (например, пустой абзац текста) и он не имеет внутренних полосок и границ, то при соприкосновении его нижней и верхней внешней полоски, они сливаются по тому же правилу.

Вопросы и упражнения

1. Какими областями окружен любой html-элемент?
2. Какими свойствами стилей можно задать параметры каждой области, окружающей элемент веб-страницы?
3. По какому правилу происходит слияние внешних полосок соприкасающихся элементов?
4. По какой формуле вычисляется ширина html-контейнера?
5. Какое css-свойство может изменить формулу расчета ширины html-контейнера?
6. **Стилизация заголовка.** С помощью секционного элемента `<header>` создайте стилизованный заголовок веб-страницы с нестандартными значениями внутренних и внешних отступов, фоном и границей.
7. **Стилизованная кнопка.** Создайте стилизованные кнопки для:
 - a. навигации по сайту;
 - b. внешних переходов;
 - c. запуска или остановки какого-либо процесса;
 - d. отправки данных;
 - e. заказа товара или услуги.

ПОЗИЦИОНИРОВАНИЕ

Нормальный поток

По умолчанию все html-элементы выводятся в нормальном потоке (слева направо, сверху вниз). Все html-элементы подразделяются на строчные и блочные. Каждый строчный элемент занимает часть строки, в которую он выводится, находясь в нормальном потоке. Если выводится подряд несколько строчных элементов, то все они будут идти в строке друг за другом слева направо. Блочный элемент, в отличие от строчного, занимает всю строку, и следующий за ним в нормальном потоке объект будет выведен на новой строке.

| Строчный элемент | Блочный элемент |
|---------------------------|-------------------------|
| <code><A></code> | <code><H1></code> |
| <code><AREA></code> | <code><P></code> |

| Строчный элемент | Блочный элемент |
|------------------|-----------------|
| <BUTTON> | |
| | |
| <INPUT> | <DL> |
| <IFRAME> | <TABLE> |
| | <FIELDSET> |
| <SELECT> | <DIV> |
| <TEXTAREA> | |
| <TD> | |

В CSS существует свойство **display**, которое может изменить тип элемента. Следующий пример иллюстрирует, как можно превратить последовательность подряд идущих абзацев в один абзац.

P {display: inline}

Аналогичным образом можно превратить блочный элемент в строчный. Нередко веб-разработчики представляют список как линейную структуру без маркеров или номеров для создания навигационной панели.

```
<STYLE TYPE = "text/css">
```

```

#navBar LI {
    display: inline;
    margin: 10px;
    padding: 10px;
    border-right: 2px solid black;
}
#navBar OL {
    list-style-type: none
}
#navBar LI:hover {
    cursor: pointer;
    background-color: brown;
    color: white
}

```

```
</STYLE>
```

```
...
```

```

<OL ID = "navBar">
  <LI> Ссылка 1
  <LI> Ссылка 2
  <LI> Ссылка 3

```

 Ссылка 4
 Ссылка 5

...

Ссылка 1 | Ссылка 2 | Ссылка 3 | Ссылка 4 | Ссылка 5 |



У строчных элементов нельзя изменить width и height. Ширина и высота строчных элементов определяется их содержимым. Возможно задание внешних горизонтальных отступов (margin) и внутренних отступов (padding). Для задания вертикальных отступов можно использовать свойство **display: inline-block**.



В нормальном потоке css-свойство позиционирования элементов имеет значение по умолчанию static.

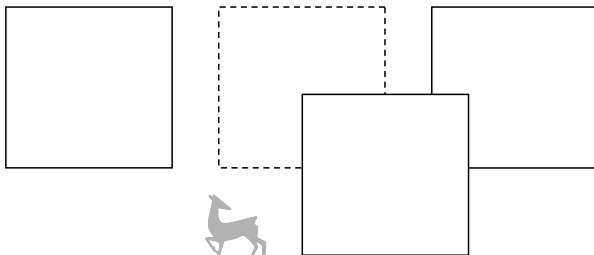
position: static;

Относительное позиционирование

При относительном позиционировании элемент смещается на заданный горизонтальный и вертикальный отступ относительно своего положения в нормальном потоке.

```
#element {  
  position: relative;  
  left: 20px;  
  top: 30px;  
}
```

Несмотря на то, что элемент сместился, он продолжает занимать первоначальное пространство в нормальном потоке. Смещенные элементы можно накладывать на другие объекты.



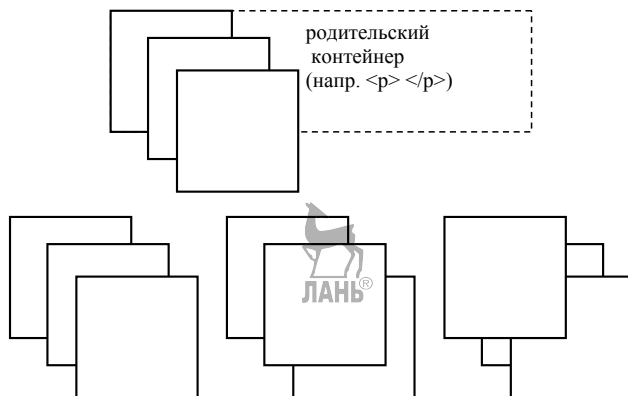
За порядок, в котором следуют объекты друг за другом, отвечает свойство z-index:

z-index: α ,

где α – целое положительное или отрицательное число; чем больше число, тем выше объект.

По умолчанию применяется значение **auto**, которое задает порядок элементов, исходя из их положения в коде HTML и принадлежности к родителю, поскольку дочерние элементы имеют тот же номер, что их родительский элемент. Значение **inherit** указывает, что оно наследуется у родителя.

Следующий пример иллюстрирует каскадное расположение картинок и выход картинки на передний план при наведении на нее курсора мышки.



```
<STYLE TYPE = "text/css">
  IMG:hover {z-index: 10}
</STYLE>
.....
<P>
  <IMG SRC = "pic1.jpg" STYLE = "position: relative">
  <IMG SRC = "pic2.jpg" STYLE = "position: relative; top: 30px; left: -
    150px">
  <IMG SRC = "pic3.jpg" STYLE = "position: relative; top: 60px; left: -
    300px">
</P>
```

Абсолютное позиционирование

В отличие от относительного позиционирования, при абсолютном позиционировании элемент извлекается из нормального потока и больше не занимает места в нем. Другие элементы, находящиеся в потоке, не замечают его. Элемент располагается относительно родительского элемента либо относительно исходного содержащего блока, в том числе всей веб-страницы. При абсолютном позиционировании html-элемент можно сместить вниз, вверх, вправо и влево. При наложении объектов порядок их следования регулирует свойство **z-index**, описанное выше.

```
#element {
  position: absolute;
  left: 20px;
```

```
right: 30px;
top: 20px;
bottom: 10px
```

```
}
```

Следующий пример иллюстрирует замену одного блока (картинки) другим (блок текста) при наведении на него курсора мыши.

```
<STYLE TYPE = "text/css">
```

```
  #баннер H6 {
```

```
    position: absolute;
```

```
    z-index: 0;
```

```
    background-color: gold;
```

```
    width: 200px;
```

```
    height: 150px
```

```
  }
```

```
  #баннер IMG {
```

```
    position: absolute;
```

```
    z-index: 1
```

```
  }
```

```
  #баннер IMG:hover {z-index: -1}
```

```
  #баннер H6:hover {z-index: 2}
```

```
</STYLE>
```

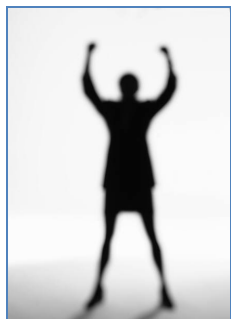
```
...
```

```
<DIV STYLE = "width: 200px; height: 150px" ID = "баннер">
```

```
  <H6> Человек, ищущий смысл жизни</H6>
```

```
  <IMG SRC = "images/man1.gif">
```

```
</DIV>
```



Человек,
ищущий
СМЫСЛ
ЖИЗНИ

При наложении объектов можно использовать свойство стилей **visibility**, которое позволяет изначально скрыть объект, а, например, при наведении курсора мыши на рядом расположенный объект показать его. Важно знать при этом особенности «включения» объектов. Сначала применяем свойство видимости к тем объектам контейнера, которые будут видны, затем присваиваем значение скрытости всему контейнеру (при этом останутся видимыми объекты с присвоенным свойством видимости). И наконец, добавляем псевдокласс `hover` к контейнеру с правилом, открывающим весь контейнер.

```
<STYLE TYPE = "text/css">
    #баннер2 Р.текст1 {visibility: visible}
    #баннер2 {visibility: hidden}
    #баннер2: hover {cursor: pointer; visibility: visible}
</STYLE>

...
<TABLE CELLPADDING = 10 CELLSPACING = 0 STYLE = "margin: 0px
auto; width: 300px; border: 5px solid gold">
    <CAPTION> Фото </CAPTION>
    <TR> <TD STYLE = "background-color: whitesmoke">
        <IMG SRC="images/itsme1.jpg" style="margin: 0px auto; width:
200px">
        <DIV ID = "баннер2">
            <P CLASS = "текст1">Подробнее...</P>
            <P CLASS = "текст2">Человек с рапирой в руке </P>
        </DIV>
    </TD>
</TR>
</TABLE>
```



Подробнее...



Подробнее...

Человек с рапирой в руке



Абсолютное позиционирование работает в родительском контейнере с относительным позиционированием или в контейнере `<BODY>`.



Вопросы и упражнения

1. **Обратная сторона.** Разместите на веб-странице изображение игральной карты или монеты. Создайте с помощью css-свойств эффект переворачивания карты (монеты) при наведении курсора мыши.

Фиксированное позиционирование

Фиксированное позиционирование – это разновидность абсолютного позиционирования. При фиксированном позиционировании, так же как и при абсолютном, элемент извлекается из нормального потока и больше не занимает места в нем. Позиция элемента задается отступами вниз, вверх, вправо и влево от верхнего левого края родительского элемента. Единственным и важным отличием фиксированного позиционирования является жесткая привязка элемента к его начальному положению на экране монитора. Это значит, что при скроллинге (прокрутке) фиксированный элемент будет всегда оставаться на своем месте.

```
#element {  
    position: fixed;  
    left: 20px;  
    right: 30px;  
    top: 20px;  
    bottom: 10px  
}
```



Известно, что в мире работает огромное число мониторов с различным разрешением, поэтому у абсолютно фиксированного элемента не всегда отступ справа будет оставаться одинаковым на различных мониторах. Особенно это заметно, когда объект изначально выровнен по правому краю. Проблема фиксированного выравнивания по правому краю имеет решение, если заключить исходный элемент (f1) в контейнер (f2) с шириной равной **100%** экрана, зафиксировать контейнер и задать свойству **float** исходного объекта значение **right**. Таким образом, исходный объект всегда будет выровнен по правому краю относительно родительского контейнера, ширина которого совпадает с шириной экрана, на котором просматривается веб-страница. Фиксированное позиционирование в данном случае получает только родительский элемент. Придав исходному элементу свойство **float**, мы сделали его плавающим. Но так как исходный плавающий элемент находится в фиксированном родительском, то он наследует свойство фиксации и всегда при прокрутке остается на своем месте.

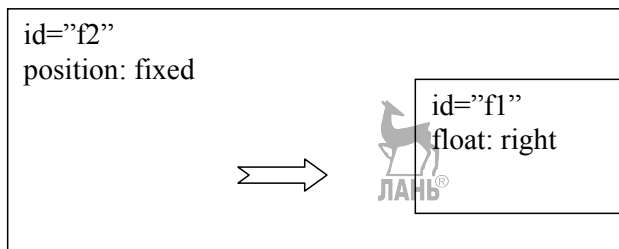
```
<STYLE TYPE = "text/css">  
#f1 {  
    width: 500px; height: 200px;  
    background-color: gold;  
    float: right
```



```

    }
    #f2 {
        width: 100%; height: 200px;
        position: fixed;
        top: 500px
    }
</STYLE>

```



Липкое позиционирование

Элемент с липким позиционированием располагается на странице на основе положения ползунка полосы прокрутки браузера. Липкий элемент переключается между относительным и фиксированным позиционированием, в зависимости от положения прокрутки. Он позиционируется относительно до тех пор, пока заданное в коде положение смещения не будет встречено в окне просмотра браузера, после чего он фиксируется на месте. Необходимо указать по крайней мере одно значение из верхних, правых, нижних или левых отступов родительского элемента.

```

#element {
    position: sticky;
    position: -webkit-sticky;
    top: 0;
    padding: 5px;
    background-color: #cae8ca;
    border: 2px solid #4CAF50;
}

```

Элемент может отлипнуть от фиксированного положения при перемещении ползунка линейки прокрутки в противоположную сторону.

Браузер Internet Explorer, Edge 15 и более ранние версии не поддерживают липкое позиционирование. Для Safari требуется префикс `-webkit-` префикс. Для реализации резервного макета можно использовать запрос функции `@supports (position: sticky)`.

Плавающие элементы

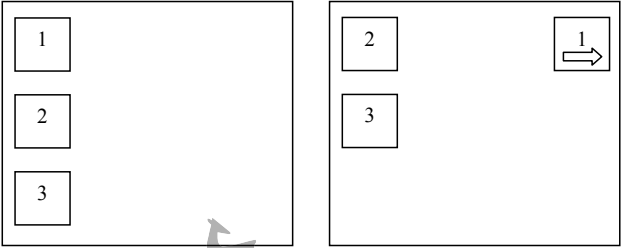
Блочные элементы можно сделать плавающими, придав их свойству **float** значение **left** или **right**. Значение **none** отменяет плавание элемента. Плаваю-

щий элемент не находится в нормальном потоке элементов и, соответственно, блочные контейнеры потока не замечают его.

Таблица 1. Значения свойства **float**

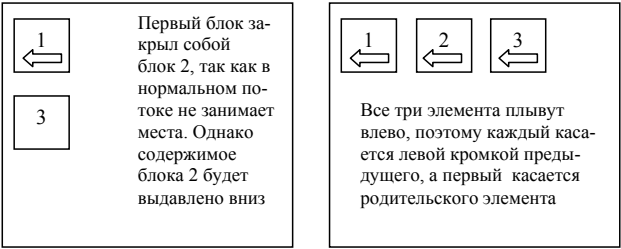
left	элемент выравнивается по левому краю родителя, остальные элементы обтекают его по правой стороне
right	элемент выравнивается по правому краю родителя, остальные элементы обтекают его по левой стороне
none	обтекания элемента не происходит
inherit	значение наследуется от родителя

Плавающий элемент перемещается (плывет) вправо или влево до тех пор, пока его внешний край не коснется края содержащего его контейнера или другого плавающего элемента.



На рисунке выше придали свойству **float** первого блока значение **right**, поэтому он переместился к правому краю контейнера, а блоки 2 и 3 поднялись вверх, так как они остались в нормальном потоке.

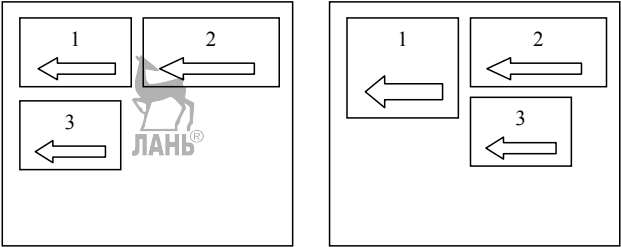
Левый рисунок ниже показывает, что произойдет, если первый контейнер сделать плавающим слева. На правом рисунке продемонстрировано поведение трех плавающих элементов. В отличие от первого не плавающего варианта (левый рисунок выше), они расположились на одной строке.



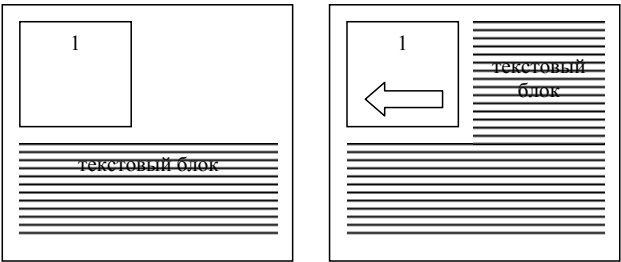
Для того чтобы элементы расположились по горизонтали один за другим, необходимо задать одно и то же значение **float** для следующих друг за другом элементов.



Если суммарная ширина следующих друг за другом элементов превышает ширину родительского контейнера, то последние не уместяющиеся элементы будут перенесены на следующую строку.



Рядом стоящие не плавающие элементы обтекают плавающий элемент слева или справа либо не обтекают его. Плавающие контейнеры специально были разработаны для того, чтобы текст мог обтекать их справа или слева. На нижнем рисунке справа плавающий контейнер 1 обтекает текстовый блок.



Для размещения блоков ниже выровненных по горизонтали необходимо использовать свойство стиля **clear** (от англ. – очистка), явно указывающее на то, что данный блок должен располагаться ниже предшествующих ему контейнеров.

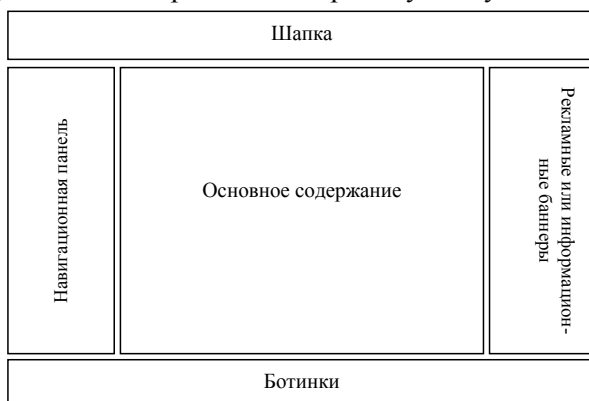
Таблица 2. Значения свойства **clear**

left	элемент будет расположен ниже всех элементов, значение атрибута float у которых равно left
right	элемент будет расположен ниже всех элементов, значение атрибута float у которых равно right
both	элемент будет расположен ниже всех элементов, значение атрибута float у которых равно left или right
none	отмена свойств атрибута clear
inherit	значение наследуется от родителя

Блочная (семантическая) разметка страницы

С помощью блоков и их свойств **float** и **clear** можно разметить страницу, не прибегая к таблицам. Такая блочная разметка отвечает принципам семантической верстки, так как каждый элемент используется по прямому назначению.

Допустим, необходимо разметить страницу следующими блоками:



Сначала создадим блоки, затем присвоим им стили.

```
<BODY>
  <DIV ID="top"> Шапка </DIV>
  <DIV ID="left"> Навигационная панель </DIV>
  <DIV ID="content"> Основное содержание </DIV>
  <DIV ID="right"> Рекламные и информационные баннеры
</DIV>
  <DIV ID="bottom"> Ботинки </DIV>
</BODY>
```

Стили для блоков разместим в заголовочном разделе html-документа. Верхний блок нет необходимости делать плавающим, так как он занимает всю ширину строки. Вторая строка состоит из трех плавающих блоков. Последняя строка состоит из одного блока, но чтобы он находился ниже плавающего сегмента, мы должны ему добавить свойство **clear**.

```
#top {
  height: 200px;
  width: 1812px;
  border: 3px double black
}
#left {
  height: 600px;
  width: 300px;
  border: 3px double black;
  float: left
}
#content {
  height: 600px;
```

```

width: 1200px;
border: 3px double black;
float: left
}
#right {
height: 600px;
width: 300px;
border: 3px double black;
float: left
}
#bottom {
height: 200px; width: 1812px;
border: 3px double black;
clear: left
}

```



Если размер содержимого превышает границы размеров контейнера, на помощь приходит свойство стиля **overflow**.

Таблица 3. Значения свойства **overflow**

visible	отображается весь контент даже за пределами контейнера
hidden	отображается только область внутри контейнера, оставшаяся часть контента скрывается
scroll	добавление полос прокрутки контейнеру, полосы будут отображаться, даже если в них нет необходимости
auto	полосы прокрутки появятся только в случае необходимости

Существует возможность управления отображением содержания контейнера отдельно по горизонтали и вертикали, соответственно при помощи атрибутов **overflow-x** и **overflow-y**, значения которых аналогичны значениям **overflow**.



Вопросы и упражнения

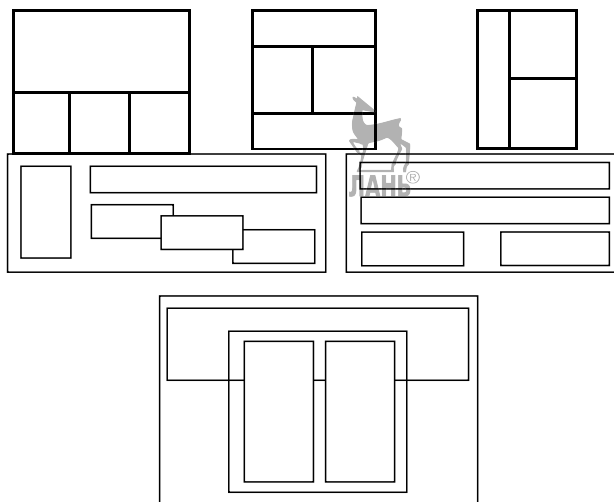
1. Каким образом располагаются блочные и строчные элементы в нормальном потоке?
2. Какое css-свойство может изменить тип html-элемента?
3. При каком типе позиционирования элемент, не покидая нормального потока, смещается в сторону?
4. При каком типе позиционирования элемента его «не замечают» контейнеры нормального потока?

5. Какое css-свойство регулирует порядок расположения перекрывающихся элементов?
6. Какой тип позиционирования оставляет элемент неподвижным при прокрутке содержимого веб-страницы?
7. Какой тип позиционирования позволяет располагать несколько контейнеров на одной строке?
8. Создайте html-документ с контейнером <DIV>, в котором разместите рисунок, имеющий размер, превышающий размер контейнера. С помощью стиля overflow скройте не уместающуюся в контейнер часть рисунка. С помощью псевдокласса hover создайте эффект: при наведении курсора мыши на рисунок, он меняет свой размер таким образом, что помещается целиком в контейнер.



9. Блочная разметка

Разбейте веб-страницу на несколько областей с помощью блоков DIV.



10. Буквица

Буквица – крупная, часто выделенная цветом, орнаментированная или иным образом украшенная начальная буква раздела рукописной или печатной книги (из статьи Википедии). Используя тег и свойства CSS, определите класс элемента, являющегося буквицей, и разместите его на веб-странице.

ЕДИНИЦЫ ИЗМЕРЕНИЯ

В тех случаях, когда значение свойства задается числовой величиной, могут быть использованы разнообразные единицы измерения. Указывать длину в CSS можно в разных единицах. Некоторые из них пришли из типографской сферы, такие как пункт (pt) и пика (pc), другие, как, например, сантиметр (cm) и дюйм (in), знакомы всем в повседневной жизни. Единица измерения пиксель (px) возникла с приходом цифровых мониторов. Единицы измерения не имеют отношения к свойствам, но имеют прямое отношение к средствам отображения: экран или бумага. Свойство со значением в пикселях (margin: 5px) также допускает и значения в дюймах или сантиметрах (margin: 1.2in; margin: 0.5cm), и наоборот.

Единицы длины	Сантиметры	Дюймы	Миллиметры	Пики	Пункты
cm	1	0,3937	10	2,3622	2 833 465
in	2,54	1	25,4	6	72
mm	0,1	0,0363	1	0,2362	2,8346
pc	0,4233	0,166	4,2333	1	12
pt	0,0352	0,0138	0,3527	0,0833	1

px – пиксель;

em – размер шрифта элемента относительно текущего размера шрифта родительского элемента. Например, если размер шрифта родительского элемента 12pt, то 1em=12pt, 2em=24pt;

ex – высота строчной буквы x текущего шрифта элемента.

Единицы времени: **ms** – миллисекунда, **s** – секунда.

Величины, заданные в процентах, обозначаются знаком %.

В CSS3 появились новые единицы измерения:

vw – ширина элементов относительно ширины выюпорта (видимая часть) браузера (100vw – максимальное значение);

vh – высота элементов относительно высоты выюпорта (видимая часть) браузера (100vh – максимальное значение);

rem – от «root em», значение, задающее размер текста относительно размера основного текста веб-страницы. Если свойство font-size установлено для основного текста, то rem единицы связываются с этим свойством. Это значит, например, что 1rem равен размеру текста в 16px (в большинстве браузеров по умолчанию это 16px), если не определено другое значение свойством font-size.

Функция calc() для вычисления значений

С помощью функции calc() можно осуществлять вычисления для определения значений у CSS свойств. При этом допустимо комбинировать единицы измерения. Это означает, что если нам нужно создать сетку, состоящую из четырех колонок, имеющих промежутков в 20px между ними, в каждой строке, то мы можем написать комбинацию процентных и пиксельных

значений, например **width: calc(25% – 20px)** или **width: calc(100% * 1/4 – 20px)**.

Пример. Разместим несколько контейнеров по 4 штуки в одном ряду с отступом 24px в родительском контейнере <section>.

```
<!doctype html>
<html>
<head>
<title>Calc()</title>
<style type="text/css">
.b {
  float: left;
  width: calc(25% - 24px);
  height: 100px;
  margin: 10px;
  border: 2px solid black;
}
section {
  border: 1px solid black;
  width: 55vw;
  height: 400px;
}
</style>
</head>
<body>
<section>
  <div class="b"></div>
  <div class="b"></div>
  <div class="b"></div>
  <div class="b"></div>
  <div class="b"></div>
  <div class="b"></div>
  <div class="b"></div>
</section>
</body>
</html>
```

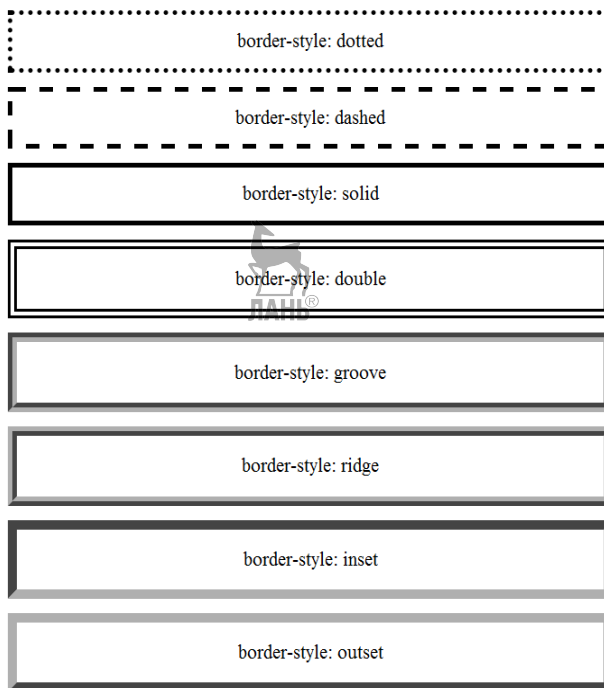


ОФОРМЛЕНИЕ ГРАНИЦ

Любой html-элемент может иметь границу, как уже говорилось в параграфе «Модель контейнеров». Свойства границ получили свое развитие от CSS1 до CSS3 [<http://www.w3.org/TR/2014/WD-css3-background-20140204/>].

В первую и вторую версию спецификации каскадных таблиц стилей вошли самые необходимые простые свойства.

Свойство	Возможные значения	Описание
border-width	thin / medium / thick / число	толщина рамки (тонкая / средняя / широкая / числовое значение)
P {border-width: 3px}		
border-color	название цвета / числовое значение цвета	цвет рамки
H2 {border-color: gold}		
border-style	none / hidden / dotted / dashed / solid / double / groove / ridge / inset / outset	вид линий рамки, (не показана / скрыта / из точек / пунктир / сплошная / двойная / двугранная / такая же, как groove, но светлая и темная границы расположены иначе / одна грань, наклоненная внутрь / такая же, как inset, но свет падает по-другому)
DIV { border-style: solid none solid none }		
border-top border-bottom border-left border-right	<width> <style> <color>	одновременное задание значений свойств width, style и color отдельно для верхней, нижней, левой и правой границы
TD {border-top: thin solid blue}		
border	<border-width> <border-style> <border-color>	одновременное задание значений свойств border-width, border-style и border-color
TABLE {border: medium double red}		



В организации W3C находится рабочий проект CSS3, в котором добавлено много новых очень интересных возможностей оформления рамок.

Рассмотрим первое свойство из совокупности, которое задает изображение в качестве шаблона для рамки. Если **border-image-source** имеет значение none или картинка не может быть отображена, к границе применяется стиль, заданный свойством border-style.



Совокупность рассматриваемых свойств применяется ко всем элементам, кроме табличных элементов, для которых свойство border-collapse имеет значение collapse.



Таблица 4. Значения свойства border-image-source

none	указывает на отсутствие фонового изображения
<url>	путь к файлу-картинке

border-image-source: url(border.png)

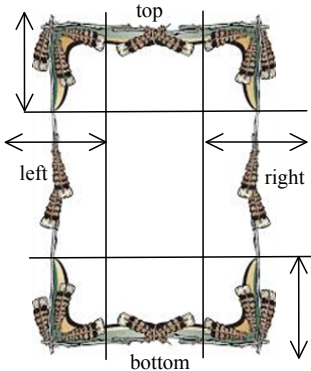
Одного свойства **border-image-source** еще не достаточно для создания красивой границы. Следующее важное свойство в совокупности **border-image-slice**. Оно задает отступы сверху, справа, снизу и слева, которые надо совершить для разделения граничного изображения на части. В результате разделения имеем 9 фрагментов граничного изображения: 4 угла (вставляются без изменений), фрагменты для верхней, правой, нижней и левой границы элемента (могут

растягиваться или повторяться, в зависимости от заданных стилей и размеров блока) и центр картинке, который по умолчанию считается прозрачным.

border-image-slice: 30 50

Таблица 5. Значения свойства **border-image-slice**

число	количество пикселей, которое надо отступить от края, чисел может быть от 1 до 4 (отступ сверху, справа, снизу и слева)
число%	значение отступа в процентах
fill	добавочное значение, означающее, что надо сохранить среднюю часть рисунка (по умолчанию она отбрасывается, то есть считается пустой)



Следующее, самое интересное, свойство **border-image-repeat** из рассматриваемой совокупности определяет, каким образом отображать боковые фрагменты изображения границы по оси x и по оси y. Если задано одно значение, оно распространяется на обе оси.

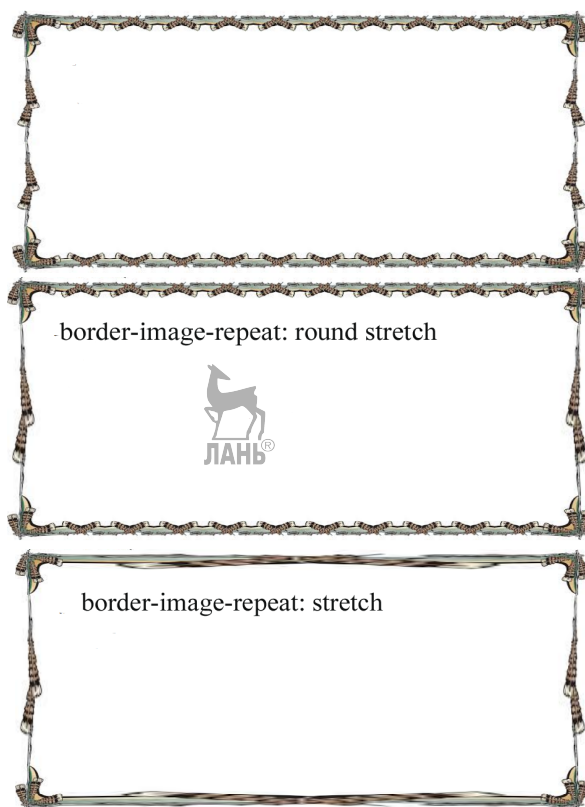
Таблица 6. Значения свойства **border-image-repeat**

stretch	фрагмент растягивается для заполнения области (по умолчанию)
repeat	фрагмент повторяется для заполнения области
round	фрагмент повторяется для заполнения области и масштабируется таким образом, чтобы количество фрагментов было целым

На первом рисунке ниже показана работа значения **round** для всех четырех сторон исходного изображения.

```
#r2 {
width: 800px; height: 300px;
border-image-width: 80px;
border-image-source: url(border.png);
border-image-slice: 77 60;
border-image-repeat: round;
}
```





Свойство **border-image-width** определяет сверху, справа, снизу и слева ширину граничного изображения. Если задано только три значения (ширина сверху, справа и снизу), то четвертое значение (ширина слева) будет равно второму (ширине справа). Если задано только два значения, то третье значение будет равно первому, а четвертое второму. Если задано одно значение, оно распространяется на все четыре стороны.

Таблица 7. Значения свойства **border-image-width**

число	точное значение ширины (не может быть отрицательным)
число%	значение ширины в процентах
число	множитель, на который умножается значение ширины границы, заданное свойством border-width
auto	в качестве значения ширины принимается значение соответствующего отступа, указанного свойством border-image-slice

Последнее свойство из рассматриваемой совокупности свойств оформления графической границы – **border-image-outset**. Оно указывает, на какое расстояние граничное изображение сверху, справа, снизу и слева выступает за размер контента.

Пример: **border-image-outset: 8px**

Таблица 8. Значения свойства **border-image-outset**

положительное число px	значение выступа за размер контейнера
целое или дробное число	множитель выступа за размер контейнера

Существует короткий вариант для всей описанной совокупности свойств граничного изображения – **border-image**. Значения первых трех свойств обязательны, остальные по желанию.

border-image: *border-image-source* *border-image-slice* *border-image-repeat* [*border-image-width* / *border-image-width* / *border-image-outset*]

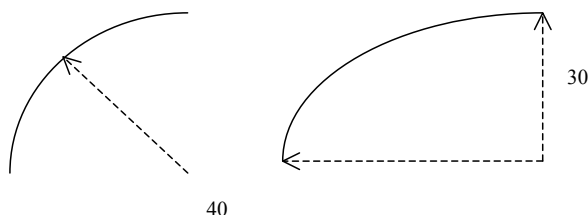
Пример: **border-image: url(border.png) 27 round stretch;**

Отдельное свойство для оформления границ элементов – **border-radius**. Оно устанавливает радиус скругления углов граничной рамки. Скругление также происходит и с фоном.

Таблица 9. Значения свойства **border-radius**

число число число число	радиус дается всем четырем углам, начиная с левого верхнего и далее по часовой стрелке
border-radius: 50px 0 0 50px	
число число	первое значение задает радиус верхнего левого и нижнего правого угла, второе значение – верхнего правого и нижнего левого угла
число число число	первое значение задает радиус для верхнего левого угла, второе – одновременно для верхнего правого и нижнего левого, а третье – для нижнего правого угла
число	задает одинаковый радиус всем четырем углам
число/число	первое значение задает радиус по горизонтали, а второе по вертикали (эллиптические углы)
border-radius: 40px/30px	

Пример: **TD {border: 1px solid red; border-radius: 40px/30px}**



Вопросы и упражнения

1. Используя **CSS**, создайте пунктирную цветную границу у отдельных ячеек таблицы.
2. Используя **CSS**, создайте главную страницу сайта, обрамленную красивой рамкой по периметру экрана (стиль `groove` примените к тегу `<BODY>`).
3. Разместите картинку на веб-странице со скругленными углами, рисунок, заключенный в графическую рамку, и создайте блок с несимметричными закруглениями границы, похожими на лимон.



4. Разместите на веб-странице блок **DIV** со скругленными углами и поместите в него фотографию кого-либо или чего-либо. Не помещающиеся в лок части фотографии скройте соответствующим **CSS**-свойством. Создайте эффект увеличения масштаба фотографии при наведении на нее курсора мыши.



ФОНОВЫЕ ФАНТАЗИИ

Каждый **html**-элемент может иметь не только границу, как уже говорилось в предыдущем параграфе, но и фон. Свойства фонов, так же как и свойства границ, получили свое развитие в таблицах стилей от **CSS1** до **CSS3** [<http://www.w3.org/TR/2014/WD-css3-background-20140204/>].

В первую и вторую версию спецификации каскадных таблиц стилей вошли самые необходимые простые свойства.

Свойство	Возможные значения	Описание
background-color	название цвета числовое значение цвета transparent	цвет фона, transparent – прозрачный фон
H2 {background-color: tomato}		
background-image	url(<имя файла>)	фоновый рисунок обои
DIV {background-image: url(bgimage.jpg)}		

Свойство	Возможные значения	Описание
background-repeat	repeat repeat-x repeat-y no-repeat	<p>◇ repeat – заполнять все пространство элемента;</p> <p>◇ repeat-x – заполнить только первый горизонтальный ряд;</p> <p>◇ repeat-y – заполнить только первый вертикальный столбец;</p> <p>◇ no-repeat – не заполнять.</p>
BODY {background-image: url(kvadrat.gif); background-repeat: repeat-y }		
background-attachment	scroll fixed	<p>свойство определяет, будет ли фон прокручиваться вместе с содержимым документа:</p> <p>scroll – прокрутка разрешена;</p> <p>fixed – прокрутка запрещена</p>
BODY {background-image: url(kvadrat.gif); background-repeat: repeat-y; background-attachment: fixed }		
background-position	0% 0% 0px 0px left top left center left bottom right top right center right bottom center top center center center bottom	<p>положение фонового рисунка задается в любых единицах измерения длины;</p> <p>если указана одна текстовая величина, она интерпретируется как значение отступа по горизонтали слева, а значение по вертикали принимается равным center</p>
TABLE {background-image: url (banner.gif); background-position: top center }		
background	<background-color> <background-image> <background-repeat> <background-attachment> <background-position>	короткий вариант для всей описанной совокупности свойств фона
P {background: url(sphere.gif) gray 50% repeat fixed}		

В рабочем проекте CSS3, который претендует на стандарт, утверждаемый организацией W3C, добавлено много новых очень интересных возможностей фоновое оформления.

Третья спецификация каскадных таблиц стилей включает новое свойство **background-size**, позволяющее задавать размеры фоновому рисунку и масштабировать его. Это важно не для фоновых обоев, а для изображений большого размера типа картин.

`background-size: ширина высота | auto | cover | contain`

Если задана только ширина, высота по умолчанию будет иметь значение `auto`.

Таблица 10. Значения свойства **background-size**

число число	задает размер ширины и высоты в любых доступных для CSS единицах – пикселы (px), сантиметры (cm), em и так далее
background-size: 100px 100px	
число% число%	задает размер фоновой картинки в процентах от ширины и высоты родительского элемента
background-size: 50%	
auto	размеры фонового изображения не масштабируются
cover	масштабирует фоновое изображение таким образом, чтобы полностью покрыть область фона элемента. Некоторые части фонового рисунка при этом могут быть не видны в области элемента
background-size: cover	
contain	масштабирует изображение по максимуму так, чтобы и ширина, и высота поместились внутри контейнера

Следующее новое css3-свойство **background-origin** определяет начало позиционирования фонового рисунка. Это свойство не работает в том случае, когда значением свойства `background-attachment` является `fixed`.

`background-origin: padding-box | border-box | content-box`

Таблица 11. Значения свойства **background-origin**

padding-box	фон позиционируется относительно внутренней пустой полосы элемента
background-origin: padding-box	
border-box	фон позиционируется относительно границ элемента, при этом линия границы может перекрывать фоновое изображение

background-origin: border-box	
content-box	фон позиционируется относительно содержимого элемента



Следующее новое css3-свойство **background-clip** очень похоже на предыдущее свойство, но применяется к цвету фона, то есть отвечает за область распространения неграфического фона. Значения этого свойства в точности совпадают со значениями свойства background-origin.

background-clip: *padding-box* | *border-box* | *content-box*

Таблица 12. Значения свойства **background-clip**

padding-box	цвет фона позиционируется относительно внутренней пустой полосы элемента
background-clip: padding-box	
border-box	цвет фона позиционируется относительно границ элемента, при этом линия границы перекрывает фон
background-clip: border-box	
content-box	фон позиционируется относительно содержимого элемента

Еще одно новое css3-свойство **opacity** задает уровень прозрачности элемента веб-страницы. При частичной или полной прозрачности через элемент проступает фоновый рисунок или другие элементы, расположенные ниже полупрозрачного объекта. В качестве значения выступает число из диапазона от 0.0 до 1.0. Значение 0 соответствует полной прозрачности элемента, а 1 – полной непрозрачности. Дробные числа вида 0.6 устанавливают полупрозрачность.

Например, **opacity: 0.6**

Градиентная заливка фона элемента или веб-страницы

Новый проект спецификации добавляет еще несколько значений в свойство **background**, которые позволяют задавать фону градиентную заливку.

Это значение **linear-gradient** и **radial-gradient**, **repeating-linear-gradient** и **repeating-radial-gradient**. Названия точно передают смысл. Формат поддерживает линейный и радиальный градиент.





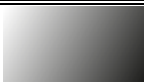

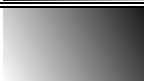


Градиент – это плавный переход цвета от светлых тонов к темным или наоборот, при использовании нескольких цветов – это плавный переход от одного цвета к другому.

Для задания линейного градиента используем первое значение, общий вид которого:

linear-gradient(направление, начальный цвет, конечный цвет)

linear-gradient(направление, цвет1, цвет2, цвет3, ...)

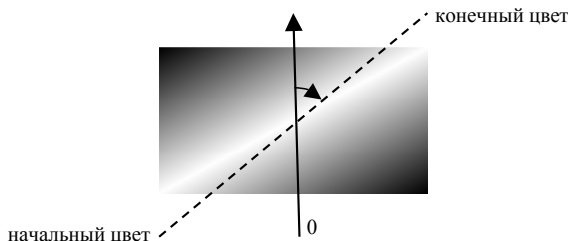
По умолчанию направление имеет значение **top to bottom**, то есть сверху вниз.

Направление	Описание	Вид
to bottom	сверху вниз	
background: linear-gradient(black, white)		
to top	снизу вверх	
to left	справа налево	
background: linear-gradient(to left, black, white)		
to right	слева направо	
to top left	от правого нижнего угла к левому верхнему	
background: linear-gradient(to top left, black, white)		
to top right	от левого нижнего угла к правому верхнему	
to bottom left	от правого верхнего угла к левому нижнему	
to bottom right	от левого верхнего угла к правому нижнему	
linear-gradient(to top left, black, white, black)		

Если predetermined направлений не хватает для нужного варианта градиента, то можно использовать числовое значение угла вместо направления.

linear-gradient(угол, начальный цвет, конечный цвет)

linear-gradient(угол, цвет1, цвет2, цвет3, ...)



Пример. `background: linear-gradient(180deg, red, blue)`

Возможны эксперименты с переходами в **прозрачность**. Так, например, можно создавать градиентную прозрачность, то есть плавный переход от непрозрачности к прозрачности. Задать прозрачность в цвете можно через функцию `rgba`, последний аргумент которой отвечает за прозрачность, варьирующуюся от 1 (полная непрозрачность) к 0 (полная прозрачность).

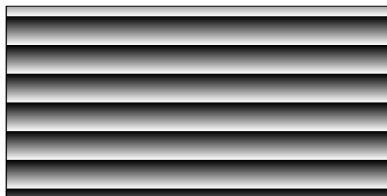
`linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1))`

Наконец, есть еще одно значение-функция, позволяющая генерировать повторяющийся линейный градиент.

repeating-linear-gradient(цвет1 [%ширины], цвет2 [%ширины], ...)

Пример. **Жалюзи**

`background: repeating-linear-gradient(black 5%, white 20%)`



Радиальный градиент, в отличие от линейного, задает направление плавного перехода не по прямой линии, а в виде круговой волны. Спецификация поддерживает две формы радиального градиента – круг (`circle`) и эллипс (`ellipse`). По умолчанию устанавливается эллиптический градиент. Начальная точка распространения волны по умолчанию – центр объекта. Общий вид функции:

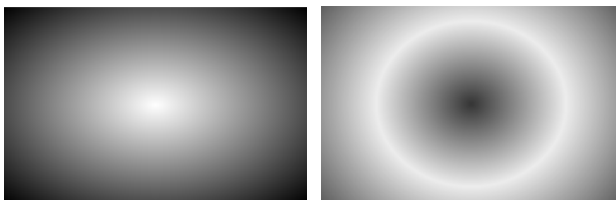
radial-gradient([форма размер центр,] цвет1, цвет2,..., цветN)

Примеры.

`radial-gradient(white, black)`

`radial-gradient(circle, red, yellow, green)`

radial-gradient(red 5%, green 15%, blue 60%)
 radial-gradient(at 60% 55%, red, yellow, green)
 radial-gradient(circle at 60% 55%, red, yellow, green)



Параметр «размер» может принимать четыре различных значения. По умолчанию работает значение **farthest-corner**.

Размер	Описание	Вид
closest-side	градиент распространяется до ближайшей стороны	
radial-gradient(circle closest-side at 130px 50px, #000, #fff)		
farthest-side	градиент распространяется до дальней стороны элемента	
radial-gradient(circle farthest-side at 130px 50px, #000, #fff)		
closest-corner	ближайший угол	
radial-gradient(circle closest-corner at 50px 50px, #000, #fff)		
farthest-corner	отдаленный угол	

Существует прием для создания чередующихся цветовых заливок фона с резким переходом цвета.

radial-gradient([форма размер центр,]цвет1 размер1, цвет2 размер2, ..., цветN размерN)

Резкие переходы получаются, когда размер одного цвета почти или полностью совпадает с размером другого. Например:

radial-gradient(circle, #000 10% ,#fff 11% ,#fff 20% ,#000 21% ,#000 30% ,#fff 31% ,#fff 40% ,#000 41% ,#000 50% ,#fff 51% ,#fff 60% ,#000 61% ,#000 70% ,#fff 71%)



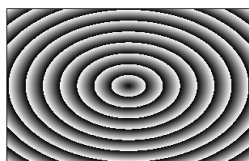
Разница в один пиксел для того, чтобы сгладить переход и избежать лестничный эффект.

Так же как и для линейного градиента, существует значение-функция, позволяющая генерировать повторяющийся радиальный градиент.

repeating-radial-gradient(цвет1 [%ширины], цвет2 [%ширины], ...)

Пример.

background: repeating-radial-gradient(#000 10% ,#fff 20%)



Для быстрого и наглядного создания градиентных заливок на основе CSS стандарта во Всемирной паутине находится множество бесплатных различных сервисов. Например **Ultimate CSS Gradient Generator** <http://www.colorzilla.com/gradient-editor/>. Данный редактор в режиме онлайн позволяет подобрать желаемый градиент и скопировать сгенерированный код градиента в свою веб-страницу.

Смешанный фон

Свойство **background-blend-mode** позволяет смешивать фоновые слои. Создавать слои фона можно известным свойством **background-image**, в котором каждый слой перечисляется через запятую. В качестве слоя может выступать градиентная заливка или фоновый рисунок.

background-image: url(bricks-459299_1920.jpg), linear-gradient(45deg, #f8a261, #6e9fc9);

Таблица 13. Значения свойства **background-blend-mode**

normal	отсутствие смешивания
overlay	перекрывтие. Используется комбинация режима наложения screen для более светлых пикселей и режим наложения multiply для темных пикселей. Полутона (50% серого) становятся прозрачными
darken	затемнение. Сохраняются темные тона всех слоев

multiply	умножение. Умножаются уровни яркости пикселей верхнего слоя с пикселями нижележащих слоев. Используется в основном для создания теней и удаления белых и светлых тонов. Умножение любого цвета на черный дает черный. Умножение любого цвета на белый сохраняет исходный цвет
lighten	осветление. Темные пиксели верхнего слоя заменяются на светлые пиксели нижележащих слоев. Светлые пиксели не заменяются
screen	экран. Удаляются темные пиксели, создавая плавный переход.
color-dodge	осветление основы. Насыщает светлые тона и блики. Темные области изображения остаются без изменений
saturation	насыщенность. Создается цвет с насыщенностью исходного цвета и оттенком и яркостью нижележащего цветового фона. Серые оттенки остаются без изменений
color	цвет. Создается цвет с оттенком и насыщенностью исходного цвета и яркость нижележащего цветового фона. Сохраняет серые тона нижележащих слоев
luminosity	яркость. Генерируется цвет с яркостью исходного цвета и оттенком и насыщенностью нижележащего фона. Эффект противоположен режиму color

Свойство **background-blend-mode** делает верхний слой полупрозрачным, что позволяет показывать сквозь него нижележащие слои. Если задано больше двух фоновых слоев, то для общего фона можно задать несколько режимов смешивания, перечислив их через запятую.


background-blend-mode: overlay, color;


Дополнительную информацию можно найти в спецификации CSS по адресу <https://www.w3.org/TR/compositing-1/#blending>.

Вопросы и упражнения

1. Используя **CSS**, создайте фон веб-страницы из графического рисунка и зафиксируйте его так, чтобы во время скроллинга он оставался неподвижным.
2. Найдите изображение высокого разрешения и сделайте его фоновым для тела веб-страницы, масштабируйте его стилями таким образом, чтобы он без повторений заполнил всю площадь.
3. Разместите на веб-странице с графическим фоном блок (**<DIV>**) текста, фон которого сделайте полупрозрачным. Цвет фона блока текста должен контрастировать с фоновым рисунком страницы.

4. Разместите на веб-странице интерактивную кнопку (тег <BUTTON>). Измените фон кнопки на градиентную заливку. Добавьте эффект изменения направления градиента при наведении курсора мыши на кнопку и эффект изменения цвета градиента при щелчке мыши на кнопке. Измените форму кнопки с прямоугольной на круглую и добавьте широкую граничную линию.
5. **Радуга.** Придайте некоторым элементам, размещенным на веб-странице, заливку цветом радуги.
6. **Внешняя гиперссылка.** Используя свойства CSS, создайте ссылки на внешние веб-страницы с графическим значком на конце. Рисунок можно оформить как фоновый для гиперссылки с размещением его в правом конце области без размножения.

Домашняя страница Иванова Андрея 

Домашняя страница Петровой Дарьи 

7. **Стилизация гиперссылок.** Используя свойства позиционирования фонового рисунка, создайте графические маркеры у пунктов списка.



8. **Динамический фон.** Создайте эффект динамического фона на веб-странице при прокрутке (используйте свойство фиксации фона в каждом блоке). При прокрутке страницы получается эффект перехода одного фона в другой.
9. **Видеофон.** Создайте на всю видимую часть окна браузера блок, в который поместите фоновое видео (бесплатно скачайте с сервиса Vimeo), поверх видео разместите блок с текстовой информацией. Для лучшего просмотра фоновый цвет блока текста сделайте полупрозрачным.

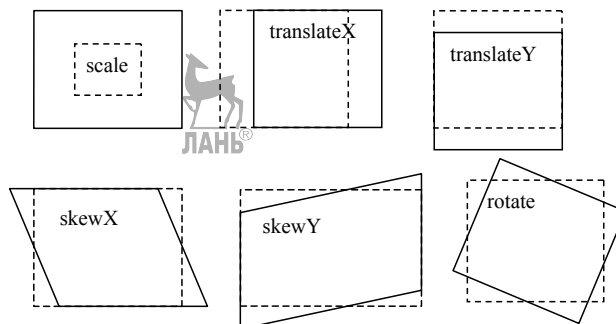
ТРАНСФОРМАЦИЯ ЭЛЕМЕНТОВ

В css3-спецификацию входит свойство, значения которого при применении позволяют масштабировать элемент или сдвигать его, наклонять или поворачивать. При этом не происходит сдвига окружающих его элементов.

`transform: scale | skew | translate | rotate | matrix`

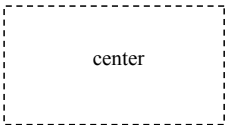
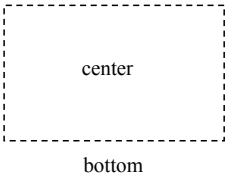
Значение	Описание
<code>scale(число, [число])</code>	значение больше 1 увеличивает масштаб элемента, меньше 1 – уменьшает масштаб
<code>scaleX(число)</code>	увеличивает или уменьшает масштаб по горизонтали
<code>scaleY(число)</code>	увеличивает или уменьшает масштаб по вертикали

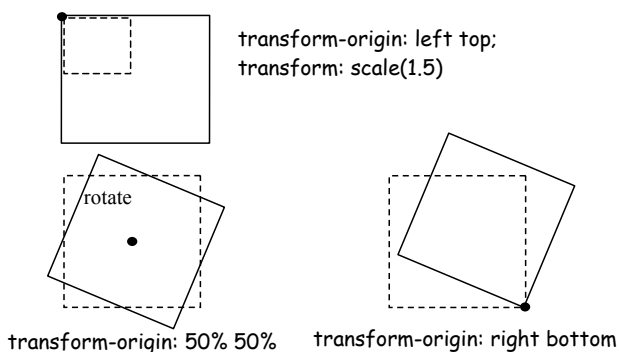
Значение	Описание
transform: scale(1.5)	
skew()	наклоняет элемент на заданный угол по горизонтали и (или) вертикали
skewX()	наклоняет элемент на заданный угол по горизонтали
skewY()	наклоняет элемент на заданный угол по вертикали
transform: skewX(25grad)	
translate(число, [число])	сдвигает элемент на заданное значение по горизонтали и (или) вертикали
translateY(число)	сдвигает элемент на заданное значение по вертикали
translateX(число)	сдвигает элемент на заданное значение по горизонтали
transform: translateX(150px)	
rotate()	поворачивает элемент на заданный угол относительно точки трансформации, задаваемой свойством transform-origin
rotateX()	поворачивает элемент на заданный угол вдоль оси X
rotateY()	поворачивает элемент на заданный угол вдоль оси Y
transform: rotate(-15deg)	
matrix (число1, число2, число3, число4, число5, число6)	задает матрицу 2D преобразований



Вместе со свойством transform используется CSS3-свойство **transform-origin**, которое определяет координаты точки, относительно которой будет происходить трансформация элемента.

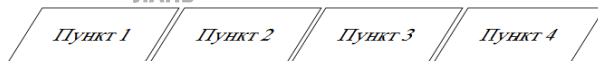
transform-origin: *x-координата* *y-координата*

Значение		Описание	
x-координата	left		
	center		
	right		
	число		
	число%		
y-координата	top		
	center		
	bottom		
	число		
	число%		
50% 50%		значение по умолчанию	
transform-origin: 75% 0%			



Вопросы и упражнения

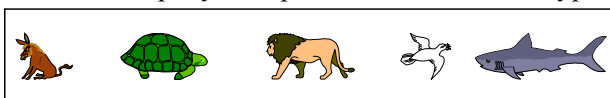
1. **Скошенное меню.** Создайте список ссылок в скошенных рамках.



2. **Увеличение размеров.** На веб-странице с фотогалереей добавьте стиль увеличения масштаба изображений в 2 раза при наведении на них курсора мыши.
3. **Разбросанные конверты.** Используя свойства абсолютного и относительного позиционирования, создайте графическую композицию «разбросанные конверты».



4. **Динамическое меню.** Создайте блок картинок и примените стиль переворачивания картинки на 180 градусов при наведении на нее курсора мыши.



5. **Прыгающий мяч.** Создайте с помощью элемента div и стилей прыгающий вертикально мяч, который при приземлении немного сжимается, при движении вверх немного растягивается.

3D-ТРАНСФОРМАЦИЯ ЭЛЕМЕНТОВ

Технология CSS позволяет производить над html-элементами 3D- (в трех измерениях) трансформацию. Для этого можно использовать ряд свойств, представленных в следующей таблице.

Свойство	Описание
transform	применяет 2D- и 3D-трансформацию к элементу
transform-origin	изменяет позицию, относительно которой будет происходить трансформация элемента
transform-style	указывает, как вложенные элементы отображаются в трехмерном пространстве
perspective	определяет перспективу просмотра 3D-элементов
perspective-origin	определяет нижнее положение трехмерных элементов
backface-visibility	определяет, должен ли элемент быть видимым, если он не обращен к экрану

Свойство `transform: rotateZ()` задает поворот элемента на заданное количество градусов относительно оси Z в трехмерном пространстве. Поворот осуществляется вокруг точки, задаваемой свойством `transform-origin`. Значение по умолчанию `transform-origin :50% 50% 0`. Это значит, что для оси Z по умолчанию значение 0. Координата по оси Z может задаваться только в любых корректных единицах для измерения длины (исключая проценты).

Свойство `transform-style` определяет, как дочерние элементы будут отображаться в трехмерном пространстве. Это свойство должно использоваться совместно со свойством `transform`. Значение по умолчанию `flat` не сохраняет 3D позицию дочерних элементов, значение `preserve-3d` – сохраняет.

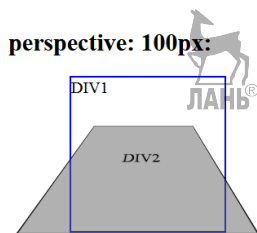
Свойство `perspective` (перспектива) предназначено, чтобы дать 3D-позиционируемому элементу некоторую перспективу. Свойство перспективы

определяет, насколько далеко объект находится от наблюдателя. Более низкое значение приводит к более интенсивному 3D-эффекту, чем более высокое значение. Свойство перспективы должно быть задано у родительского контейнера того элемента, который получит перспективу. Свойство `perspective-origin` определяет, в какой позиции наблюдатель смотрит на 3D-объект.

П р и м е р.

```
<!DOCTYPE html>
<html>
<head>
<style>
#div1 {
    position: relative;
    height: 150px;
    width: 150px;
    margin-left: 60px;
    border: 1px solid blue;
    perspective: 100px;
}
#div2 {
    padding: 50px;
    position: absolute;
    border: 1px solid black;
    background-color: red;
    background: rgba(100,100,100,0.5);
    transform-style: preserve-3d;
    transform: rotateX(45deg);
}
</style>
</head>
<body>
  <h1>Свойство perspective</h1>
  <h2>perspective: 100px:</h2>
  <div id="div1">DIV1
    <div id="div2">DIV2</div>
  </div>
</body>
</html>
```



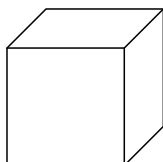


(Пример взят с сайта https://www.w3schools.com/cssref/css3_pr_perspective.asp.)

В CSS 3 существует свойство **backface-visibility**, которое задает видимость обратной стороны элемента при повороте. По умолчанию, даже если элемент повернут к нам обратной стороной – она будет видна (visible по умолчанию), однако это поведение можно изменить значением **hidden** данного свойства.

Вопросы и упражнения

1. **Куб.** Создайте стилизованный куб, используя 3D-свойства таблиц стилей.



2. **Обратная сторона.** Создайте эффект, когда при наведении курсора на фотографию появляется эффект ее разворота и отображения обратной стороны с текстом.

ОТ ПЛОСКОГО ТЕКСТА К 3D

Для текстовой информации существует несколько групп свойств: свойства текста как единого блока, свойства отдельных элементов текста и свойства шрифта.

Свойство	Возможные значения	Описание
text-indent	число число%	отступ первой строки абзаца
P {text-indent: -3em}		
text-align	left right center justify	выравнивание по горизонтали (по левому краю по правому краю по центру по ширине)
H1 {text-align: center}		
text-transform	capitalize uppercase lowercase none	<ul style="list-style-type: none"> ◇ capitalize – преобразует первую букву каждого слова в прописную ◇ uppercase – преобразует все буквы текста в прописные ◇ lowercase – преобразует все буквы текста в строчные

Свойство	Возможные значения	Описание
text-transform: capitalize		
text-decoration	none underline overline line-through blink	<ul style="list-style-type: none"> ◇ underline – подчеркивание текста; ◇ overline – черта сверху; ◇ line-through – перечеркивание; ◇ blink – мигание (поддерживается не всеми браузерами)
text-decoration: underline		
text-shadow	х-отступ у-отступ [размытость] [цвет]	<p>создание тени</p> <ul style="list-style-type: none"> ◇ х-отступ – смещение тени по горизонтали; ◇ у-отступ – смещение тени по вертикали; ◇ размытость – ширина распыления контура тени; ◇ цвет – цвет тени
text-shadow	х-отступ у-отступ [размытость] [цвет]	допускается указывать несколько параметров тени, разделяя их между собой запятой
text-shadow: 5px 5px 10px red text-shadow: -1px -1px 0 red, -7px -7px 6px rgba(0,0,0,0.6)		

В Интернете существует масса генераторов 3D текста. С помощью кнопочек и ползунков можно быстро добиться нужного варианта оформления, а программа сгенерирует CSS для копирования и вставки в свой html-документ.

3D CSS Текст

Если изменить цвет текста на прозрачный (transparent), а цвет тени оставить непрозрачным и ненулевым параметр размытости, то получится эффект размытого шрифта.

Следующая группа css-свойств относится к структурным элементам текста, таким как буквы и строки.

Свойство	Возможные значения	Описание
letter-spacing	число normal	определение межбуквенного расстояния

Свойство	Возможные значения	Описание
letter-spacing: -3px		
word-spacing	число normal	определение расстояния между словами, число может быть и отрицательным
word-spacing: 30px		
white-space	normal nowrap pre pre-line pre-wrap	<p>определение обработки браузером пробелов между элементами текста</p> <ul style="list-style-type: none"> ◇ normal – удаление лишних пробелов; ◇ nowrap – запрет на автоматический разрыв строк; ◇ pre – пробелы остаются без изменения (по аналогии с тегом PRE); ◇ pre-line – удаление лишних пробелов, текст автоматически переносится на следующую строку, если он не помещается в заданную область; ◇ pre-wrap – пробелы остаются без изменения, текст автоматически переносится на следующую строку
white-space: pre		
line-height	normal число число%	определение расстояния между базовыми линиями соседних строк текста в px, pt, cm, и т. д.
line-height: 150%		
vertical-align		<p>выравнивание по вертикали</p> <ul style="list-style-type: none"> ◇ baseline – выравнивание базовой линии элемента с базовой линией родителя; ◇ число – повышает или понижает элемент на заданное расстояние, возможно отрицательное значение; ◇ sub – нижний индекс; ◇ super – верхний индекс; ◇ top – выравнивается по вершине самого высокого элемента в строке; ◇ text-top – выравнивается по вершине самого высокого текстового элемента в строке; ◇ middle – выравнивается по середине родительского элемента

Свойство	Возможные значения	Описание
vertical-align	baseline число sub super top text-top middle bottom text-bottom	<p>◇ bottom – выравнивается по низу самого низкого элемента в строке;</p> <p>◇ text-bottom – выравнивается по низу самого низкого текстового элемента в строке</p> 
vertical-align: middle		

Последняя группа css-свойств текста относится к шрифтам. Шрифт – это графический рисунок начертаний букв и знаков, составляющих единую стилистическую и композиционную систему. Каждый шрифт имеет свое уникальное имя и набор свойств, к которым относятся размер, начертание и тому подобное.

Браузеры в зависимости от операционной системы поддерживают целый набор различных шрифтов. Но так как операционных систем и их версий существует много, то, соответственно, и наборы шрифтов отличаются. В таблице указаны стандартные шрифты для любых браузеров и операционных систем.

Windows шрифты	Образец шрифта
Arial	Образец шрифта
Arial Black	Образец шрифта
Book Antiqua	Образец шрифта
Comic Sans MS	Образец шрифта
Courier New	Образец шрифта
Georgia	Образец шрифта
Impact	Образец шрифта
Lucida Console	Образец шрифта
Lucida Sans Unicode	Образец шрифта
Palatino Linotype	Образец шрифта
Tahoma	Образец шрифта
Times New Roman	Образец шрифта
Trebuchet MS	Образец шрифта
Verdana	Образец шрифта

Если веб-разработчик не указывает шрифт в html-коде, то браузер использует шрифт по умолчанию. Для явного назначения шрифта используется css-свойство font-family:

font-family: *название шрифта1, название шрифта2, ...*






font-family: Georgia

font-family: "Times New Roman", Georgia, Serif

Шрифтов можно указать несколько через запятую в порядке значимости. Браузер будет пытаться применить первый в списке шрифт, при отсутствии его – второй и так далее. Заканчивать список лучше названием обобщенного шрифта, которое задает вид начертания – serif, sans-serif, cursive, fantasy или monospace.

Универсальные семейства шрифтов:

- serif – шрифты с засечками типа Times и Georgia;
- sans-serif – рубленые шрифты (sans означает «без», шрифты без засечек) типа Arial и Verdana;
- cursive – курсивные шрифты, имитирующие почерк (Comic Sans MS);
- fantasy – декоративные шрифты для заглавий (Impact);
- monospace – моноширинные шрифты, ширина каждого символа в таком семействе одинакова (шрифт Courier, Lucida Console).

				
serif	sans-serif	cursive	fantasy	monospace

Если среди стандартных шрифтов не оказалось подходящего для дизайна страницы, то в CSS существует правило **@font-face**, с помощью которого можно добавлять новые шрифты.

@font-face {*свойства шрифта*}

@font-face {font-family: *шрифт*; src: url(*шрифт.ttf*)}

Правило @font-face позволяет определить параметры шрифтов, а главное – загрузить специфичный шрифт на локальный компьютер пользователя.

@font-face {

font-family: NewFont; /* Имя шрифта */

src: url(NewFont.ttf); /* Путь к файлу со шрифтом */

}

P {font-family: NewFont}

Подобрать оригинальный шрифт и внедрить его в свой html-документ можно, например, и с сайта <http://webfont.ru/>. Для этого достаточно для выбранного шрифта в раздел стилей скопировать с сайта строку со словом @import <имя шрифта>, затем в свойство font-family добавить имя импортированного шрифта.

Для изменения свойств выбранного шрифта существует целый ряд css-свойств.

Свойство	Возможные значения	Описание
font-style	normal italic oblique	определение стиля шрифта: нормальный или наклоненный (италик)
font-style: italic		
font-variant	normal small-caps	изменение всех строчных символов на прописные уменьшенного размера
font-variant: small-caps		
font-weight	normal bold bolder lighter число от 100 до 900 через 100	определение насыщенности шрифта ◇ нормальное начертание шрифта (которое установлено по умолчанию) эквивалентно 400, ◇ стандартный полужирный текст – значению 700, ◇ значения bolder и lighter изменяют жирность относительно насыщенности родителя, соответственно в большую и меньшую сторону
font-weight: 900		
font-size	число xx-small x-small small medium large x-large xx-large larger smaller	изменение размера шрифта: ◇ для задания абсолютного размера используются следующие значения: xx-small, x-small, small, medium, large, x-large, xx-large, ◇ значения larger и smaller устанавливают относительные размеры шрифта, ◇ число в любых допустимых единицах CSS: em, ex, pt, px, % и др.
font-size: 24pt		

Свойство	Возможные значения	Описание
font	<i>font-style font-variant font-weight font-size/line-height</i> caption icon	короткий вариант для всей совокупности свойств шрифта: ◇ caption – для текста элементов форм, ◇ icon – для текста под иконками, ◇ menu – для выпадающего меню, message-box – для диалогового окна,
	menu message-box small-caption status-bar	◇ small-caption – для подписей к небольшим элементам управления, ◇ status-bar – для статусной строки
font: 15px arial, sans-serif; font: italic bold 12px/30px Georgia, serif;		

Вопросы и упражнения

1. **3D-заголовок.** Используя интернет-генератор 3D-текста, выберите подходящий стиль для заголовка вашей веб-страницы и примените его с последующей локальной адаптацией свойств.
2. **Размытое меню.** Используя свойства тени, создайте стилизованное меню с эффектом размытого шрифта.
3. **Сторонний шрифт.** Разместите на веб-странице заголовок и примените к нему нестандартный шрифт, взятый со стороннего веб-сайта.

СТИЛИЗАЦИЯ СПИСКОВ И ТАБЛИЦ

Для изменения параметров списков в стандарте CSS находится небольшой ряд свойств. К ним относятся list-style-type, list-style-image, list-style-position и сокращенный вариант list-style.

Свойство	Возможные значения	Описание
list-style-type	none disc circle square decimal lower-roman upper-roman lower-alpha upper-alpha lower-greek	определение вида маркеров в списке: без маркеров круги окружности квадраты арабские цифры цифры на основе строчных латинских букв римские цифры на основе прописных латинских букв строчные латинские буквы прописные латинские буквы строчные греческие буквы

Свойство	Возможные значения	Описание
OL {list-style-type: lower-alpha}		
list-style-image	none url(<путь к рисунку>)	определение рисунка, который будет использоваться в качестве маркера
UL {list-style-image: url(marker.gif)}		
list-style-position	inside outside	определение положения маркера относительно списка: маркер внутри списка (компактная форма) маркер вне списка
list-style	list-style-type list-style-position list-style-image	короткий способ определения стиля списка
UL {list-style: circle outside}		

- | | | | |
|-------------|-------------|----------------|--------------|
| ○ Coffee | ■ Coffee | I. Coffee | a. Coffee |
| ○ Tea | ■ Tea | II. Tea | b. Tea |
| ○ Coca Cola | ■ Coca Cola | III. Coca Cola | c. Coca Cola |

Направления музыки (list-style-position: inside)

1. Классическая
2. Джаз
3. Рок
4. Фолк

Любимые композиторы (list-style-position: outside)

• Моцарт
• Бах
• Вивальди



Для таблиц в CSS определены следующие важные свойства.

Свойство	Возможные значения	Описание
caption-side	top bottom left right	определяет положение заголовка таблицы, который задается с помощью тега <caption> относительно самой таблицы: по верхнему краю таблицы под таблицей слева от таблицы справа от таблицы
TABLE {caption-side: bottom}		
border-collapse	separate collapse	определяет способ прорисовки границ таблицы: раздельные рамки ячеек совместные рамки ячеек
TABLE {border-collapse: separate}		
про свойство border см. с. 116		определяет границы ячеек и таблицы
TABLE {border: 1px solid red}		
vertical-align	top bottom baseline middle	выравнивает содержимое ячейки по вертикали: по верхнему краю ячейки по нижнему краю ячейки по первой строке текста в ячейке по
TD {vertical-align: middle}		
table-layout	auto fixed	определяет способ отображения браузером ширины ячеек таблицы, основываясь на ее содержимом: браузер загружает таблицу, анализирует ее для определения размеров ячеек и после этого отображает ширина колонок либо определяется с помощью тега <col>, либо вычисляется на основе первой строки, для корректной работы этого значения обязательно должна быть задана ширина таблицы
TABLE {table-layout: }		

border-collapse: separate

Страна	Население в 1975	Население в 1976	Население в 1977
Австрия	7578903	7565525	7568430
Албания	2401602	2455497	2509909

border-collapse: collapse

Страна	Население в 1975	Население в 1976	Население в 1977
Австрия	7578903	7565525	7568430
Албания	2401602	2455497	2509909

Вопросы и упражнения

- Используя **CSS**, создайте графические маркеры у списка.
 - ☒ Молоко
 - ☒ Масло
 - ☒ Сыр
- С помощью **CSS** создайте такие стили, чтобы при прохождении курсора над ячейкой таблицы его вид менялся на тот, который указан в ячейке.

ВИДЫ КУРСОРА			
crosshair	default	auto	hand
ne-resize	nw-resize	w-resize	text
se-resize	s-resize	n-resize	sw-resize
wait	help	move	e-resize

- Группировка столбцов таблицы.** Создайте веб-страницу, на которой разместите таблицу размером 4 строки × 10 столбцов, где организуйте столбцы в несколько групп и придайте им общие свойства (фон, граница, размер, шрифт), как показано на рисунке. Используйте теги `<COL>` и `<COLGROUP>` для группировки столбцов.

Страна	Население в 1975	Население в 1976	Население в 1977	Женщины в 1975	Женщины в 1976	Женщины в 1977	Мужчины в 1975	Мужчины в 1976	Мужчины в 1977
Австрия	7578903	7565525	7568430	4618455	4644100	4710300	4462800	4517200	4558800
Албания	2401602	2455497	2509909	2550193	2558469	2567052	2506844	2512827	2519844
Андорра	26558	28500	29396	1554000	1572900	1621400	1581100	1639500	1660800

АНИМАЦИЯ В CSS

Возможности анимирования (плавных динамических изменений) свойств html-элементов только начинают приходить в CSS. В настоящее время Консорциум Всемирной паутины публикует описание анимационных возможностей таблиц стилей в рабочей спецификации (working draft). Большинство производителей браузеров уже поддерживает эти возможности.

Одна из анимационных возможностей связана с так называемым **правилом @keyframes**, которое позволяет разработчику описать анимацию css-свойств в виде перечня ключевых кадров.

```
@keyframes имя анимации {
    селектор ключевого кадра 1 {
        свойство 1: значение 1;
        свойство 2: значение 1;
    }
    селектор ключевого кадра 2 {
        свойство 1: значение 2;
        свойство 2: значение 2;
    }
    селектор ключевого кадра n {
        свойство 1: значение n;
        свойство 2: значение n;
    }
}
```

Такой способ организации анимации называется твинингом.

твининг (tweening) – процесс генерации программой промежуточных кадров между ключевыми кадрами, создающий впечатление, что объект на предыдущем ключевом кадре постепенно превращается в объект на следующем ключевом кадре



В качестве значения параметра *селектор ключевого кадра 1* можно использовать слово **from** или **0%**. Внутри этого селектора описываются значения анимированных свойств в начальный момент анимации.

В качестве значения параметра *селектор ключевого кадра 2* можно использовать слово **to**, если это конечный ключевой кадр анимации. Если

ключевых кадров больше двух, то параметр может принимать любые значения в процентах.

Пр и м е р ы.

```
@keyframes myMovie {  
  from {left: 500px}  
  to {left: 0px}  
}
```



```
@keyframes myMove2 {  
  from {transform: rotate(0deg)}  
  25% {transform: rotate(-15deg)}  
  50% {transform: rotate(0deg)}  
  75% {transform: rotate(15deg)}  
  to {transform: rotate(0deg)}  
}
```

Чтобы применить описанное правило к какому-либо html-элементу, используем CSS-свойство **animation**, которое имеет большое число параметров.

Свойство	Возможные значения	Описание
animation-name	none <i>имя</i>	задает имя анимации, при обращении к которому будет производиться анимация (можно указать несколько имен через запятую)
animation-name: slide		
animation-duration	число s число ms	задает продолжительность одного цикла анимации (можно указать несколько через запятую). Задается в секундах или миллисекундах
animation-duration: 10s		
animation-timing-function	ease linear ease-in ease-out ease-in-out cubic-bezier	описывает метод расчета скорости смены кадров анимации (можно указать несколько через запятую)
animation-timing-function: cubic-bezier(0.5, 0.3, 0.25, 0.25)		
animation-delay	число s число ms	задает время задержки анимации перед стартом (можно указать несколько через запятую)
animation-delay: 5s		

Свойство	Возможные значения	Описание
animation-iteration-count	1 infinite число	определяет количество циклов анимации (можно указать несколько через запятую). Дробные значения циклов округляются до целых в большую сторону, а отрицательные приравняются к 0
animation-iteration-count: 2		
animation-direction	normal alternate	определяет, нужно ли проигрывать анимацию в обратном направлении (можно указать несколько через запятую). Если нужно, тогда все нечетные циклы проигрывают анимацию в прямом направлении, а все четные – в обратном; анимация проигрывается только в прямом направлении анимация проигрывается в прямом и обратном направлениях
animation-direction: alternate		
animation-play-state	running paused	переключает анимацию на паузу или снова включает ее (можно указать несколько значений через запятую); при установке значения paused анимация останавливается, а при установке значения running продолжается с прерванного места
animation-play-state: paused		
animation-fill-mode	none forwards backwards both	определяет, будет ли видимым эффект от анимации, когда сама анимация уже закончилась (можно указать несколько через запятую); эффект от анимации будет наблюдаться лишь в течение времени анимации эффект от анимации будет виден даже тогда, когда анимация закончилась начальный ключевой кадр анимации будет отображаться в течение всего времени задержки начальный ключевой кадр анимации будет отображаться в течение всего времени задержки анимации, а последний будет отображаться даже тогда, когда анимация закончилась

Свойство	Возможные значения	Описание
animation-fill-mode: forwards		
animation		короткий способ задания свойств анимации, за исключением свойств animation-play-state и animation-fill-mode
animation: myMovie 5s linear 2s infinite alternate		

Пр и м е р. Перемещение картинки справа налево

<HTML>

<HEAD>

<TITLE>Анимация</TITLE>

<STYLE TYPE = "text/css">

.эффект1 {

position: relative;

animation: myMove 5s infinite;

animation-direction: alternate;

}

@keyframes myMove {

from {left: 500px}

to {left: 0px}

}

</STYLE>

</HEAD>

<BODY>

<P>

</P>

</BODY>

</HTML>

Кроме правила keyframes и свойства animation, для организации простых плавных изменений свойств элементов в CSS3 включили еще свойство **transition** (с англ. – переход). Это свойство является сокращенным вариантом целого ряда свойств, задающих параметры динамического перехода.

Свойство	Возможные значения	Описание
transition-property	none all имя свойства [,имя свойства]	задает имя свойства, значение которого будет меняться
transition-property: top;		
transition-duration	число с число ms	задает длительность перехода. По умолчанию значение равно 0s, то есть мгновенно
transition-duration: 2s;		
transition-timing-function	ease ease-in ease-out ease-in-out linear cubic-bezier(<i>n,n,n,n</i>)	устанавливает быстроту изменения значения стилевых свойств; анимация начинается медленно, затем ускоряется и к концу опять замедляется медленно начинается, к концу ускоряется начинается быстро, к концу замедляется начинается и заканчивается медленно одинаковая скорость от начала и до конца задает функцию движения в виде кривой Безье, где <i>n</i> – число от 0 до 1
transition-timing-function: linear;		
transition-delay	число с число ms	устанавливает время ожидания перед запуском эффекта перехода. Допустимо указывать несколько значений через запятую. Каждое значение будет применяться к свойству, заданному в transition-property
transition-delay: 0.5s;		

Пр и м е р. Вращение и масштабирование текстового блока:

```
<HEAD>
<TITLE> transition </TITLE>
<STYLE TYPE = "text/css">
.logo1 {
    width: 100px;
    height: 70px;
    background: pink;
    text-align: center;
    font-size: 10pt;
```

```

margin: 100px;
padding: 10px;
transition: width 2s, height 2s, transform 3s, font-size 3s,
padding 3s;
}
.logo1:hover {
width: 150px;
height: 100px;
font-size: 15pt;
padding: 15px;
transform: rotate(360deg);
}
</STYLE>
</HEAD>
<BODY>
    <DIV CLASS="logo1">
        <H3>CSS 3 <BR> transition</H3>

    </DIV>
</BODY>

```

Вопросы и упражнения

1. **Плавный переход.** С помощью свойства transition создайте у html-элемента эффект плавного изменения:
 - прозрачности от 1 до 0,
 - масштаба,
 - положения, то есть, например, выдвижение прямоугольной области, частично спрятанной за границей веб-страницы.
2. **Растворение.** Разместите на веб-странице картинку и с помощью css-свойств придайте ей эффект плавного растворения и появления.
3. **Катящийся объект.** Подберите рисунок и с помощью css-свойств создайте на веб-странице анимацию катящегося объекта (например, мяча).



4. **Масштабирование текста.** Разместите на веб-странице блок текста очень маленького размера. Разработайте стиль, многократно увеличивающий размер текста и применяющийся к миниатюре по наведению курсора мыши.
5. **Подпрыгивающая кнопка.** Создайте кнопку, которая при наведении указателя мыши будет подпрыгивать.

6. **Бегущая строка.** Создайте на веб-странице бегущую строку, не используя тег `<marquee>`. Используйте анимационные свойства таблиц стилей для тонкой настройки плавности и скорости перемещения строки.
7. **Выдвижной блок.** Создайте блок с контентом и разместите его частично за пределами окна браузера. Добавьте эффект, при котором при наведении указателя мыши на видимую часть блока он будет плавно целиком выходить на веб-страницу, при убирании указателя мыши – также плавно уходить на свое прежнее место.
8. **Слайдер.** Создайте на веб-странице блок, в котором автоматически плавно и циклично будут перемещаться картинки. Можно добавить кнопки перемещения картинок по щелчку указателя мыши.

ИМИТАЦИЯ HTML-ЭЛЕМЕНТОВ, ИЛИ ПСЕВДОЭЛЕМЕНТЫ

В начале книги говорилось о псевдоклассах таблиц стилей. Они определяют динамическое состояние элементов, которое изменяется в зависимости от действий посетителя веб-страницы. Например, текстовая ссылка изменяет свой цвет при наведении на нее курсора мыши.

Псевдоэлементы же как бы создают новые элементы и задают им стиль и содержимое, которого нет в исходном html-коде.

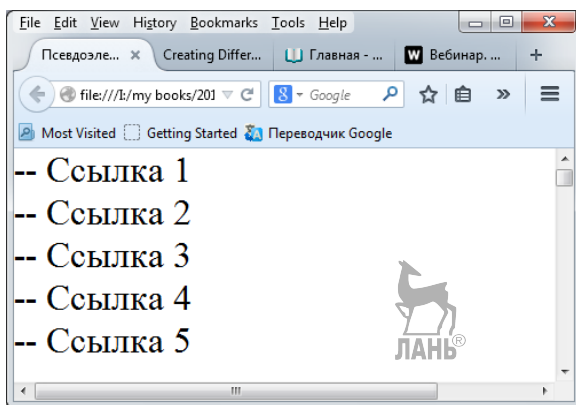
Первые псевдоэлементы появились в спецификации CSS 1, но самые интересные псевдоэлементы `:before` и `:after` были рекомендованы стандартом CSS 2.1. Общий вид синтаксиса:

Селектор:Псевдоэлемент { Описание правил стиля }

Вначале идет имя селектора, после него ставится двоеточие, а затем без пробела следует имя псевдоэлемента.

```
LI:before {content: "--"; /* Добавляем желаемый символ перед элементом списка */ }
```

```
LI {list-style: none; /* Убираем исходные маркеры */ }
```



Свойство **content** является обязательным. Оно позволяет вставлять генерируемое содержание в текст веб-страницы, которое первоначально в тексте отсутствует. По умолчанию значением этого свойства является пустая строка.

Кроме текстовой строки, значением свойства content могут быть следующие зарезервированные слова:

attr(параметр)	Возвращает строку, которая является значением параметра тега, указанного в скобках. Например: a:after {content:attr(href)} добавит после ссылки ее адрес, т.е. значение атрибута href. Если указанного атрибута нет, вернется пустая строка
open-quote	Вставляет открывающую кавычку
close-quote	Вставляет закрывающую кавычку
url(путь к файлу)	Вставляет указанный объект. Если указанный файл браузер не может отобразить, то значение игнорируется. Пример: content:url(smiley.gif)

Кроме свойства content можно указывать через точку с запятой и любые другие css-свойства.

Ранее для обозначения псевдокласса и псевдоэлемента применялось одно двоеточие, однако в CSS 3 для разделения псевдоклассов и псевдоэлементов ввели два двоеточия (::before и ::after). Но до сих пор можно продолжать использовать одно двоеточие (кроме ::selection).

Список псевдоэлементов

Имя	Описание
::before	применяется для вставки контента перед элементом
::after	применяется для вставки контента после элемента
::first-letter	представляет собой первый символ первой строки текста внутри элемента
::first-line	представляет собой первую линию форматированного текста
::selection	представляет собой часть документа, которая была выделена мышкой

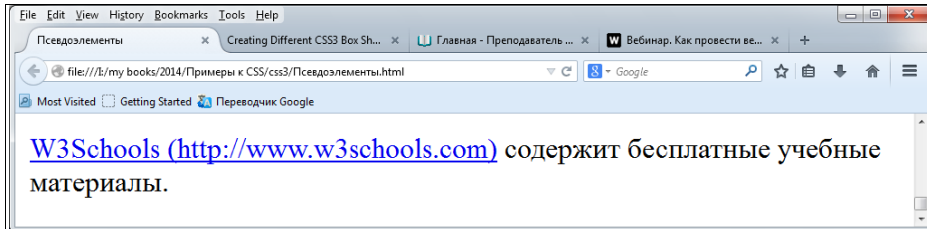
Псевдоэлементы :before и :after создают дополнительные элементы до или (и) после содержимого основного элемента. Необходимо иметь в виду, что элементы, которые генерируются псевдоэлементами, не создаются на самом деле. Поэтому их невозможно найти в исходном коде.

Пр и м е р.

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
  A:after { content: " (" attr(href) " ); }
</STYLE>
```



```
</HEAD>
<BODY>
  <P><A HREF="http://www.w3schools.com">W3Schools</A> содержит
    бесплатные учебные материалы.</P>
</BODY>
</HTML>
```



П р и м е р. Стилизованная цитата:

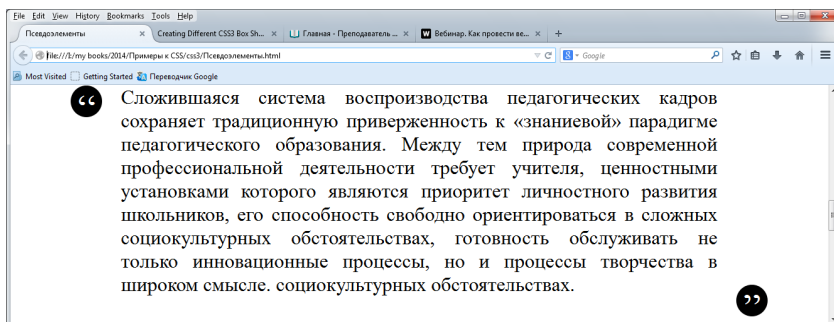
```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
BLOCKQUOTE:before {
  content: open-quote;
  font-size: 24pt;
  text-align: center;
  color: #fff;
  background: black;
  float: left;
  border-radius: 25px;
  height: 25px;
  width: 25px;
}
BLOCKQUOTE:after {
  content: close-quote;
  font-size: 24pt;
  text-align: center;
  color: #fff;
  background: black;
  float: right;
  position: relative;
  bottom: 25px;
  border-radius: 25px;
```



```

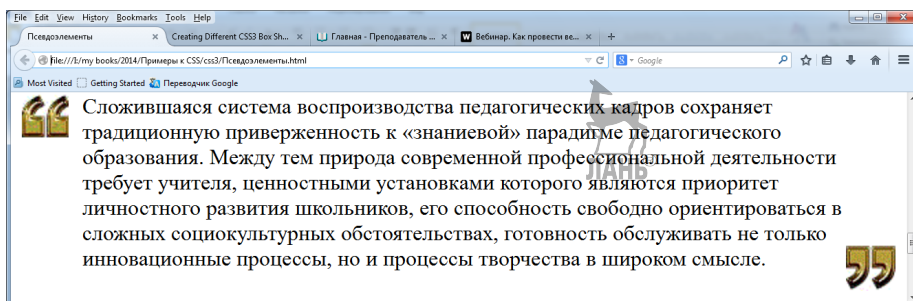
height: 25px;
width: 25px;
}
</STYLE>
</HEAD>
<BODY>
  <BLOCKQUOTE>
    <P STYLE="margin-left: 40px; margin-right: 40px; text-align: justify"> Сложившаяся система воспроизводства педагогических кадров сохраняет традиционную приверженность к «знаниевой» парадигме педагогического образования... </P>
  </BLOCKQUOTE>
</BODY>
</HTML>

```



Вопросы и упражнения

1. **Стилизованная цитата.** Оформите абзац текста на веб-странице как цитату, взятую в графические кавычки.



2. **Эффектная кнопка.** Создайте кнопку с эффектом плавной полупрозрачной заливки при наведении на нее указателя мыши. Заливка может распространяться:
 - а. Сверху вниз

- b. Снизу вверх
- c. Справа налево
- d. Слева направо

3. **Приподнятая иконка.** Для графической иконки создайте эффект приподнятия при наведении указателя мыши.

Примечание. Используйте псевдоэлемент и анимируйте его за счет изменения высоты. Для псевдоэлемента задайте фоновый цвет, и иконка будет приподниматься на расстояние равное высоте псевдоэлемента, как будто ее «подталкивают» вверх.



4. **Иконка, появляющаяся слева.** Разместите на веб-странице одноцветную иконку какой-либо соцсети. Создайте эффект, при котором при наведении указателя мыши на иконку цветной вариант этой иконки будет плавно выдвигаться слева и выталкивать ее одноцветный вариант.

Примечание. Используйте псевдоэлемент для двух состояний (обычного и при наведении). Псевдоэлемент перемещаться внутри контейнера может благодаря свойству `left`, а за счет свойства `transition` движение элементов можно сделать плавным.

5. **Титры на картинке.** Разместите на веб-странице изображение, на котором при наведении указателя мыши будет появляться небольшой поясняющий текст на затемненном фоне картинки.

Примечание.

1. Темный фон на картинке – это слой, который располагается поверх изображения и занимает все доступное пространство, отведенное под изображение. У данного слоя будут такие же параметры ширины и высоты, как и у изображения, независимо от размеров самого изображения. Слой добавляется с помощью псевдоэлемента `before`. Полупрозрачность дается свойством `opacity`. Цвет фона – черный. Данный слой появляется, когда пользователь наводит указатель мыши на изображение.

2. Появляющийся текст – это слой, который располагается поверх слоя с темным фоном. Данный слой добавляется с помощью псевдоэлемента `after`. Это прозрачный слой, который содержит только текст, полученный из значения атрибута `title` у элемента `a`. Данный слой спозиционирован по центру (по вертикали и горизонтали). Его анимация включает два эффекта: увеличение и поворот. Данные эффекты достигаются с помощью двух свойств: `transform` и `transition`. Свойство `transform` использует функции `scale` и `rotate`, чтобы увеличивать и поворачивать данный слой. Вся анимация начинается выполняться, когда пользователь наводит указатель мыши на изображение.

ИГРА С ТЕНЬЮ

Способ придания тени плоскому тексту был рассмотрен в параграфе «От плоского текста к 3D». Однако тень можно сгенерировать не только у текста, но и у любых других html-элементов, включая элементы разработчика (<div>). Для этого существует свойство box-shadow,

box-shadow: none | *тень* [, *тень*]

где *тень*:

сдвиг_по_х сдвиг_по_у радиус_размытия растяжение цвет inset

Параметр *растяжение* формирует толщину тени.

none	Отменяет добавление тени
сдвиг по х	Обязательный параметр. Смещение тени по горизонтали. Положительное значение этого параметра задает сдвиг тени вправо, отрицательное – влево
сдвиг по у	Обязательный параметр. Смещение тени по вертикали. Положительное значение задает сдвиг тени вниз, отрицательное – вверх
размытие	Радиус размытия тени. По умолчанию равен 0
растяжение	Положительное значение растягивает тень, отрицательное – сжимает. По умолчанию устанавливается 0, при этом тень будет того же размера, что и элемент
цвет	Цвет тени в любом из доступных CSS форматов. По умолчанию тень черная
inset	Тень выводится внутри элемента (внутренняя тень)

box-shadow: 0 0 10px rgba(0,0,0,0.5); /* Параметры тени */

Пр и м е р.

<HTML>

<HEAD>

<TITLE>Параметры тени</TITLE>

<STYLE TYPE = "text/css">

/*----- Эффект 1 -----*/

.тень1 {

width: 300px;

height: 100px;

text-align: center;

border: 1px solid black;

box-shadow: 0 10px 6px -6px #777;

}

/*----- Эффект 3 -----*/

```

.тень3 {
    position: relative;
    background-color: white;
    margin-top: 30px;
    width: 300px;
    height: 100px;
    text-align: center;
    border: 1px solid black;
}

.тень3:before {
    content: "";
    z-index: -1;
    position: absolute;
    bottom: 15px;
    left: 10px;
    width: 100px;
    height: 10px;
    top: 75%;
    background: #777;
    box-shadow: 0 15px 5px 3px #777;
    transform: rotate(-5deg);
}

.тень3:after {
    content: "";
    z-index: -1;
    position: absolute;
    bottom: 75%;
    right: 10px;
    width: 100px;
    height: 10px;
    top: 15px;
    background: #777;
    box-shadow: 0 15px 5px 3px #777;
    transform: rotate(-185deg);
}

```

</STYLE>

</HEAD>



<BODY>

<H1>Теневые эффекты</H1>

<DIV CLASS="тень1"> <P>Эффект 1</P> </DIV>

<DIV CLASS="тень3"> <P>Эффект 3</P> </DIV>

</BODY>

</HTML>



Вопросы и упражнения

1. **Теневые эффекты.** Создайте на веб-странице элемент с тенью, как показано на рисунке.



2. **Стек изображений.** Используя псевдоэлементы и теневые эффекты, создайте на веб-странице стилизованную под стопку изображений композицию.



АДАПТИВНАЯ ВЕРСТКА

Веб-дизайн сайтов с фиксированной шириной и полным соответствием макету прошли в связи с появлением широкоформатных мониторов, интернет-телевидения, планшетов и смартфонов различных размеров. Современный веб-дизайн должен принимать в расчет любые разрешения – от 320 до 7680 пикселей.

Широкий спектр разрешений предъявляет требования к изображениям – они должны растягиваться и уменьшаться, сильно не теряя качества. Это может

оказаться проблемой, так как за исключением векторной графики, большинство изображений имеет фиксированную ширину в пикселях.

Нынешнее, самое распространенное решение можно найти в CSS-коде практически любого адаптивного сайта:

```
img {max-width: 100%; height: auto;}
```

Этот код использует свойство `max-width: 100%` для того, чтобы изображение не выдавалось за пределы родительского контейнера. Если родительский контейнер станет меньше ширины изображения, изображение будет уменьшаться вместе с ним. Настройка `height: auto;` присутствует для того, чтобы при уменьшении сохранялись пропорции изображения.

Свойство `min-width` устанавливает минимальную ширину элемента. Если окно браузера по ширине становится меньше заданной минимальной ширины элемента, то ширина элемента остается неизменной, а в окне появляется горизонтальная полоса прокрутки.

Также для адаптации элемента можно использовать новые относительные единицы задания длины и ширины `vw` и `vh`.

Гибкая компоновка блоков

Спецификация CSS Flexible Box Layout Module [<https://www.w3.org/TR/css-flexbox-1/>] изменила ситуацию разметки в сторону гибкого решения огромного количества задач. Гибкий (`flex`) контейнер позволяет управлять размером, порядком и выравниванием элементов по нескольким осям, распределением свободного места между элементами. Гибкий контейнер автоматически адаптируется к изменению размера окна браузера.

Одним из основных понятий в `flexbox` являются оси.

- Главной осью `flex`-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы.
- Поперечной осью называется направление, перпендикулярное главной оси.

По умолчанию `flex`-контейнер всегда располагает блоки внутри себя в одну линию.

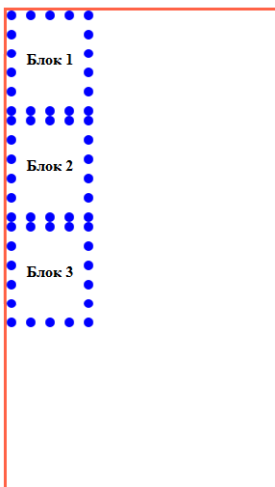
CSS свойства `flex-direction`, `justify-content`, `align-items` должны применяться непосредственно к `flex`-контейнеру, а не к его дочерним элементам.

Свойство	Возможные значения	Описание
flex-direction	row row-reverse column column-reverse	направление главной оси: слева направо справа налево сверху вниз снизу вверх
flex-direction: column		

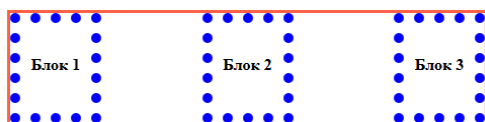
Свойство	Возможные значения	Описание
justify-content	flex-start flex-end center space-between space-around	выравнивание по главной оси блоки прижаты к началу главной оси блоки прижаты к концу главной оси блоки располагаются в центре главной оси первый блок располагается в начале главной оси, последний блок – в конце, все остальные блоки равномерно распределены в оставшемся пространстве все блоки равномерно распределены вдоль главной оси, разделяя все свободное пространство поровну
justify-content: center		
align-items	stretch flex-start flex-end center baseline	выравнивание элементов вдоль поперечной оси блоки растянуты, занимая все доступное место по поперечной оси, при этом все же учитываются min-width/max-width , если таковые заданы. блоки прижаты к началу поперечной оси блоки прижаты к концу поперечной оси блоки располагаются в центре поперечной оси блоки выровнены по базовой линии
align-items: center		

Примеры.

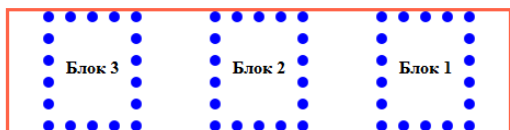
flex-direction: column
justify-content: flex-start



flex-direction: row
justify-content: space-between



flex-direction: row-reverse
justify-content: space-around



Многострочная организация блоков внутри flex-контейнера

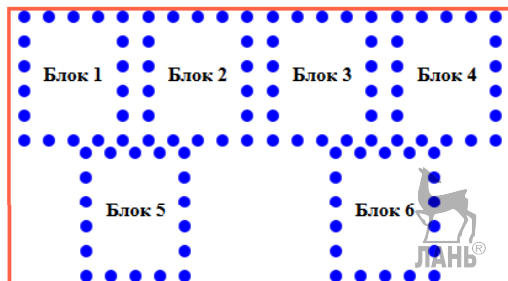
CSS свойства flex-wrap и align-content должны применяться непосредственно к flex-контейнеру, а не к его дочерним элементам.

flex-wrap	nowrap wrap wrap-reverse	разрыв строки на несколько строк: блоки расположены в одну линию слева направо (в rtl справа налево) блоки расположены в несколько горизонтальных рядов (если не помещаются в один ряд). Они следуют друг за другом слева направо (в rtl справа налево) то же, что и wrap , но блоки располагаются в обратном порядке
flex-wrap: wrap		
flex-flow		сокращение для flex-direction и flex-wrap, возможность в одном свойстве описать направление главной и многострочность поперечной оси
flex-flow: row nowrap		
align-content	flex-start flex-end center space-between space-around stretch	определяет то, каким образом образовавшиеся ряды блоков будут выровнены <i>по вертикали</i> и как они поделят между собой все пространство flex-контейнера: ряды блоков прижаты к началу flex-контейнера ряды блоков прижаты к концу flex-контейнера ряды блоков находятся в центре flex-контейнера первый ряд блоков располагается в начале flex-контейнера, последний ряд блоков блок – в конце, все остальные ряды равномерно распределены в оставшемся пространстве ряды блоков равномерно распределены в от начала до конца flex-контейнера, разделяя все свободное пространство поровну ряды блоков растянуты, дабы занять все имеющееся пространство
align-content: space-around		

Примеры.

flex-flow: row wrap

justify-content: space-around

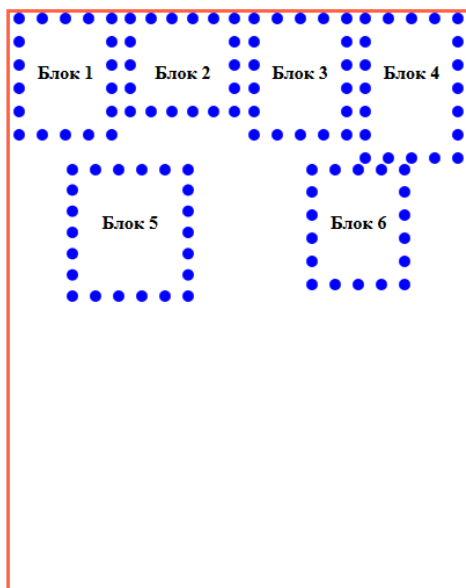


Если увеличивать ширину окна браузера, то блок5 и блок6 перейдут на верхнюю строку, и наоборот: если уменьшать ширину окна браузера, то блок4 и блок3 будут переходить на следующую строку.

flex-flow: row wrap

justify-content: space-around

align-content: flex-start



CSS правила для дочерних элементов flex-контейнера (flex-блоков)

flex-basis	auto	задает начальный размер по главной оси для flex-блока до того, как к нему будут применены преобразования, основанные на других flex-факторах. Может быть задан в любых единицах измерения длины (px, em, %, ...) или auto (по умолчанию). Если задан как auto – за основу берутся размеры блока (width, height), которые, в свою очередь, могут зависеть от размера контента, если не указаны явно
flex-basis: 100px		
flex-grow	0 1 2 3	определяет, насколько отдельный flex-блок может быть больше соседних элементов, если это необходимо. flex-grow принимает безразмерное значение (по умолчанию 0). Если все flex-блоки внутри flex-контейнера имеют flex-grow:1, то они будут одинакового размера, если один из них имеет flex-grow:2, то он будет в 2 раза больше, чем все остальные
flex-grow: 1		
flex-shrink	1	определяет, насколько flex-блок будет уменьшаться относительно соседних элементов внутри flex-контейнера в случае недостатка свободного места. По умолчанию равен 1
flex		короткая запись для свойств flex-grow, flex-shrink и flex-basis
flex: 12 3 30em		

Так как flex-контейнер может содержать достаточно большое число блоков, то удобно при обращении к отдельному блоку или некоторой группе элементов из всей совокупности использовать псевдокласс **:nth-of-type**. Он используется для добавления стиля к элементам указанного типа на основе их нумерации в дереве элементов. В примере показано обращение ко второму блоку всей совокупности элементов flex-контейнера.

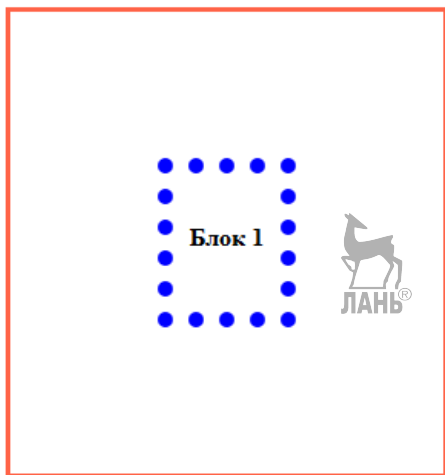
```
div:nth-of-type(2) {
    flex-basis: 80px;
}
```


CSS правила для отдельно взятых дочерних элементов flex-контейнера (flex-блоков)

align-self	stretch flex-start flex-end center baseline	выравнивание отдельно взятого flex-блока по поперечной оси flex-блок растянут, чтобы занять все доступное место по поперечной оси, при этом учитываются min-width/max-width, если таковые заданы flex-блок прижат к началу поперечной оси flex-блок прижат к концу поперечной оси flex-блок располагается в центре поперечной оси flex-блок выровнен по baseline
align-self: center		
order	0 целое число	изменяет порядок следования отдельно взятого flex-блока внутри flex-контейнера
order: 3		
margin	auto	выравнивание flex-блока по горизонтали и вертикали одновременно
margin: auto		

Пример.

display: flex
margin: auto



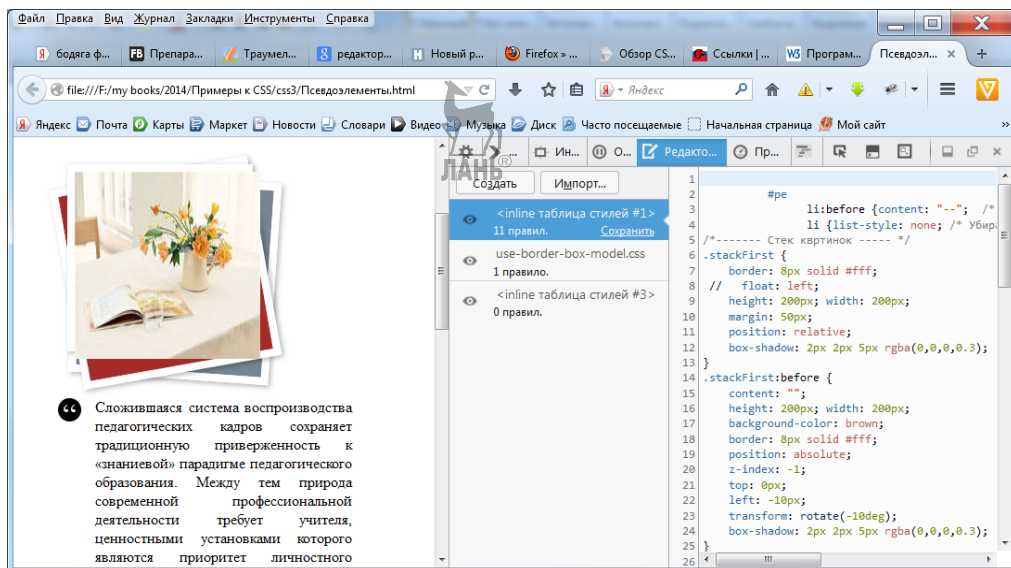
ЗНАЧОК ВЕБ-САЙТА

У каждого веб-сайта или веб-страницы может быть свой значок-иконка, который браузер размещает в заголовке окна, слева от содержимого тега <TITLE>. Как правило, этот значок совпадает с логотипом сайта. Размер иконки должен быть небольшим, например 16×16, или 32×32, или 64×64 и так далее. Если сохранить иконку под именем **favicon.ico** в корневом каталоге сайта, то браузер автоматически найдет ее и разместит в заголовке.

РЕДАКТОР СТИЛЕЙ В БРАУЗЕРЕ FIREFOX

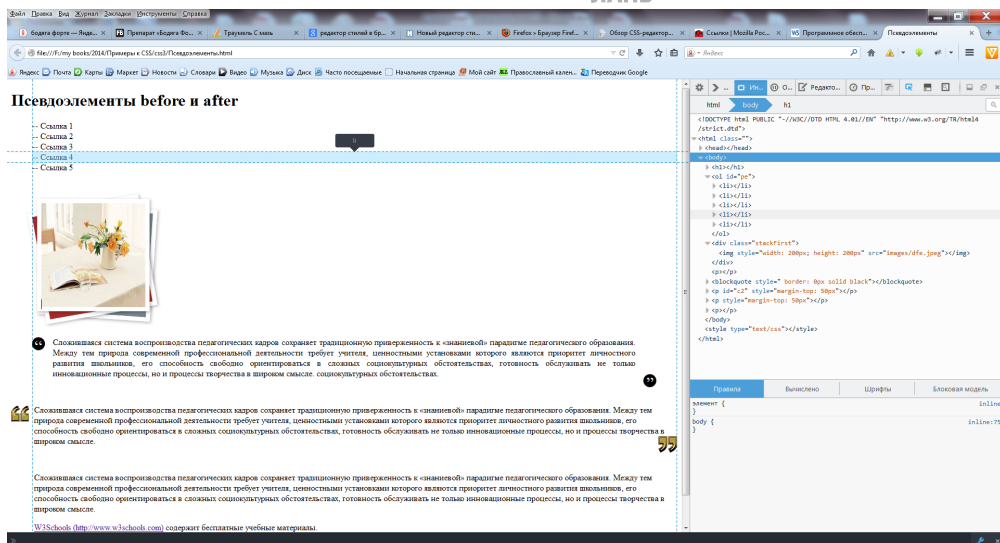
Для удобства редактирования стилей в последних версиях браузеров встроены специальные инструменты веб-разработчика. Так, в браузере Mozilla Firefox редактор стилей можно вызвать с помощью меню Инструменты → Разработка → Стили.

С помощью этого редактора можно создавать новые таблицы стилей для текущего документа и сохранять их в отдельном файле для последующего автономного использования. От текстовых этот редактор отличается тем, что изменения применяются моментально, как только вы заканчиваете печатать. Инструмент представляет собой двухпанельное окно, одна из панелей которого предназначена для отображения списка таблиц стилей, а вторая – css-кода выделенной таблицы. Разработчики имеют возможность создавать новые стили, импортировать их из файла, включать и выключать отдельные стили, получая возможность оценить мгновенно как выглядит страница.



Кроме редактора стилей в набор инструментов веб-разработчика для браузера Firefox входит **Инспектор** веб-страницы, который вызывается Инструменты → Разработка → Инспектор. Это очень удобный инструмент для

изучения или исследования структуры (дерева) просматриваемой в браузере веб-страницы. Можно щелчком мыши выбирать интересующий исследователя объект на веб-странице, а инспектор покажет в окне соответствующий ему код.



Практическая работа

Задание. Проект «Разработка веб-сайта»

Выберите индивидуальную интересную тему для создания веб-сайта и исследуйте пространство Интернета на наличие ресурсов по выбранной теме. Составьте аннотированный список ссылок этих ресурсов. Для раскрытия темы спроектируйте структуру сайта и изобразите ее графически в каком-либо графическом редакторе или в тетрадке. Составьте список информационных источников для своего веб-сайта.

Технические требования к сайту:

1. Содержит не менее 5 дочерних страниц.
2. Главная страница имеет фиксированный графический фоновый рисунок высокого разрешения, тему сайта, краткую аннотацию и навигацию.
3. Каждая страница имеет блочную разметку (не менее трех блоков).
4. Включает в себя не менее трех виджетов.
5. Содержит не менее одной внешней стилизованной гиперссылки.
6. На каждой странице располагается навигационная панель с интерактивными стилизованными ссылками.
7. Содержит не менее одной css-анимации.
8. Содержит абзацы текста и рисунки, списки и таблицы.

Список литературы

1. CSS Current Status [Electronic resource]. — Mode of Access : <http://www.w3.org/standards/techs/css#>
2. W3School Online Web Tutorial. CSS Tutorial [Electronic resource]. — Mode of Access : <http://www.w3schools.com/>
3. *Гастон П.* CSS3. Руководство разработчика. — СПб. : БХВ-Петербург : Русская Редакция, 2012.
4. *Макфарланд Д. С.* Большая книга CSS3. — СПб. : Питер, 2014.
5. Справочник по CSS [Электронный ресурс]. — Режим доступа : <http://htmlbook.ru/css>



Андрей Валентинович ДИКОВ
КЛИЕНТСКИЕ ТЕХНОЛОГИИ ВЕБ-ДИЗАЙНА
HTML5 И CSS3
Учебное пособие



Зав. редакцией
литературы по информационным технологиям
и системам связи *О. Е. Гайнутдинова*
Ответственный редактор *Т. С. Спирина*
Корректор *А. В. Попова*
Выпускающий *Н. А. Крылова*

ЛР № 065466 от 21.10.97
Гигиенический сертификат 78.01.10.953.П.1028
от 14.04.2016 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»
lan@lanbook.ru; www.lanbook.com
196105, Санкт-Петербург, пр. Юрия Гагарина, д. 1, лит. А
Тел./факс: (812) 336-25-09, 412-92-72
Бесплатный звонок по России: 8-800-700-40-71



Подписано в печать 12.07.19.
Бумага офсетная. Гарнитура Школьная. Формат 70×100 ¹/₁₆.
Печать офсетная. Усл. п. л. 15,28. Тираж 100 экз.

Заказ № 481-19.

Отпечатано в полном соответствии с качеством
предоставленного оригинал-макета в АО «Т8 Издательские Технологии».
109316, г. Москва, Волгоградский пр., д. 42, к. 5.