Основные понятия в веб-разработке

Что представляет собой веб-страница. Виды сайтов. Процесс разработки сайта. Гипертекст, теги и атрибуты. Структура HTML-документа. Основные теги оформления текста. Горячие клавиши для перемещения по документам.

[Как работает интернет](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.c713xnuw3xem)

[Браузеры](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.vj6333j387av)

[Что представляет собой веб-страница](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.ra1eclafl5qj)

[Виды сайтов](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.uifygeyjy4no)

[Протоколы передачи данных](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.ytqvmv91w6f)

[Схема HTTP-запроса страницы](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.5x4l7slytrp5)

[Процесс разработки сайта](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.r746v7y64wux)

[Гипертекст](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.8noua1eqwul)

[Теги и атрибуты](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.ef2f0zbajq5h)

[Структура HTML-документа](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.lukggz4a865t)

[Основные теги оформления текста](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.u3agn0mamdfg)

[Заголовки](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.fbatvkms52ar)

[Параграфы](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.awxnt8x1pe9u)

[Теги выделения текста](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.n6921ufc05wa)

[Вложенные теги](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.91sjigbxeknk)

[Необходимые инструменты](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.llwrfh1q6cnx)

[Горячие клавиши](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.4l6lpy2wy421)

[Практика](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.vg3uy1csazbk)

[Практическое задание](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.fd4r03flxxtd)

[Дополнительные материалы](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/14enjNoJxrY38RYu9xK3hyKqkA9HLjb2l2ybtDH1ywBo/edit#heading=h.uvp6qax5r1ok)

**Как работает интернет**

Прежде всего ответим на вопрос: что такое интернет? В Википедии находим следующее определение:

**Интерне́т (англ. Internet)** – всемирная система объединенных компьютерных сетей, построенная на использовании протокола TCP/IP и маршрутизации пакетов данных. Интернет образует глобальное информационное пространство, служит физической основой для Всемирной паутины (WWW, World Wide Web) и множества других систем (протоколов) передачи данных. Часто упоминается как «Всемирная сеть» и «Глобальная сеть», в обиходе иногда употребляют сокращенное наименование «Ине́т».

Другими словами, интернет состоит из множества домашних и корпоративных сетей, принадлежащих различным пользователям, компаниям и предприятиям, работающих по самым разнообразным протоколам, связанных между собой различными линиями связи, которые могут передавать данные по телефонным проводам, оптоволокну, через спутники и радиомодемы.

Интернет – это множество компьютеров по всему миру, объединенных в единую сеть, которые постоянно обмениваются какой-либо информацией.

Структура интернета напоминает паутину, в узлах которой находятся компьютеры, связанные между собой линиями связи. Как и объекты в реальной жизни, каждый компьютер имеет уникальный адрес, называемый IP-адресом. Существуют два типа IP-адресов:

* Постоянные, закрепленные за определенным компьютером.
* Динамические – присваиваются в тот момент, когда пользователь соединяется с интернетом.

Структура IP-адреса позволяет узнать, в какой стране и в каком городе находится компьютер пользователя. Таким образом, например, при настройке объявлений по контекстной рекламе можно задавать регионы показа объявления.

Пример IP-адреса: 192.168.1.2

Чтобы узнать свой IP-адрес, нужно зайти в «Панель управления» – «Сетевые подключения» – выбрать «Подключение по локальной сети» и перейти на вкладку «Поддержка».

Чтобы не запоминать сложные IP-адреса, придумали так называемые доменные имена.

**Доменное имя** – уникальное имя, которое данный поставщик услуг избрал себе для идентификации, например mail.ru или google.com.

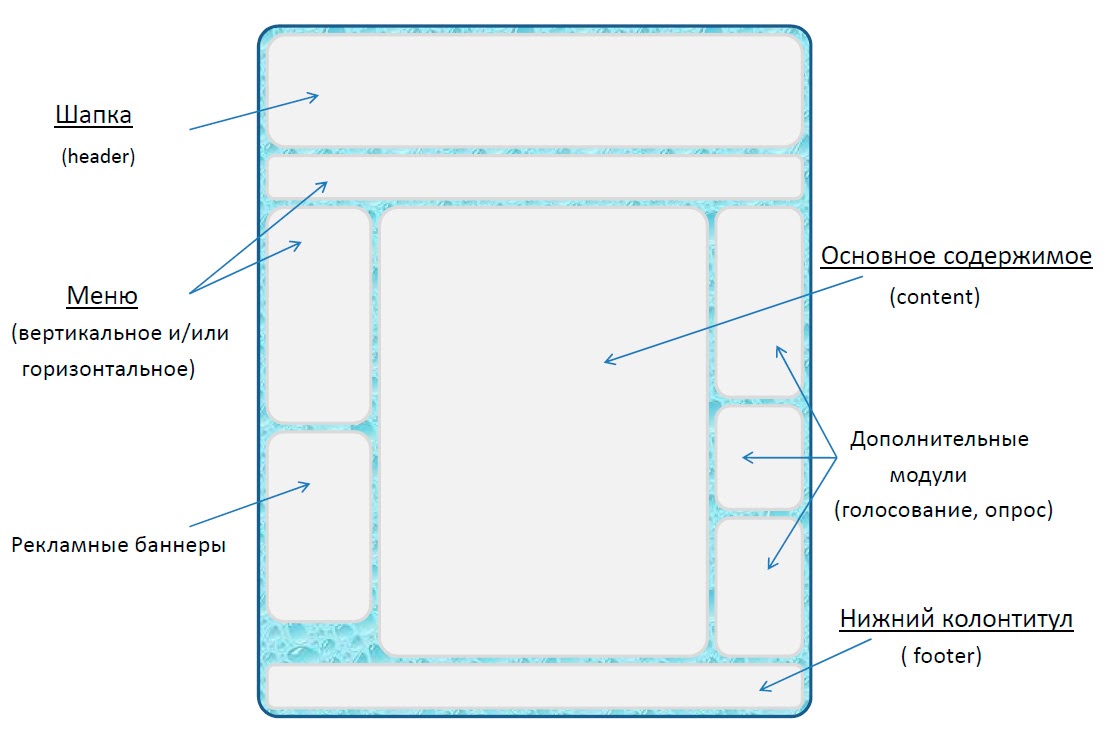
Доменное имя может иметь несколько уровней. Домен первого уровня обычно определяет страну местоположения сервера (ru – Россия; ua – Украина; uk – Великобритания; de – Германия) или вид организации (com – коммерческие организации; edu – научные и учебные организации; gov – правительственные учреждения; org – некоммерческие организации). С недавнего времени стало возможным использование русскоязычных доменов (рф).

**Браузеры**

**Браузер** – программа, с помощью которой вы имеете возможность просматривать любые сайты. То есть браузер является проводником между всемирной сетью интернет и пользователем. Именно он позволяет получать доступ ко всему множеству доступной в интернете информации.

Существует большое количество браузеров. Большинство из них практически одинаково отображают информацию, за исключением Internet Explorer, который некоторые теги и свойства стилей отображает иначе – к примеру, ширину блоков. Эти особенности необходимо учитывать при верстке, но подробнее об этом – в конце курса.

**Что представляет собой веб-страница**



Веб-страница сайта состоит из различных блоков:

* Шапка (header), в которой могут размещаться логотип компании, название сайта, телефоны организации и проч.
* Вертикальная и/или горизонтальная навигация по сайту (меню).
* Основное содержимое (content).
* Рекламные баннеры и ссылки на другие сайты.
* Дополнительные модули, например, голосование, опросы, корзина покупателя.
* Нижний колонтитул (footer), где размещается дополнительная информация: автор сайта, счетчики посещаемости.

**Виды сайтов**

**По содержимому**

Статические – содержимое подготавливается заранее и выдается пользователю в том виде, в котором хранится на сервере.

Динамические – содержимое генерируется при помощи серверных языков программирования.

**По схеме представления информации**

Коммерческие – сайты компаний, интернет-магазины и т.д.

Информационные – доносят до пользователя какую-либо информацию.

Веб-сервисы (порталы) – поисковые системы, электронная почта, форумы, социальные сети.

**По размеру**

Фиксированной ширины – задается фиксированная ширина сайта (в настоящее время ширина обычно составляет 1000 пикселей, т.к. разрешение современных мониторов по ширине начинается от 1024 пикселей). Тогда все блоки сайта будут выглядеть так, как было задумано, и не будут изменять размер в зависимости от разрешения монитора или размера экрана браузера.

«Резиновые» – ширина строго не задается, а изменяется в зависимости от размера экрана браузера или разрешения монитора.

Адаптивная верстка/тип макета – дизайн, который подстраивается (адаптируется) под размер экрана, в том числе может происходить перестройка блоков с одного места на другое или их замена блоками, отображаемыми только при определенном разрешении.

**Протоколы передачи данных**

Взаимодействие компьютеров, серверов, маршрутизаторов, коммутаторов определяется протоколами. Каждый протокол – четкий и определенный набор правил и соглашений, предписывающий, каким образом производится обмен и обработка информации.

Наиболее известные протоколы, используемые в сети интернет:

* HTTP (HyperText Transfer Protocol) – протокол передачи гипертекста. Используется при пересылке веб-страниц с одного компьютера на другой.
* HTTPS (HyperText Transfer Protocol Secure) – тоже протокол для передачи гипертекста, но использует дополнительное шифрование данных для более безопасной передачи информации.
* FTP (File Transfer Protocol) – протокол передачи файлов со специального файлового сервера на компьютер пользователя. FTP дает абоненту возможность обмениваться двоичными и текстовыми файлами с любым компьютером сети. Установив связь с удаленным компьютером, пользователь может скопировать файл с удаленного компьютера на свой и наоборот.

**Схема HTTP-запроса страницы**



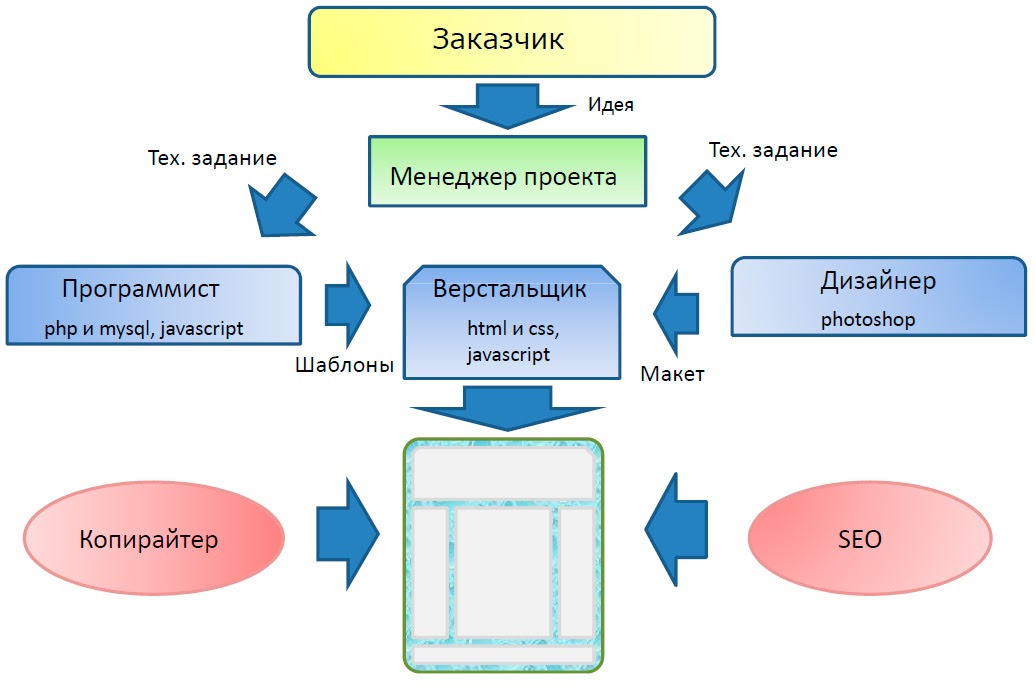
Пользователь набирает в браузере адрес нужного ему сайта, после чего посылается HTTP-запрос серверу. Сервер определяет тип страницы по расширению запрашиваемого файла.

Если расширение .html, это статическая страница, и сервер сразу формирует HTML-страницу и отправляет ее обратно пользователю. Если расширение, к примеру, .php, то в этом случае страница динамическая. Сервер подключает интерпретатор php, который, в свою очередь, может обратиться в базу данных и получить оттуда необходимые данные.

Результат – HTML-код, который веб-сервер посылает обратно на компьютер пользователя. Сгенерированный HTML-код веб-сервер упаковывает в HTTP-пакет, который и передается по сети обратно клиенту. Полученный HTTP-ответ попадает в браузер клиента, который извлекает из него HTML-код и генерирует на его основе графическое представление запрошенной страницы.

Простым языком: если мы обратились к HTML-файлу (.html), браузеру этого документа достаточно и Apache в случае такого запроса возвращает нам HTML-разметку. Если используется другое расширение файлов, например, .php, в данном файле может содержаться программный код, который должен быть обработан (выполнен) интерпретатором языка. На выходе php-интерпретатор должен нам выдать HTML-код, потому что именно его понимает браузер.

**Процесс разработки сайта**



На схеме представлен процесс разработки сайта на примере веб-студии, где определенные функции выполняет определенный человек.

Когда заказчик приходит со своей идеей сайта в веб-студию, с ним начинает работать менеджер проекта. Очень часто бывает, что идея у заказчика есть, но она настолько размыта, что тяжело представить на этом этапе, что в итоге хочет получить клиент.

В разных организациях используется свой подход к выяснению потребностей заказчика. Один из вариантов – специальная анкета, после составления которой можно уже представить, как будет выглядеть будущий сайт и каким функционалом обладать.

Далее менеджер составляет два технических задания: одно дизайнеру на составление макета сайта (дизайна), а другое – программисту с указанием, какой функционал необходимо реализовать в этом проекте.

После этого верстальщику отправляется макет сайта от дизайнера и различные модули от программиста, и после того, как сайт будет сверстан, копирайтер наполняет его контентом, т.е. заполняет товарами, пишет статьи и т.д. В это же время seo-специалист начинает заниматься продвижением сайта.

**Гипертекст**

Гипертекст – термин, введенный Тедом Нельсоном в 1965 году для обозначения «текста ветвящегося или выполняющего действия по запросу». Обычно гипертекст представляется набором текстов, содержащих узлы перехода между ними, которые позволяют избирать читаемые сведения или последовательность чтения.

Общеизвестным и ярким примером гипертекста служат веб-страницы – документы HTML (язык разметки гипертекста), размещенные в сети. В более широком понимании термина гипертекстом является любая повесть, словарь или энциклопедия, где встречаются отсылки к другим частям данного текста, имеющим отношение к данному термину. В компьютерной терминологии гипертекст – текст, сформированный с помощью языка разметки, потенциально содержащий в себе гиперссылки.

Теги и атрибуты

Язык HTML – язык тегов. Теги описывают структуру HTML-документа. Они оформляются угловыми скобками <имя тега>, между которыми прописывается имя тега. Теги HTML-документа предназначены для управления конструкциями разметки: заголовками, абзацами, списками, таблицами и картинками.

Теги бывают парные и одиночные.

Парные теги сначала открываются, потом закрываются. Все, что описывают парные теги, находится внутри них. Так, тег <b></b> делает текст полужирным. Вот как это выглядит:

|  |
| --- |
| <b>Какое прекрасное утро</b> |

Одиночные теги стоят сами по себе и, как правило, не меняют контент вокруг себя. Пример:

|  |
| --- |
| <br /> -- тег переноса строки <hr /> -- тег горизонтальной линии |

Атрибуты тегов определяют какие-либо дополнительные, уточняющие параметры того или иного тега. Вы можете указать внутри тега столько атрибутов, сколько вам необходимо. Атрибуты разделяются пробелами.

<название\_тега атрибут1="значение\_атрибута1"> текст внутри тега </ название\_тега >

Пример:

|  |
| --- |
| <a href="contacts.htm">Контакты</a>  <img src="book.png" border="1" alt="Книга" /> |

**Структура HTML-документа**

Каждая страница в HTML-документе состоит из трех обязательных элементов:

1. Тип документа.
2. Раздел <head></head> с технической информацией о странице: заголовок, описание, ключевые слова для поисковых машин, кодировка. Введенная в нем информация в основном не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.
3. Раздел <body></body>, где располагаются все элементы, которые видит пользователь. Именно с этим разделом мы в основном и будем работать.

DOCTYPE отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в интернете.

Элементы, находящиеся внутри тега <html>, образуют дерево документа, так называемую объектную модель документа, DOM (document object model). При этом элемент <html> является корневым.

Пример структуры HTML 5:

|  |
| --- |
| <!doctype html> <html> <head>     <title>Hello HTML</title>     <meta charset="UTF-8"> </head> <body> ... </body> </html> |

**Основные теги оформления текста**

Заголовки

Как и в газетах и журналах, в HTML-странице любая статья или новость должна начинаться с заголовка. Предусмотрено 6 уровней заголовков: первый из них будет отображен самым крупным шрифтом, а далее чем выше уровень, тем меньше будет размер шрифта.

.Для отображения заголовков существует тег <h> и указывается цифра от 1 до 6, которая соответствует уровню заголовка. Тег заголовка – парный, не забудьте его закрыть.

<h1>Заголовок первого уровня</h1>

<h2>Заголовок второго уровня</h2>

<h3>Заголовок третьего уровня</h3>

<h4>Заголовок четвертого уровня</h4>

<h5>Заголовок пятого уровня</h5>

<h6>Заголовок шестого уровня</h6>

Параграфы

После заголовка обычно следует какой-нибудь текст, который необходимо заключать в параграфы, или абзацы. При составлении документа выделяйте блоки текста в отдельные параграфы, как это сделано в книгах – в противном случае может получиться сплошной текст, который очень трудно будет читать посетителю сайта.

В HTML для параграфов используется парный тег <p>, и внутри него помещается тот текст, который нужно отобразить. Пример:

|  |
| --- |
| <p>Здесь мы напишем первый параграф</p> |

Теги выделения текста

Иногда необходимо выделить слово, словосочетание, предложение или целый участок текста, чтобы привлечь внимание читателя или поискового робота. Чтобы это сделать, можно выделить фрагмент жирным, курсивом или подчеркнуть.

Аккуратнее с подчеркиванием текста: по стандартам принято, что подчеркнуты гиперссылки, и многие пользователи уже настолько к этому привыкли, что, видя подчеркнутый текст, хотят «кликнуть» по этому участку текста, ожидая перехода на другую страницу.

Теги <b> и <strong> внешне делают одно и то же: выделяют текст полужирным. Разница в том, что <strong> указывает на важность текста, а <b> просто делает текст полужирным. Так, текст в тегах <strong> устройство для чтения текста вслух будет выделять интонацией, а <b> – нет.

Аналогично с тегами <em> и <i>, делающими текст курсивным: <em> указывает на важность текста, а <i> нет.

<small> уменьшает размер шрифта на единицу по отношению к обычному тексту.

<sub> используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.

<sup> используется для создания степеней и верхних индексов. Сдвигает текст выше уровня строки, уменьшая его размер.

<ins> выделяет текст в новой версии документа, подчеркивая его.

<del> перечеркивает текст. Используется для выделения текста, удаленного из документа.

<code> служит для выделения фрагментов программного кода. Отображается моноширинным шрифтом.

<pre> позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.

<q> используется для выделения коротких цитат. Браузерами заключается в кавычки.

Вложенные теги

Очень часто при верстке веб-страниц необходимо вкладывать одни теги в другие. Эта вложенность может достигать несколько уровней. Поэтому следует запомнить одно простое правило: закрывать теги необходимо именно в той последовательности, в которой вы их открывали. Первый открытый тег закрывается последним.

|  |
| --- |
| <p>Использовать вложенность тегов     <b>легко             <i>и просто</i>     </b>  </p> |

Тег <i> был открыт последним – закрываем его первым, а далее уже все остальные теги по очереди.

**Необходимые инструменты**

* Чтобы начать создавать свои веб-странички, необходим текстовый редактор. Можно воспользоваться встроенным в Windows блокнотом, но для начала удобнее использовать более функциональные текстовые редакторы Sublime text, Notepad++ или Brackets.
* Браузер. Выберите для себя наиболее удобный браузер, только на начальном этапе НЕ используйте Internet Explorer. Я рекомендую пользоваться браузером Google Chrome.
* Графический редактор (Photoshop или Gimp) – для «нарезки» макета сайта и подбора цветов.

**Горячие клавиши**

Alt + tab – перемещение между открытыми окнами.

Ctrl + z – отменить действие.

Ctrl + y – вернуть действие.

Ctrl + x – вырезать.

Ctrl + c – копировать.

Ctrl + v – вставить.

Ctrl + s – сохранить.

Ctrl + a – выделить все.

Ctrl + tab – перемещение между открытыми вкладками.

F5 – обновить страницу.

F2 – переименовать.

Shift + Tab – вернуть одну табуляцию назад.

Shift + home – выделить с указанного места до начала строчки.

Shift + end – выделить с указанного места до конца строчки.

Shift + стрелка – при зажатом Shift выделять в указанную стрелками сторону.

**Практика**

* Создаем на любом из жестких дисков новую папку.
* Заходим в эту папку, щелкаем правой кнопкой мыши, выбираем пункт «Создать текстовый документ» с расширением «.txt».
* Переименовываем этот файл, даем ему любое имя, а расширение меняем на «.html».
* Щелкаем по файлу правой кнопкой мыши, выбираем пункт «Открыть с помощью», выбираем текстовый редактор и открываем.
* Напишем теперь первое приветствие миру. Переходим в текстовый редактор и набираем: <h1>Hello world!</h1> <p>I’m here</p>.
* Чтобы посмотреть результат в браузере, щелкаем по этому файлу.

**Практическое задание**

Создание страницы товара каталога

1. Установить вышеуказанные инструменты.
2. Создать в удобном для вас месте на жестком диске папку, где будет храниться ваш интернет-магазин.
3. Выбрать тематику интернет-магазина.
4. Создать HTML-страницу  каталога товара по вашей тематике.

На странице должны присутствовать:

* + Структура HTML-документа.
  + Название товара.
  + Заголовок «Описание товара».
  + Картинка товара.
  + Текст краткого описания товара.
  + Заголовок «Характеристики товара».
  + Текст характеристик товара, который вы выбрали.
  + Заголовок «Подробное описание товара».
  + Текст подробного описания товара.
  + Горизонтальная черта.
  + Текст «Все права защищены».

1. Выучить комбинации горячих клавиш
2. **\*** Добавить произвольные подразделы описания товара, например описание внешнего вида или особенностей товара.
3. **\*** Добавить 2 страницы товаров по вашей тематике.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

**Дополнительные материалы**

[Что такое IP адрес](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81).

[Какие есть браузеры и как его выбрать](http://myblaze.ru/chto-takoe-brauzer-kakie-est-brauzeryi-i-kak-ego-vyibrat/).

[Полезная статья о структуре HTML-документа](http://htmlbook.ru/samhtml/struktura-html-koda).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/samhtml/struktura-html-koda>.
2. <https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80>.
3. <http://html.manual.ru/book/html/hierarchy.php>.
4. <https://webref.ru/>.

Основы HTML

Спецсимволы. Комментарии в HTML. Гиперссылки. Типы изображений для web. Загрузка изображений на страницу. Списки, формы и их элементы.

[Спецсимволы](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.c713xnuw3xem)

[Комментарии](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.ytqvmv91w6f)

[Гиперссылки](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.vzr1qdmjfpzz)

[Виды ссылок](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.6qq0n7xlrbts)

[Якоря в гиперссылках](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.c0u521wgb0mz)

[Изображения](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.pc0khmlyr7vi)

[JPEG](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.1dxl93ac4vgs)

[GIF](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.nzix2mb2heu8)

[PNG](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.agq3tx5cr36i)

[SVG](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.njotcuvuq8nh)

[Загрузка изображений на страницу](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.gpzo8ztt8ear)

[Списки](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.et1e1g51a5ff)

[Структура вложенного списка](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.myslvzvndp10)

[Формы и их элементы](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.1ayswm1qdtbp)

[Структура форм](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.firj37crlh1a)

[Элементы форм](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.mgb9stlk35n)

[Тег <label>](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.jbxz3w3z3xdp)

[Практика](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.vg3uy1csazbk)

[Гиперссылки](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.dw2h70i55it3)

[Элементы форм](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.itvlyt3k87h7)

[Практическое задание](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.3tfrjxxltv85)

[Дополнительные материалы](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.uvp6qax5r1ok)

[http://htmlbook.ru/html/img.](https://docs.google.com/document/d/1h5mvdlnIAhVdRwhymRNU7991SBL2imS9P99rGEy2lqs/edit#heading=h.bydcd3k7dbm4)

**Спецсимволы**

Используя спецсимволы в HTML, можно заменять уже имеющиеся на клавиатуре компьютера символы либо указать символы, которые отсутствуют на клавиатуре, например значок копирайта.

Любые теги никак не показываются в окне браузера, поскольку воспринимаются им как команды для вставки элементов и изменения их свойств. Но иногда требуется на веб-странице вывести теги, например для демонстрации HTML-кода. В этом случае используйте спецсимволы &lt; и &gt; для замены угловых скобок < и >.

Вот некоторые варианты спецсимволов:

|  |  |  |
| --- | --- | --- |
| Код в HTML | Внешний вид | Описание |
| &nbsp; |  | неразрывный пробел |
| &copy; | © | знак copyright |
| &quot; и &raquo; | « и » | двойные кавычки |
| &lt; и &gt; | < и > | символы |
| &prime; | ′ | одиночный штрих |

**Комментарии**

В HTML-документах можно оставить какой-либо комментарий, например, название части страницы, который не будет отображен в браузере. Но любой желающий сможет увидеть этот текст, если воспользуется функцией просмотр кода страницы в любом браузере.

Комментарии можно применять для того, чтобы не отображать теги в браузере. Это может пригодиться при редактировании веб-страницы, при этом вы можете комментировать целые блоки кода.

Пример:

|  |
| --- |
| <!--Текст, который не будет отображаться на сайте-->  <p>Текст основной части сайта</p> |

**Гиперссылки**

Гиперссылка – активный элемент (текст, изображение, кнопка и т.п.) одного документа, при нажатии на который происходит переход к другому документу или к его элементу. Документы, содержащие гиперссылки, называются гипертекстовыми. Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически.

В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом, при наведении на них курсора мыши в окне браузера меняют цвет или выделяются подчеркиванием. Посещенная ранее ссылка обычно выделяется цветом, отличным от цвета непосещенной ссылки.

Тег ссылки называется <a>, и у него есть обязательный атрибут href, в значении которого мы указываем на файл, место в документе или страницу другого сайта, куда хотим перейти. В содержимом этого тега необходимо написать текст, который увидит пользователь.

Виды ссылок

Относительные ссылки используются для перемещения внутри документа или сайта.

|  |
| --- |
| <a href="file\_name.html">      текст ссылки, который видит пользователь  </a> |

Абсолютные ссылки используются для перехода на страницы внешнего сайта. Для этого в значении атрибута href нужно указать полный путь до той страницы, на которую мы хотим перейти, включая тип протокола.

|  |
| --- |
| <a href="<http://yandex.ru>" target="\_blank">      страница yandex.ru откроется в новой вкладке  </a> |

У тега ссылки существует атрибут target. Если указать target="\_blank" , то страница откроется в новой вкладке браузера. Раньше это значение официально не поддерживалось в HTML, но люди все равно использовали его, поскольку оно работало. Теперь в HTML5 оно полностью поддерживается.

Когдаиспользовать target="\_blank":

* Если необходимо открыть сторонний ресурс и не потерять страничку вашего сайта.
* Если пользователь работает над чем-то, что может быть потеряно, если текущая страница изменится.

Когда не использовать target="\_blank":

* В меню (иначе пункты меню будут открываться в новых вкладках).
* Из-за личных предпочтений.
* Если клиент думает, что так будет лучше.

Якоря в гиперссылках

При помощи гиперссылок можно перемещаться не только между страницами документа, но и внутри данной страницы. Это бывает очень полезно, когда ваша статья состоит из нескольких разделов и имеет довольно объемное содержание.

В этом случае разумно использовать так называемые якоря, чтобы быстро переходить в интересующий раздел. В том элементе, на который следует перейти, нужно определить уникальный идентификатор.

Для этого существует атрибут id, который можно определять практически для любого тега. Название этого атрибута может быть любым и обязательно уникальным. В значении атрибута href тега-ссылки ставится символ решетки, после которого указывается имя того идентификатора, к которому необходимо обратиться.

|  |
| --- |
| <a href="#p10">Прочитать 10 параграф</a> <p id="p1">Текст параграфа №1</p> <p id="p2">Текст параграфа №2</p> ... <p id="p10">Вот и долгожданный параграф №10</p> |

**Изображения**

В настоящее время в веб-документах используются четыре формата изображений: jpeg, gif, png, svg. Рассмотрим каждый из них подробнее.

JPEG

*Плюсы:*

* Небольшой размер файла.
* Поддерживает 16 млн цветов.
* Можно управлять качеством изображения при сохранении.

*Минусы:*

* При сжатии размеров теряется качество.
* Не поддерживает прозрачность.

GIF

*Плюсы:*

* Поддерживает анимацию.
* Поддерживает прозрачность.
* При сжатии не теряет в качестве.

*Минусы:*

* Количество цветов до 256.

PNG

Растровый графический формат PNG, набирающий всё большую популярность, появился в 1995 году как замена GIF. Формат PNG, в свою очередь, подразделяется на форматы PNG-8 (аналог формата GIF, за исключением того, что PNG-8 не поддерживает анимацию) и PNG-24.

Еще существует формат PNG-32 – в этом формате растровой графики для одного пикселя возможно использовать четыре байта информации. Три отвечают за формирование полноцветного изображения, а четвертый байт выделяется отдельно для формирования альфа-канала, позволяющего использовать PNG-32 для получения картинок с прозрачным фоном.

Проблема поддержки 32-битной (полной) прозрачности картинки формата PNG в Microsoft Internet Explorer была решена в седьмой версии браузера.

PNG – 8

*Плюсы:*

* Поддерживает прозрачность.
* При сжатии не теряет в качестве.

*Минусы:*

* Количество цветов до 256.

PNG – 24

*Плюсы:*

* При сжатии не теряет в качестве.
* Использует 16 млн цветов.
* Плавный переход от прозрачной области к цветной.

*Минусы:*

* Большой размер файла.

SVG

SVG – формат векторной графики, файлы SVG можно читать и редактировать (при наличии некоторых навыков) при помощи обычных текстовых редакторов. Существует возможность увеличить любую часть изображения SVG без потери качества.

Широко доступно использование растровой графики в SVG-документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG. Текст в графике SVG является текстом, а не изображением, поэтому его можно выделять и копировать, он индексируется поисковыми машинами, не нужно создавать дополнительные метафайлы для поисковых роботов.

*Плюсы:*

* Файл имеет маленький размер, если изображение простое, без большого количества деталей.
* Возможно увеличить любую часть изображения SVG без потери качества.

*Минусы:*

* Вес файла начинает сильно увеличиваться при добавлении новых деталей на изображение.
* Сложность использования.

Из рассмотренных сильных и слабых сторон форматов изображений можно сделать вывод. Для сохранения фотографий с четкими краями лучше всего подходит формат JPEG, так как в нем размер файла получается небольшим. Для анимированных изображений единственный вариант – формат GIF, а для сохранения качественных изображений, в которых используется прозрачность, лучше всего подойдет формат PNG-24. Для векторной графики идеально подойдет SVG.

Загрузка изображений на страницу

Для отображения изображений на странице существует тег <img>, который является одиночным. У этого тега есть обязательный атрибут src, в значении которого указывают путь к изображению. Этот путь может быть как относительным, так и абсолютным.

|  |
| --- |
| <img src="img/my\_foto.jpg"       alt="это моя фотография"      title="это моя фотография"      width="300" height="300"  /> |

У тега <img /> можно указать дополнительные атрибуты.

Атрибут alt означает альтернативный текст. Используется он для того, чтобы: во-первых, если картинка не загрузится на страницу, пользователь увидит тот текст, который вы указали в этом атрибуте, во-вторых, альтернативный текст необходим при продвижении сайта, в нем можно указать ключевые слова, и тогда это изображение будет участвовать в поиске по картинкам.

Следующий атрибут – title, который является универсальным, его можно использовать практически для любого тега. То, что вы напишете в его значении, будет выводиться в виде всплывающей подсказки.

Любому изображению можно задать ширину и высоту, указав эти значения в атрибутах width и height соответственно. При помощи этих атрибутов изображение лучше не увеличивать, иначе получится очень плохое качество. Если задать ширину и высоту изображению, браузер при загрузке страницы будет сразу выделять заданную область под картинку.

Всегда храните все изображения в отдельной папке, чтобы знать, где искать то или иное изображение.

**Списки**

Списки в HTML-документах бывают двух видов: маркированные и нумерованные. Отличаются они тем, что у маркированных элементы списка начинаются с маркеров, а у нумерованного – с цифр или букв.

Структура маркированного списка:

|  |
| --- |
| <ul>     <li>Первый элемент списка</li>     <li>Второй элемент списка</li>     <li>Третий элемент списка</li> </ul> |

Структура нумерованного списка:

|  |
| --- |
| <ol>     <li>Первый элемент списка</li>     <li>Второй элемент списка</li>     <li>Третий элемент списка</li> </ol> |

Как видно из примеров, структура обоих типов списков одинаковая, единственное различие в том, что в случае с маркированным списком контейнер будет называться <ul>, а в нумерованном – <ol>. В нумерованном списке иногда может потребоваться начать список с какой-нибудь определенной цифры или буквы. Для этого существует атрибут start, в значении которого указывается, с какого элемента будет начинаться список. В списках есть возможность менять вид маркеров, например, арабские цифры изменить на римские, либо вообще убрать маркеры.

Структура вложенного списка

|  |
| --- |
| <ol>     <li>Первый элемент списка</li>     <li>Второй элемент списка           <ul>                <li>Первый элемент вложенного списка</li>                <li>Второй элемент вложенного списка</li>                <li>Третий элемент вложенного списка</li>          </ul>     </li>     <li>Третий элемент списка</li> </ol> |

Вложенные списки предназначены для организации сложной иерархической структуры текста, обычно юридических или технических документов. Также вложенные списки используются для создания многоуровневых меню и навигации по сайту.

**Формы и их элементы**

Формы необходимы для того, чтобы было возможно получать от пользователя информацию, отправлять эти данные на сервер, а там уже их обрабатывать. Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна. По этой причине данные форм должны быть независимы друг от друга.

Допускается внутрь контейнера <form> помещать другие теги, при этом сама форма никак не отображается на веб-странице, видны только ее элементы и результаты вложенных тегов.

Структура форм

Все элементы форм должны обязательно находиться внутри контейнера <form>. В этом теге могут находиться атрибуты, необходимые для правильной работы формы.

Элемент <fieldset> предназначен для группирования элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных. Например, один блок может быть предназначен для ввода текстовой информации, а другой – для флажков. Внутрь тега <legend> помещают название данной формы. Тогда браузер выделит эту часть рамкой, в которой будет находиться название.

Элемент <legend> не обязателен, но если присутствует, должен идти сразу же после тега <fieldset>. Другие теги или текст перед <legend> недопустимы. Внутри <legend> можно использовать теги форматирования вроде <b>, <i>

|  |
| --- |
| <form action="URL"> <fieldset>          <legend>Контактные данные</legend> ...  </fieldset>  </form> |

Элементы форм

В HTML существует три основных тега элементов форм:

1. <textarea> используется для того, чтобы сформировать многострочное поле ввода для большого количества информации. С помощью атрибута cols можно задать ширину этого поля, а с помощью атрибута rows указать число строк. Тег <textarea> парный.
2. <select> представляет собой выпадающий список, элементы которого указываются в теге <option>. У этого тега существует несколько атрибутов. Чтобы сделать возможность выбора нескольких пунктов, необходимо задать атрибут multiple, в значении которого также необходимо указать multiple. В атрибуте size указывается число строк выпадающего списка, которые увидит пользователь в браузере. Если необходимо, чтобы какой-то из элементов выпадающего списка был выбран, нужно этому элементу в теге <option> задать атрибут selected. Тег <select> также парный.
3. Тег <input /> в зависимости от значения атрибута type будет выглядеть по-разному и иметь разные значения. Тег <input /> одиночный.

Рассмотрим более подробно тег <input />

|  |
| --- |
| <input type="text" size="30" placeholder="Ваше Имя"> <input type="password" size="8"> <input type="checkbox" checked="checked"> <input type="radio" name="radio"> <input type="file"> <input type="submit" value="Сохранить"> <input type="reset" value="Очистить"> <input type="button" value="просто кнопка"> |

У первого тега <input /> значение атрибута type="text". Это означает, что это обычное однострочное поле ввода. В значении атрибута size указывается размер этого поля в символах. Атрибут placeholder выводит текст внутри текстового поля, который исчезает при начале ввода символов.

<input /> со значением password выглядит так же, как и предыдущий, но используется для ввода пароля, и все символы будут показываться как точки, чтобы пароль нельзя было подсмотреть.

Следующие два варианта – <input type="checkbox" /> и <input type="radio" /> – предоставляют пользователю возможность выбора некоторых пунктов. В первом случае есть возможность множественного выбора нескольких пунктов в виде флажков, а во втором – единственного выбора в виде переключателей. В обоих случаях можно отметить любой пункт по умолчанию, если задать атрибут checked="checked".

<input type="file" /> предоставляет пользователю возможность выбора файла на своем компьютере, чтобы загрузить его на сервер.

Оставшиеся 3 тега <input /> – кнопки. submit отправляет данные на сервер, reset предназначена для того, чтобы очистить все поля в форме. А третья type="button" – просто кнопка, при нажатии на которую ничего не произойдет, но ее, например, можно привязать к событию javascript. У всех этих кнопок в значении атрибута value указывается текст, который будет отображен на кнопке.

Тег <label>

Тег <label> устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы (<input>, <select>, <textarea>). Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст. Кроме того, с помощью <label> можно устанавливать горячие клавиши на клавиатуре и переходить на активный элемент подобно ссылкам.

Существует два способа связывания объекта и метки. Первый заключается в использовании идентификатора id внутри элемента формы и указании его имени в качестве атрибута for тега <label>. При втором способе элемент формы помещается внутрь контейнера <label>.

|  |
| --- |
| *<!-- Первый способ -->* <input id="идентификатор"> <label for="идентификатор">Текст</label> *<!-- Второй способ -->* <label><input type="...">Текст</label> |

**Практика**

Гиперссылки

|  |
| --- |
| *<!-- Создание меню-->* <ul>     <li>Главная</li>     <li><a href="catalog.html">Каталог</a></li>     <li><a href="contacts.html">Контакты</a></li> </ul> |

Элементы форм

|  |
| --- |
| <form action=""> <fieldset>           <legend>Страница регистрации</legend>          <input type="text" size="30" placeholder="Ваше Имя">          <input type="password" size="8">          <input type="checkbox" checked="checked">          <input type="radio" name="radio">          <input type="file">          <input type="submit" value="Сохранить">          <input type="reset" value="Очистить">          <input type="button" value="просто кнопка"> </fieldset>  </form> |

**Практическое задание**

Создание основного наполнения нашего сайта, главной страницы, каталога, товара каталога и контактов.

1. Создать файл index.html – это будет главная страница. На этой странице сделайте ссылки в виде маркированного списка на следующие страницы:
   1. Меню (должно располагаться на каждой странице сайта).
      1. Главная.
      2. Каталог.
      3. Контакты.
   2. Отделить горизонтальной чертой шапку сайта.
   3. Добавить название вашего магазина.
   4. Напишите приветственное сообщение.
   5. Горизонтальная черта.
   6. Подвал (должен располагаться на каждой странице сайта).
2. Что должно присутствовать в footer:
   1. Значок копирайта.
   2. Текст «Все права защищены».
3. На странице «Каталог»:
   1. Меню сайта.
   2. Горизонтальная черта.
   3. Заголовок «Каталог».
   4. Уменьшенные копии изображений товаров из вашего магазина.
   5. Ссылки под картинками для перехода в подробное описание товара.
   6. Footer.
4. На странице «Товар каталога» (страницы из первого ДЗ):
   1. Добавить картинку товара с возможностью нажатия на нее. Картинка должна открыться в полном размере, в новом окне.
   2. В подразделе «Характеристики товара» необходимо добавить маркированный или нумерованный список характеристик товара.
5. На странице «Контакты»:
   1. Меню сайта.
   2. Горизонтальная черта.
   3. Заголовок «Напишите нам».
   4. Поля для заполнения:
      1. Имя.
      2. E-mail.
      3. Тема.
      4. Текстовое поле.
      5. **\*** Произвольные поля, которые на ваше усмотрение нужны на вашем сайте.
   5. Заголовок «Адрес».
      1. Контактный номер телефона интернет-магазина.
      2. Адрес.
      3. E-mail.
      4. **\*** Любая информация о магазине.
6. **\*** На страницу контактов добавьте реальную карту Yandex или Google.
7. **\*** Если вы сделали несколько товаров, добавьте описание и характеристики для каждого товара.
8. **\*** Количество страниц товаров не ограничено.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

**Дополнительные материалы**

[Статья про типограф, с чего всё начиналось и для чего же он нужен](https://habrahabr.ru/post/21727/).

[Чуть подробней про гипертекст](http://www.webremeslo.ru/html/glava5.html).

[Какие изображения применяются для html](http://htmlbook.ru/metki/izobrazheniya).

[Формы html](http://htmlbook.ru/html/type/form).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/samhtml/tekst/spetssimvoly>.
2. <http://htmlweb.ru/html/symbols.php>.
3. <http://htmlbook.ru/html/a>.
4. <http://htmlbook.ru/html/img>.
5. <http://htmlbook.ru/html/form>.

Основы CSS

Что такое CSS. Синтаксис CSS. Способы объявления CSS. Селекторы (id, class, tag). Селекторы атрибутов. Основные свойства стилей. Вложенность. Наследование и группирование свойств.

[Что такое CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.5rqceql2xao9)

[Синтаксис СSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.aq1v5poq3om3)

[Оформление CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.146w10qzgti6)

[Комментарии в CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.bqcjgqd5xha8)

[Способы объявления CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.mc1snqa5blc1)

[Inline-стили](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.hg7y4xc6horw)

[Cтили в разделе head](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.o7k47b9z7cdl)

[Внешний CSS-файл](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.d84cozpz9mfc)

[Какой способ подключения стилей выбрать?](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.5wlwkw43sar6)

[Селекторы в CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.gkuhpympvx8r)

[Селекторы тегов](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.beknw3ttjjlp)

[Селекторы идентификаторов (id)](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.c19q9aaxgoqz)

[Селекторы классов (class)](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.fitevrqcmira)

[Селекторы атрибутов](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.obdehnhttete)

[Свойства стилей](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.rj0r17s820ls)

[Единицы измерения в CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.i1byglf3r0lf)

[Относительные единицы измерения](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.bivzvh8dmnpr)

[Пиксели](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.50z6kn9iw4kb)

[Проценты](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.xnpd77y0yr2t)

[Абсолютные единицы измерения](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.47ozshu9px3q)

[Способы задания цветов](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.sxteqcvnz7th)

[Функциональный RGB](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.l5vum9d7vwm5)

[Шестнадцатеричный RGB](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.31vfsaynp7ky)

[Свойства стилей CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.b5ta3yxlgshy)

[Ширина и высота: width и height](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.g4plsgdey5zt)

[Фон элемента – background](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.xo3wd6dt0ijq)

[border – рамка вокруг элемента](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.kzlqnu7ox0l8)

[Цвет текста – color](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.6yn782ytylus)

[Шрифт – font](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.n9wp2fnimspv)

[font-family устанавливает шрифт текста.](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.nqi1jw13m2qn)

[Оформление списков – list-style](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.kf1n4wp79i8r)

[Вложенность](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.41fo0k78avag)

[Контекстные селекторы](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.orjc7272zf3o)

[Дочерние селекторы](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.4hmahl3jhl2v)

[Соседние селекторы](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.k6b5aa3nqk74)

[Наследование](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.vimpk65e9qqd)

[Группировка свойств](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.ot8cixmd898u)

[Приоритеты стилей в CSS](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.c5d9qfauyv0k)

[Приоритеты источников стилей](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.ce59s82gu9ns)

[Приоритеты стилей автора](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.3wuczow10oeo)

[Практика](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.vg3uy1csazbk)

[Создание стилей для меню сайта](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.69nq3b6y1irv)

[Практическое задание](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.r2fm9gmb215y)

[Дополнительные материалы](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1VvlSUTy-osttxbvpARtRex7tdCuiSjinl-hLFSbXFIU/edit#heading=h.uvp6qax5r1ok)

# Что такое CSS

CSS (Cascading Style Sheets) – это каскадные листы стилей, которые применяются для описания внешнего вида веб-документа, написанного при помощи языка разметки HTML.

Другими словами, с помощью HTML появляется структура документа, а CSS – это уже его оформление. При помощи CSS можно менять цвета, шрифты у текста, изменять положение элементов на странице, их размеры, задавать элементам рамки, границы и отступы.

Раньше, когда не было CSS, документы оформляли при помощи атрибутов, но возможности такого способа очень ограничены, поэтому сайты в то время были скучными и однотипными.

# Синтаксис СSS

|  |
| --- |
| Селектор {     свойство1: значение1;     свойство2: значение2; } |

Сначала обязательно указывается селектор, в роли которого могут выступать теги, идентификаторы, классы, атрибуты тегов. В фигурных скобках указываются свойства данного селектора в виде пары: название свойства и через двоеточие его значение. После каждой пары ставится точка с запятой, которая свидетельствует о разделении свойств. Если после последней пары свойство-значение, либо если эта пара одна, не поставить точку с запятой, ошибки не будет, но возьмите в привычку всегда ставить этот знак, так вы просто не будете про него забывать.

## Оформление CSS

1-й способ

|  |
| --- |
| Селектор {свойство1: значение1;свойство2: значение2;} |

2-й способ

|  |
| --- |
| Селектор  {     свойство1: значение1;     свойство2: значение2; } |

3-й способ

|  |
| --- |
| Селектор {     свойство1: значение1;     свойство2: значение2; } |

Первый способ, в котором необходимо записывать все свойства в одну строку, не очень удобен, т.к. свойств у селектора может быть много, они не уместятся на экран редактора. Соответственно, появится горизонтальная полоса прокрутки, и ваш лист стилей будет неудобно читать. Лучше использовать второй или третий способ оформления.

## Комментарии в CSS

|  |
| --- |
| */\* Внешний вид\*/* p { color: blue; } */\* Стили для параграфа \*/* |

В CSS можно также указывать комментарии для того, свойства каких элементов будут описываться, или для комментирования самих стилей при редактировании документа.

# Способы объявления CSS

Чтобы использовать стили CSS в веб-документе, необходимо их сначала подключить. Для этого существует три способа.

## Inline-стили

|  |
| --- |
| <body style="background:#0f0;">       <h1 style="color: blue;text-align: center;">             Заголовок       </h1> </body> |

Для подключения CSS этим способом в HTML существует тег style, который можно указывать практически у любого HTML-тега. В значении атрибута style перечисляются в том же формате стили свойств и их значений.

## Cтили в разделе head

|  |
| --- |
| <head>       <style type="text/css">       body {              color: blue;              background: #0f0;       }       h1 {              text-align: center;       }       </style> </head> |

Чтобы подключить стили этим способом, существует HTML-тег style. У него в атрибуте type указываем тип данных, в данном случае это text/css. Внутри этого тега прописываем стили, которые будут действовать для всей данной страницы.

## Внешний CSS-файл

Создаем файл с расширением .css. Обычно, как и в случае с картинками, все css-файлы размещают в отдельной папке.

**style.css**

|  |
| --- |
| body {        color: blue;        background: #0f0; } h1 {        text-align: center; } |

А в нужном HTML-файле этот файл подключаем.

**index.html**

|  |
| --- |
| <head>     <link rel="stylesheet" type="text/css" href="style.css"> </head> |

Для подключения CSS-файла используется тег link, который помещается в раздел head нужного HTML-файла. И для того, чтобы правильно подключить файл стилей, у тега link нужно указать несколько атрибутов. В атрибуте rel указывается значение stylesheet, т.е. лист стилей – это нужно для того, чтобы браузер понимал, что подключается файл стилей CSS. В атрибуте href указывается путь к CSS-файлу, причем, как и в случае с гиперссылками, этот путь может быть относительным и абсолютным. Указывается атрибут type со значением text/css.

## Какой способ подключения стилей выбрать?

Плюс inline-стилей в том, что можно быстро прописать какой-нибудь простой стиль для элемента, например, какое-то слово в тексте выделить красным цветом. Недостатки такого подхода в том, что для каждого тега необходимо прописывать стили. Допустим, у нас несколько параграфов, и все они должны быть определенного стиля, тогда каждому параграфу нужно прописывать этот стиль. Еще один существенный недостаток – при таком подходе стили сложно редактировать. Что делать, если нужно будет в проекте поменять размер шрифта во всех параграфах?

При втором способе уже можно прописывать стили для нескольких элементов, только все стили будут применяться в пределах одного документа. В этом и состоит главный минус этого подхода. Получается, что если на сайте большое количество страниц и у нас задача поменять тот же цвет или размер шрифта всех параграфов, нужно будет открывать каждую страницу. Если подключить отдельный файл, он будет действовать на все страницы, где мы подключим данный файл. И тогда, чтобы изменить цвет или размер шрифта всех параграфов, нужно будет изменить его один раз в одном месте. Еще одно преимущество в том, что браузер кеширует файл стилей, т.е. сохраняет его у себя в памяти, чтобы не обращаться при каждом запросе на сервер.

# Селекторы в CSS

## Селекторы тегов

|  |  |
| --- | --- |
| **html** | **css** |
| <h1>Для всех заголовков первого уровня цвет текста будет синим</h1> | h1 { color: blue; } |

При использовании селекторов тегов стиль будет применяться ко всем указанным тегам. В качестве селектора указывается название любого HTML-тега.

## Селекторы идентификаторов (id)

|  |  |
| --- | --- |
| **html** | **css** |
| <p id=”first”>Цвет фона данного параграфа будет серым</p> | #first { background: #ccc; } |

В качестве селекторов можно использовать идентификаторы. Определенному тегу в значении атрибута id указывается название, которое придумываем сами, а в селекторе ставится знак #, а затем это название. Очень важно запомнить следующее: идентификатор должен быть уникальным, т.е. нельзя задавать одно и то же имя двум и более элементам.

## Селекторы классов (class)

|  |  |
| --- | --- |
| **html** | **css** |
| <h1 class=”border”>Заголовок с рамкой</h1>  <p class=”border”>Параграф с рамкой</p> | .border { border: 1px solid black; } |

Классы используются аналогично id, только вместо атрибута id указывается атрибут class, а в селекторе вместо решетки – точка. Классы отличаются от идентификаторов тем, что можно применять один и тот же стиль к разным элементам.

## Селекторы атрибутов

В качестве селекторов можно указывать атрибуты HTML-тегов. Существует множество различных способов указывать селекторы атрибутов, но для начала рассмотрим 2 примера. Остальные способы вы сможете найти в справочнике, если они вам понадобятся.

|  |  |
| --- | --- |
| **html** | **css** |
| <img src="pic.jpg" alt="Фото"> <input type="text"> | img[alt] { width: 100px; } input[type="text"] { font-size: 10px; } |

В данном примере сначала указывается стиль для всех картинок, у которых присутствует атрибут alt. Для этого название атрибута указывается в квадратных скобках, сразу после названия тега.

Второй пример. Стиль будет применяться для всех тегов <input />, в значении атрибута type которого присутствует значение text, т.е. для всех обычных текстовых полей ввода.

# Свойства стилей

## Единицы измерения в CSS

В CSS существует достаточное количество единиц измерения, с помощью которых можно определять длину, ширину элементов, а также размеры шрифтов. Не все они используются в повседневной верстке, но нужно иметь представления о них. Единицы измерения подразделяются на относительные и абсолютные. Относительными называются единицы, которые могут изменяться в зависимости от различных факторов. К таким факторам относятся: разрешение монитора пользователя, ширина области просмотра (окна браузера), различные настройки пользователя. Относительные единицы измерения наиболее часто используются на веб-страницах.

## Относительные единицы измерения

* px – пиксель.
* % – процент.
* em – высота текущего шрифта.

### Пиксели

Пиксель px – это самая базовая, абсолютная и окончательная единица измерения. Количество пикселей задается в настройках разрешения экрана, один px – это как раз один такой пиксель на экране. Все значения браузер в итоге пересчитает в пиксели.

Пиксели могут быть дробными, например, размер можно задать в 16.5px. Это совершенно нормально, браузер сам использует дробные пиксели для внутренних вычислений. К примеру, есть элемент шириной в 100px, его нужно разделить на три части – волей-неволей появляются 33.333...px. При окончательном отображении дробные пиксели округляются и становятся целыми.

Для мобильных устройств, у которых много пикселей на экране, но сам экран маленький, браузер автоматически применяет масштабирование, чтобы обеспечить читаемость.

* Четкость и понятность.

       -    Другие единицы могут устанавливать соотношения между различными размерами.

### Проценты

Проценты %, как и em – относительные единицы. Когда мы говорим «процент», возникает вопрос: «Процент от чего?» Как правило, процент берется от значения свойства родителя с тем же названием, но не всегда. Это очень важная особенность процентов, про которую, увы, часто забывают.

**Относительно шрифта: em**

1em – текущий размер шрифта.

Можно брать любые пропорции от текущего шрифта: 2em, 0.5em и т.п.

Размеры в em относительные, они определяются по текущему контексту.

## Абсолютные единицы измерения

* cm – сантиметр.
* mm – миллиметр.
* in – дюйм.
* pt – пункт.
* pc – пика.

К абсолютным единицам относятся единицы измерения, которые используются в обычной жизни. Но они в веб-страницах применяются достаточно редко, поэтому использовать их крайне нежелательно.

## Способы задания цветов

Цвета в CSS можно задавать тремя различными способами. Первый способ – задавать цвета, используя их названия на английском языке, например: red, green, blue, black, yellow, white и т.д. Но при таком подходе есть ограничения в выборе цвета, невозможно получить различные оттенки цветов. Чтобы можно было выбрать один из более чем 16 млн цветов, нужно использовать способ выбора цвета: либо как функциональный RGB, либо шестнадцатеричный RGB. RGB – это аббревиатура, и расшифровывается она как Red-Green-Blue, то есть Красный-Зеленый-Синий. Таким образом, любой цвет можно получить, смешав эти три цвета.

### Функциональный RGB

RGB(255, 130, 0) 0 – 255

RGB(100%, 70%, 0%) 0 – 100%

В этом случае выбирается насыщенность любого из трех цветов в диапазоне от 0 до 255 или в процентах от 0 до 100%, и используется значение RGB, где в скобках через запятую перечисляется насыщенность каждого из трех цветов.

### Шестнадцатеричный RGB

#FA96CF; 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

#FFAA00 => #FA0

Если использовать шестнадцатеричный RGB, каждый цвет можно представить в виде пары значений, т.е. первый и второй символ – это красный цвет, третий и четвертый – это зеленый, пятый и шестой – синий. Каждый символ может быть представлен одним из шестнадцати знаков, от 0 до буквы F латинского алфавита. При шестнадцатеричном RGB перед кодом цвета ставится символ решетки, а дальше уже записывается сам код. Если совпадают две буквы или цифры одного и того же цвета, можно использовать сокращенную форму записи, т.е. каждый цвет будет состоять не из пары значений, а из одного, и в коде цвета будет 3 символа после знака решетки. При подборе цвета на первое время лучше использовать любой графический редактор. Там можно выбрать любой нужный вам цвет из палитры, и редактор покажет код выбранного цвета, который можно скопировать и вставить на страницу.

# Свойства стилей CSS

## Ширина и высота: width и height

|  |
| --- |
| p {      height: 200px;      width: 300px; } |

Можно задавать ширину и высоту в любых единицах измерения CSS. Если содержимое блока превышает указанную высоту, высота элемента останется неизменной, а содержимое будет отображаться поверх него. Из-за этой особенности может получиться наложение содержимого элементов друг на друга, когда элементы в коде HTML идут последовательно. Чтобы этого не произошло, добавьте overflow: auto к стилю элемента.

Для изменения размеров изображения средствами HTML предусмотрены атрибуты height и width. Допускается использовать значения в пикселях или процентах. Если установлена процентная запись, размеры изображения вычисляются относительно родительского элемента – контейнера, где находится тег <img>. При отсутствии родительского контейнера в его качестве выступает окно браузера. Иными словами, width="100%" означает, что рисунок будет растянут на всю ширину веб-страницы. Добавление только одного атрибута width или height сохраняет пропорции и соотношение сторон изображения. Браузер при этом ожидает полной загрузки рисунка, чтобы определить его первоначальную высоту и ширину.

## Фон элемента – background

|  |
| --- |
| background-color: #ff0; background-image: url(img/photo.jpg); background-position: top; (bottom | left | right) background-repeat: repeat-x; (repeat-y | no-repeat) background-attachment: fixed; |

* background-color задает цвет фона, который можно задавать любым из трех способов задания цветов.
* background-image используется для того, чтобы в качестве фона можно было установить изображение. Для этого необходимо в значении свойства указать путь к изображению в скобках url.
* background-position указывает, где будет располагаться фоновое изображение. Может иметь значения: top, bottom, left, right.
* background-repeat определяет, нужно ли повторять фоновое изображение: repeat-x – изображение повторяется по горизонтали, repeat-y – по вертикали, no-repeat – изображение не повторяется. По умолчанию у этого свойства установлено значение repeat, что означает, что изображение будет повторяться по горизонтали и по вертикали.
* background-attachment определяет, будет ли изображение прокручиваться вместе с содержимым элемента. По умолчанию оно установлено как scroll, что означает, что изображение будет прокручиваться, а при значении fixed изображение будет оставаться неподвижным.

Существует короткая форма записи, в которой можно записать все перечисленные значения в одну строку, разделяя значения пробелом. Если пропускать какие-либо значения, будут подставляться значения по умолчанию.

|  |
| --- |
| background: #ff0 url(img/photo.jpg) top repeat-x; |

## border – рамка вокруг элемента

|  |
| --- |
| border-color: red; (#f00 | RGB(255, 0, 0)) border-style: solid; (dotted | dashed | groove | ridge | solid | double | inset | outset) border-width: 2px; |

border тоже подразделяется на различные свойства:

* border-color – цвет рамки.
* border-style – стиль рамки, которая может быть разных значений: dotted, dashed, solid, double, groove, ridge, inset, outset.
* border-width задает толщину рамки, причем ее можно задать для каждой из 4 сторон отдельно:

(1px 2px) – 1px: верхняя и нижняя, 2px: левая и правая.

(1px 2px 3px) – 1px: верхняя, 2px: левая и правая, 3 нижняя.

(1px 2px 3px 4px) – 1px: верхняя, 2px: правая, 3px: нижняя, 4px: левая.

Также возможно перечислять свойства в одну строку, разделяя пробелом. В этом случае тоже не важен порядок следования свойств.

|  |
| --- |
| border: 1px solid black; |

Есть возможность каждую границу задавать отдельно, когда необходима, к примеру, только одна граница.

|  |
| --- |
| border-top: 2px dotted green; border-bottom: 3px double blue;  border-left: 1px solid red;  border-right: 4px inset #000; |

## Цвет текста – color

|  |
| --- |
| color: red; color: #78fa2e; color: RGB(34, 21, 56); |

Цвет текста также можно задавать любым из 3 способов.

## Шрифт – font

## font-family устанавливает шрифт текста.

|  |
| --- |
| font-family: "Times New Roman", serif, Verdana; |

* serif – шрифты с засечками.
* sans-serif – рубленые шрифты, без засечек.
* cursive – курсивные шрифты.
* fantasy – декоративные шрифты.
* monospace – моноширинные шрифты.

Существует 5 основных семейств шрифтов. У каждого семейства существуют несколько видов шрифтов. Какие шрифты относятся к какому семейству, можно узнать из справочников. Можно через запятую указывать несколько шрифтов. Первым будет использоваться шрифт Times New Roman, если по каким-либо причинам он не установлен на компьютер, то будет отображаться следующий шрифт. Если название шрифта состоит из нескольких слов, его заключают в кавычки.

|  |
| --- |
| font-style: italic; (oblique | normal) font-variant: small-caps; font-weight: bold; (bolder | lighter| 100 | 200); font-size: 20px; (small | medium | large); |

* font-style – стиль шрифта. По умолчанию установлен шрифт в значении normal. italic – это курсивное начертание, которое имитирует рукописный текст, а oblique – наклонное начертание, которое получается путем наклона знаков вправо.
* font-variant имеет только 2 значения. По умолчанию установлено значение normal и small-caps, которое у строчных букв имитирует заглавные буквы, только уменьшенного размера.
* font-weight задает насыщенность шрифта. Можно указывать значения предопределенными словами, например, bold – полужирный, bolder – жирный, lighter – светлый. Еще есть возможность указывать насыщенность цифрами от 100 до 900.
* font-size определяет размер шрифта. Можно указывать в любых единицах измерения или предопределенными словами. Указывать стиль шрифта можно при помощи сокращенной записи. В данном случае важен порядок следования значений.

|  |
| --- |
| font: font-style       font-variant       font-weight       font-size       font-family; font: bold 24px Arial, Verdana; |

## Оформление списков – list-style

|  |
| --- |
| list-style-type: circle; (disc | square | armenian | decimal) list-style-position: inside; list-style-image: url(img/list.png); |

Свойство list-style определяет стиль маркера у списков.

* list-style-type – тип маркера, который может быть разных видов, в примере приведены только некоторые из них. Остальные виды маркеров можно найти в справочнике.
* List-style-position определяет, где располагается маркер, по умолчанию у него значение outside. В этом случае маркеры будут располагаться за пределами текстового блока. При значении inside – наоборот, внутри текстовых блоков.
* list-style-image позволяет вместо маркера установить изображение, для этого нужно указать к нему путь в скобках url.

 Для определения стиля маркеров также существует сокращенная запись.

|  |
| --- |
| list-style: list-style-type             list-style-position             list-style-image; list-style: circle inside; |

|  |
| --- |
| text-align: center; (justify | left | right) text-decoration: none; (line-through | overline | underline | none) text-transform: capitalize; (lowercase | uppercase) |

Еще некоторые полезные свойства.

* text-align – выравнивание содержимого блока по горизонтали. Принимает 4 значения: left, right, center и justify (выравнивание происходит по ширине, т.е. одновременно по левому и по правому краю).
* text-decoration применяется для следующего оформления текста: line-through – перечеркивает текст, overline – задает линию над текстом, underline – задает линию под текстом (подчеркивает текст), none (по умолчанию) – отменяет все эффекты.
* text-transform используется для изменения регистра символов. При значении capitalize каждое слово в предложении будет начинаться с заглавной буквы, при значении lowercase все символы будут строчными, а при значении uppercase все символы будут заглавными.

# Вложенность

При изучении тегов HTML мы рассматривали, что можно вкладывать одни HTML теги в другие. А при помощи CSS есть возможность управлять различными вложенными конструкциями. Для управления вложенностью в CSS существует несколько специальных селекторов. Рассмотрим их на примерах.

## Контекстные селекторы

|  |  |
| --- | --- |
| **html** | **css** |
| <p class=”main”>В этом параграфе  <strong><a href=“#”>эта ссылка</a></strong> будет размером 18px и красного цвета,  <a href=”#”>а эта будет обычной</a>.  </p> | .main strong a {       font-size: 18px;       color: red; } |

В этом примере можно увидеть параграф с классом main. В параграф вложена ссылка. Этой ссылке задаем определенный стиль. Обратимся к этой ссылке при помощи контекстного селектора. Для этого в качестве селектора сначала указывается тег параграфа с классом main, затем ставится пробел и следующим указывается тег strong, после этого обращаемся к тегу нужной нам ссылки. Таким образом, заданный стиль будет применяться ТОЛЬКО к первой ссылке в параграфе с классом main.

|  |  |
| --- | --- |
| **html** | **css** |
| <p class=”main”>В этом параграфе  <strong><a href=“#”>эта ссылка</a></strong> будет размером 18px и красного цвета,  <a href=”#”>а эта будет обычной</a>.  </p> | .main a {       font-size: 18px;       color: red; } |

Теперь из контекстного селектора убираем тег strong. В этом случае обе ссылки из параграфа с классом main приобретут заданный стиль, т.е. станут красного цвета с размером шрифта в 18 px. Запись данного контекстного селектора означает, что нужно применить стиль ко всем тегам ссылки, которые находятся внутри параграфов с классом main.

## Дочерние селекторы

|  |  |
| --- | --- |
| **html** | **css** |
| <p class=”main”>  <a href=“#”>ссылка 1</a>  </p>  <p class=”main”>  <i><a href=“#”>ссылка 2</a></i>  </p> | .main > a {      font-size: 18px;      color: red; } |

При помощи дочерних селекторов можно выбрать только те теги, которые являются прямыми потомками определенного элемента. В этом примере в первом параграфе ссылка является дочерним элементом для параграфа, т.к. вложена только в параграф. А во втором параграфе ссылка для параграфа не будет являться дочерним элементом, т.к. она вложена в тег i, и, соответственно, будет являться дочерним элементом для тега i. А тег i для этой ссылки будет родительским элементом. В разделе CSS кода после обращения к селектору параграфа р с классом main ставится знак больше, а затем тег а. Это означает, что необходимо обратиться к дочернему тегу ссылки параграфа с классом main. В этом примере такому запросу соответствует ссылка в первом параграфе, и ТОЛЬКО к данной ссылке будет применен заданный стиль.

## Соседние селекторы

|  |  |
| --- | --- |
| **html** | **css** |
| <p class=”main”>  <a href=“#”>ссылка 1</a>  </p>  <p class=”main”>  <i>Соседний элемент</i>  <a href=“#”>ссылка 2</a>  </p> | .main i + a {       font-size: 18px;       color: red; } |

Для демонстрации работы соседних селекторов изменим предыдущий пример. Ссылка 2 теперь будет не вложена в тег i, а находиться рядом, т.е. будет являться для тега i соседним элементом. Чтобы обратиться к этой ссылке и задать ей определенный стиль, в разделе CSS кода после тега i надо поставить знак плюс, а затем указать тег ссылки. В этом случает ТОЛЬКО ссылка 2 приобретет заданный стиль.

## Наследование

Наследование – это перенос стилей от элемента к вложенным в него тегам.

|  |  |
| --- | --- |
| **html** | **css** |
| <p>В этом параграфе весь текст <b>**будет**</b>красного  цвета и шрифтом <b><i>***размером 18 px***</i><b>  </p> | p {     font-size: 18px;     color: red; } |

В данном примере в тег <p> вложены два тега <b>, в один из которых также вложен тег <i>. В этом случае весь текст в этом параграфе будет заданного стиля. То есть все теги, вложенные в параграф, унаследовали заданный стиль. Но не все свойства CSS наследуются.

|  |  |
| --- | --- |
| **html** | **css** |
| <p>В этом параграфе весь текст <b>**будет**</b>красного цвета и шрифтом <b><i>***размером 18 px***</i></b>, только  <a href=“#”>эта ссылка</a> не будет красной  </p> | p {     font-size: 18px;     color: red; } |

Если добавить в этот пример еще и ссылку, то эта ссылка унаследует только свойство font-size, но свойство color не унаследует. Узнать, наследуется ли определенное свойство CSS, можно только из справочников. В данном примере, чтобы применить к ссылке свойство color так же, как у всех остальных элементов, можно для этой ссылки задать значение inherit для свойства color. inherit означает, что элементу необходимо наследовать данное свойство от родителя.

## Группировка свойств

Группировку свойств необходимо использовать, когда для разных элементов заданы одинаковые стили. И в этом случае надо стараться избегать повторения кода.

|  |  |
| --- | --- |
| **html (без группировки свойств)** | **html (правильный вариант)** |
| h1 {  text-align: center;  color: blue;  font-family: Verdana;  }  h3 {  text-align: center;  color: blue;  font-family: Arial;  }  p {  text-align: center;  color: blue;  font-size: 12px;  } | h1, h3, p {  text-align: center;  color: blue;  }  h1 {  font-family: Verdana;  }  h3 {  font-family: Arial;  }  p {  font-size: 12px;  } |

Чтобы свойства сгруппировать, надо через запятую перечислить те селекторы, для которых будет нужно сгруппировать свойства, затем перечислить повторяющиеся стили. Дальше уже для каждого элемента задать уникальные стили.

# Приоритеты стилей в CSS

Вы можете столкнуться с ситуацией, когда при разработке сайтов вы задаете определенное свойство какому-нибудь элементу, а это свойство не работает, т.е. элемент не приобретает заданный стиль. Это происходит потому, что где-то уже был установлен определенный стиль этому элементу. Чтобы решить эту проблему и задать нужный стиль, нужно знать приоритеты применения стилей. Существует такое понятие, как каскадирование, которое применяется тогда, когда одному и тому же элементу пытаются присвоить разные стили. Например, мы всем параграфам пытаемся присвоить сначала черный цвет, а потом зеленый. Какое правило должно тогда примениться?

|  |
| --- |
| h1 {      color: black; } h1 {      color: green; } |

В данном случае все параграфы будут зелеными, потому что по правилам, если одинаковому селектору присваивать одинаковые свойства, то применится тот стиль, который стоит ниже.

## Приоритеты источников стилей

1. Стиль автора документа обладает самым высоким приоритетом. Этот стиль задает сам разработчик сайта.
2. Стиль, заданный пользователем в настройках браузера. Стиль CSS может задать конечный пользователь этого сайта, если подключит свой собственный файл стилей. Этот источник является менее приоритетным.
3. Стиль самого браузера, т.е. тот стиль, который определен в настройках самого браузера. Это источник обладает самым низким приоритетом.

## Приоритеты стилей автора

Рассмотрим приоритеты стилей автора проекта. Самым важным свойством является то, у которого после значения свойства установлена директива !important.

|  |
| --- |
| h1 {      color: black!important; }  h1 {      color: green; } |

Добавим свойству первого параграфа из предыдущего примера директиву !important. Теперь в этом случае у всех заголовков H1 цвет текста будет черным, несмотря на то, что это объявление свойства стоит первым. Если эту директиву применит пользователь в своем собственном файле стилей, то этот стиль станет важнее стиля автора. Это нужно для того, чтобы люди с ограниченными возможностями могли устанавливать свои стили, которые будут важнее всех.

Вторым по приоритетности является стиль, объявленный в атрибуте style любого тега.

|  |  |
| --- | --- |
| **html** | **css** |
| <h1 style=”color: red;”>  Заголовок первого уровня  </h1> | h1 {      color: green; } |

В данном примере цвет текста заголовка первого уровня будет красным, так как этот стиль переопределен в атрибуте style.

Следующий уровень – это уровень приоритета селекторов. Существует такое понятие, как специфичность. Смысл его в том, что браузер будет начислять определенное количество баллов за разные типы селекторов, а также их количество. И больший приоритет получают те стили, которые набирают большее количество баллов.

* Селекторы тегов и псевдоэлементы – по 1 баллу (0, 0, 0, 1).
* Селекторы атрибутов, классы и псевдоклассы – по 10 баллов (0, 0, 1, 0).
* Идентификаторы – по 100 баллов (0, 1, 0, 0).
* Атрибут style – 1000 баллов (1, 0, 0, 0).

Пример начисления баллов за специфичность:

* p { } – 1 балл (селектор тегов).
* p:first-letter { } – 2 балла (1 – селектор тегов и 1 – псевдоэлемент).
* input[type=“submit"] { } – 11 баллов (по 1 селектору тегов и атрибутов).
* div.head .new { } – 21 балл (2 класса и 1 селектор тегов).
* #header a:hover { } – 111 баллов (идентификатор, селектор тегов и псевдокласс).

Следующие по приоритетности стили указаны в порядке убывания:

* Стили, заданные в разделе <head>.
* Стили, подключаемы из внешних файлов.
* Наследуемые стили от предков.

# Практика

## Создание стилей для меню сайта

|  |  |
| --- | --- |
| <!-- Меню-->  <ul class=”menu”>      <li>Главная</li>      <li><a href="catalog.html">Каталог</a></li>      <li><a href="contacts.html">Контакты</a></li>  </ul> | body {     background-color: #f8f8f8; } .menu li {     font-style: italic;     font-size: 20px;     list-style-type: none; } .menu li a {     color: #8B00FF; } |

# Практическое задание

Вы будете дорабатывать задание второго урока, придавая ему стили CSS.

1. Создать файл style.css, в котором будут храниться все стили вашей работы. Подключить этот файл ко всем страницам.
2. В качестве фона на всех страницах установить цвет #f8f8f8.
3. Меню сайта:
   1. Для всех ссылок меню задать определенный стиль (цвет текста, размер шрифта, начертание шрифта и т.д.)
   2. Убрать маркеры списка.
4. Страница «Подробное описание товара»:
   1. Заголовки (Краткое описание товара, Характеристики, Подробное описание товара):
      1. Цвет текста черный.
      2. Размер шрифта 18px.
      3. Насыщенность шрифта 400 (font-weight).
      4. Цвет фона #eaeaea.
   2. Для текста краткого описания товара:
      1. Цвет текста #707070.
      2. Размер шрифта 14px.
      3. Начертание шрифта italic (font-style).
      4. Высота текста 16px (line-height).
   3. Для текста подробного описания товара:
      1. Цвет текста  #484343.
      2. Размер шрифта 16px.
      3. Насыщенность шрифта 400 (font-weight).
      4. Высота текста 24px (line-height).
      5. Расположение текста по левому краю (text-align).
   4. Для списка внурти подраздела «Характеристики товара»:
      1. Задайте списку стили, отличные от всего остального текста.
      2. \* Установите в качестве маркеров произвольные изображения.
5. Странице «Контакты»:
   1. Задать значения ширины и высоты для полей ввода.
   2. Задать стили для текста внутри полей input (цвет текста, размер шрифта и т.д.).
   3. Следите за тем, чтобы на странице все выглядело гармонично, не выбирайте слишком резких цветов.
6. \* Для изображений, размещенных на странице подробного описания товара, задать рамку произвольного цвета.
7. \* Приветствуются самостоятельные дополнения ваших работ в пределах пройденной темы.

# Дополнительные материалы

[Статья про добавление стилей на страницу](http://css.manual.ru/articles/adding_css_to_a_web_page).

[Интересная статья про селекторы](https://learn.javascript.ru/css-selectors).

[30 CSS-селекторов, о которых полезно помнить](http://codeharmony.ru/materials/42).

[Вложенность селекторов](http://site-on.net/create/html/5-selectori-css-tochechnoe-primenenie-svoistv-css-optimizaciya-html).

[Группировка и вложенность селекторов](http://technologyweb.org/%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B8-%D0%B2%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C/).

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/samcss/sposoby-dobavleniya-stiley-na-stranitsu>.
2. <http://htmlbook.ru/samcss/dochernie-selektory>.
3. <https://learn.javascript.ru/css-selectors>.
4. <https://learn.javascript.ru/css-units>.
5. <http://technologyweb.org/%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B8-%D0%B2%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C/>.

Псевдоклассы и псевдоэлементы, табличная верстка

Псевдоклассы и псевдоэлементы. Создание таблиц. Объединение ячеек. Вложенные таблицы. Стилевое оформление таблиц.

[Псевдоклассы и псевдоэлементы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.5v83ltpdsujj)

[Псевдоклассы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.jtuq5c5y9l3m)

[Псевдоклассы, определяющие состояние элементов](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.dey8qkng34o8)

[Структурные псевдоклассы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.ax95oi8jgriy)

[Псевдоклассы по типу дочернего элемента](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.necrc3u5xq6q)

[Пример использования псевдоклассов](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.eo82qigoo8w3)

[Комбинирование псевдоклассов](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.418ytxrf3kkz)

[Псевдоэлементы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.ytqvmv91w6f)

[Пример использования псевдоэлементов](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.5dm0ax2xkh8k)

[Для чего нужны таблицы в HTML](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.nyyp3ll3l7u3)

[Структура таблицы в HTML](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.uvcsltxlq7b)

[Создание ячеек таблицы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.i402ugj4712l)

[Теги группирования строк HTML-таблицы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.tsusinu0jsjo)

[Группировка строк и столбцов таблицы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.2gahdjr0atme)

[Объединение ячеек таблицы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.sv5wrg7vbubb)

[Вложенные таблицы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.mqnarn7pw52a)

[Стилевое оформление таблиц](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.505l1wujgy1u)

[Основные теги для верстки](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.c5d9qfauyv0k)

[Блочные элементы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.2fzj0m80vl0g)

[Строчные элементы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.adjbwa43pupu)

[Практика](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.vg3uy1csazbk)

[Создание стилей для меню сайта](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.69nq3b6y1irv)

[Практическое задание](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.rnwyxlqq3x23)

[Дополнительные материалы](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1YB1LAfccxcNet03WADqHjaM4eJG91rlVDCN9dL40TY8/edit#heading=h.v65nmrqa6w6p)

**Псевдоклассы и псевдоэлементы**

Псевдоклассы и псевдоэлементы позволяют назначить CSS-стили структурам, существование которых на веб-странице не обязательно. Т.е. стили применяются к частям документа не на основании той информации, которая содержится в его структуре, и даже изучив эту структуру, невозможно понять, что к этим элементам применяются какие-либо стили.

Псевдоклассы

Псевдоклассы позволяют добавлять особые классы к элементам, выбирая объекты, которых нет в структуре веб-страницы либо которые нельзя выбрать с помощью обычных селекторов, например, первая буква или первая строка одного абзаца.

Вы наверняка замечали на многих сайтах: когда вы наводите мышкой на какой-либо пункт меню, он меняет свой вид, у него может изменяться цвет фона, цвет ссылки, даже шрифт или его размер. Это происходит благодаря псевдоклассам. Рассмотрим их синтаксис.

|  |
| --- |
| Селектор:псевдокласс {       свойство1: значение1; } a:hover {      color: #ccc; } |

После селектора ставится двоеточие, и сразу после него без пробела указывается название псевдокласса.

Псевдоклассы, определяющие состояние элементов

a:link – ссылается на непосещенную ссылку.

a:visited – ссылается на уже посещенную ссылку.

a:hover – ссылается на любой элемент, по которому проводят курсором мыши.

a:focus – определяет стиль поля, получающего фокус (например, текстового поля формы, в которое устанавливается курсор).

a:active – ссылается на элемент, который был активизирован пользователем.

:valid – выберет поля формы, содержимое которых прошло проверку в браузере на соответствие указанному типу.

:invalid – выберет поля формы, содержимое которых не соответствует указанному типу.

:enabled – выберет все доступные (активные) поля форм.

:disabled – выберет заблокированные поля форм, т.е. находящиеся в неактивном состоянии.

:in-range – выберет поля формы, значения которых находятся в заданном диапазоне.

:out-of-range – выберет поля формы, значения которых не входят в установленный диапазон.

:lang() – выбирает абзацы на указанном языке.

:not(селектор) – выберет элементы, которые не содержат указанный селектор, например класс, идентификатор или селектор элемента.

:not([type="submit"]). :target – выбирает элемент с символом #, на который ссылаются в документе.

 :checked – выбирает выделенные (выбранные пользователем) элементы.

Структурные псевдоклассы

:nth-child(odd) – выбирает нечетные дочерние элементы.

:nth-child(even) – выбирает четные дочерние элементы.

:nth-child(3n) – выбирает каждый третий элемент среди дочерних.

:nth-child(3n+2) – выбирает каждый третий элемент, начиная со второго дочернего элемента (+2).

:nth-child(n+2) – выбирает все элементы, начиная со второго.

:nth-child(3) – выбирает третий дочерний элемент.

:nth-last-child() – в списке дочерних элементов выбирает элемент с указанным местоположением, аналогично с :nth-child(), но начиная с последнего, в обратную сторону.

:first-child – позволяет оформить только самый первый дочерний элемент тега.

:last-child – позволяет форматировать последний дочерний элемент тега.

:only-child – выбирает элемент, являющийся единственным дочерним элементом.

:empty – выбирает элементы, у которых нет дочерних элементов.

:root – выбирает элемент, являющийся корневым в документе (элемент html).

Псевдоклассы по типу дочернего элемента

:nth-of-type() – выбирает элементы по аналогии с :nth-child(), при этом берет во внимание только тип элемента.

:first-of-type – позволяет выбрать первый дочерний элемент.

:last-of-type – выбирает последний тег конкретного типа.

:nth-last-of-type() – выбирает элемент заданного типа в списке элементов в соответствии с указанным местоположением, начиная с конца.

:only-of-type – выбирает единственный элемент указанного типа среди дочерних элементов родительского элемента.

Пример использования псевдоклассов

|  |  |
| --- | --- |
| **HTML** | **CSS** |
| <ul>      <li>Первый элемент</li>      <li>Второй элемент</li>      <li>Третий элемент</li>  </ul> | li:first-child{     font-size: 24px;      color: #F23401;  } |

Чтобы понять, как работает данный псевдокласс, рассмотрим простой пример. Задаем элементу списка <li> псевдокласс first-child, и у него прописываем определенный стиль. Как видно, маркированный список состоит из трех элементов. Заданный стиль будет применен ТОЛЬКО к первому элементу данного списка. Это происходит потому, что первый элемент списка <li> будет именно первым дочерним у тега <ul>.

Комбинирование псевдоклассов

Псевдоклассы можно комбинировать в одном селекторе, просто перечисляя их через двоеточие, как показано в следующем примере.

|  |
| --- |
| /\* При наведении на не посещенную ссылку цвет текста будет зеленым \*/  a:link:hover {      color: #0F0;  }  /\* При наведении на посещенную ссылку цвет текста будет красным\*/  a:visited:hover {      color: #F00;  } |

**Псевдоэлементы**

Псевдоэлементы – практически то же самое, что и псевдоклассы, только они позволяют ввести несуществующие элементы в веб-документ и придать им определенные стили. Псевдоэлементы появились еще в CSS1, но пошли в релиз только в CSS2.1. В самом начале в синтаксисе использовалось одно двоеточие, но в CSS3 используется двойное двоеточие для отличия от псевдоклассов. Современные браузеры умеют понимать оба типа синтаксиса псевдоэлементов, кроме Internet Explorer 8, который воспринимает только одно двоеточие. Поэтому надежнее использовать одно. С помощью свойства content можно изменить внешний вид части элемента.

 В CSS существует 4 псевдоэлемента:

:first-letter – выбирает первую букву каждого абзаца, применяется только к блочным элементам.

:first-line – выбирает первую строку текста элемента, применяется только к блочным элементам.

:before – вставляет генерируемое содержимое перед элементом.

:after – добавляет генерируемое содержимое после элемента.

Пример использования псевдоэлементов

|  |  |
| --- | --- |
| **HTML** | **CSS** |
| <ul>      <li>Первый элемент</li>      <li>Второй элемент</li>      <li>Третий элемент</li>  </ul> | li:after{      content: "new";      color: #F00;  } |

После элемента списка li появится текст new красного цвета.

**Для чего нужны таблицы в HTML**

Основное назначение таблиц в HTML – представление табличных данных, таких как: просмотр информации о пользователях, просмотр заказанных товаров в интернет-магазине, просмотр отчетов о продажах и др. Второе назначение таблиц в HTML таково, что при помощи их можно верстать веб-страницы. А верстать – это значит разбивать страницу на элементы, т.е. формировать различные блоки сайта, такие как шапка, меню, контент, подвал и другие. Табличный способ верстки в настоящее время считается неправильным и устаревшим, потому что таблицы используются не по назначению, но многие сайты либо уже сверстаны при помощи таблиц, либо продолжают верстаться этим способом. Поэтому разработчику сайтов важно уметь пользоваться как табличным способом верстки, так и версткой слоями. Оба способа верстки мы рассмотрим в данном курсе.

Структура таблицы в HTML

|  |
| --- |
| <table>       <tr>             <td>Столбец 1</td>             <td>Столбец 2</td>             <td>Столбец 3</td>         </tr> </table> |

Любая таблица в HTML помещается в контейнер <table>, после чего в контейнер строки <tr> помещаются уже столбцы таблицы, которые помещаются в тег <td>. В данном примере таблица будет состоять из трех столбцов и одной строки, т.е. в таблице будет 3 ячейки. Чтобы добавить еще одну строку в данную таблицу, нужно вложить в контейнер <table> еще один контейнер строки <tr>, и в него поместить то же количество столбцов <td>, т.е. 3 столбца.

|  |
| --- |
| <table>       <tr>             <td>Столбец 1</td>             <td>Столбец 2</td>             <td>Столбец 3</td>        </tr>       <tr>             <td>Столбец 1</td>             <td>Столбец 2</td>             <td>Столбец 3</td>        </tr> </table> |

Создание ячеек таблицы

Элемент <td> создает ячейки таблицы с данными, добавляя их в строку таблицы. Парные теги <td>…</td>, расположенные между тегами соответствующей строки, определяют количество ячеек в пределах одной строки. Количество пар ячеек таблицы должно быть равным количеству пар ячеек заголовка.

Элемент <th> создает заголовок – специальную ячейку, текст в которой выделяется полужирным шрифтом. Количество ячеек заголовка определяется количеством пар тегов <th>…</th>.

Теги группирования строк HTML-таблицы

Для создания более сложных таблиц можно использовать теги:

* <caption>...</caption> – обрамление заголовка таблицы (текст, размещенный между этими тегами, выравнивается по центру). В исходном коде тег <caption> ставится сразу после тега <table>.
* <thead>...</thead> – обрамление шапки таблицы (в одной таблице допускается не более одного тега <thead>).
* <tbody>...</tbody> – группирует строки таблицы в один табличный блок (таких блоков в таблице может быть несколько).
* <tfoot>...</tfoot> – обрамление футера, т.е. нижней, завершающей части таблицы (в одной таблице допускается не более одного тега <tfoot>). В исходном коде тег <tfoot> ставится до тега <tbody>.

Группировка строк и столбцов таблицы

Элемент <colgroup> создает структурную группу столбцов, выделяя логически однородные ячейки. Группирует один или более столбцов для форматирования, позволяя применить стили к столбцам, вместо того чтобы повторять стили для каждой ячейки и для каждой строки. Добавляется непосредственно после тегов <table> и <caption>. Элемент <col> формирует неструктурные группы столбцов, которые делят таблицу на разделы, не относящиеся к общей структуре, т.е. не содержащие информацию одного типа. Позволяет задавать свойства столбцов для каждого столбца в пределах элемента <colgroup>. С помощью атрибута <style> можно изменить основной цвет фона ячеек. Для элемента <col> доступен атрибут span, задающий количество столбцов для объединения.

Объединение ячеек таблицы

Чтобы в полной мере начать использовать таблицы в HTML, необходимо научиться объединять ячейки. Для этого у тега <td> или <th> существуют атрибуты colspan и rowspan.

* colspan объединяет ячейки по горизонтали.
* rowspan объединяет ячейки по вертикали.

|  |  |  |
| --- | --- | --- |
| **rowspan** – объединение по вертикали (строк) | **colspan** – объединение по горизонтали (столбцов) | |
|  |  |
|  |  |  |

В данном примере при помощи атрибута rowspan в первом столбце объединяются 2 строки. При помощи атрибута colspan в первой строке объединяются уже два столбца.

Посмотрим, как это выглядит в HTML.

|  |
| --- |
| <table>     <tr>         <td rowspan="2">Столбец 1</td>          <td colspan="2">Столбец 2</td>      </tr>     <tr>         <td>Столбец 5</td>          <td>Столбец 6</td>      </tr>     <tr>         <td>Столбец 7</td>          <td>Столбец 8</td>          <td>Столбец 9</td>      </tr> </table> |

В первой строке данной таблицы будет всего два столбца, в первом столбце первая строка объединяется со второй строкой, поэтому у первой ячейки указываем атрибут rowspan со значением 2, что означает, что нужно объединить две ячейки. В этой же строке во втором и третьем столбце объединяем две ячейки по горизонтали при помощи атрибута colspan, также со значением 2. Далее во второй строке нужно учесть, что не надо указывать для нее первый столбец, т.к. он уже объединен с первой строкой. Поэтому во второй строке указываются два столбца, начиная со второго. В третьей строке присутствуют все три столбца по порядку.

Вложенные таблицы

При помощи объединения ячеек можно сделать каркас будущего сайта, т.е. макет сайта будет состоять из одной большой таблицы. А уже в эту таблицу можно вкладывать другие таблицы, например, вложить галерею изображений, которую тоже можно сделать в виде таблицы, т.е. каждое изображение будет находиться в отдельной ячейке. Или использовать таблицу по ее прямому назначению, для представления табличных данных. Для этого можно вложить в определенную ячейку еще одну таблицу, соблюдая ту же структуру.

|  |
| --- |
| <table>  <tr>  <td>  <table>  <tr>  <td>Столбец 1</td>  <td>Столбец 2</td>  <td>Столбец 3</td>  </tr>  </table>  </td>  <td></td>  <td></td>  </tr>  </table> |

**Стилевое оформление таблиц**

По умолчанию таблица и ячейки таблицы не имеют видимых границ и фона, при этом ячейки внутри таблицы не прилегают вплотную друг к другу. Ширина ячеек таблицы определяется шириной их содержимого, поэтому может быть разной. Высота всех ячеек сетки строки по умолчанию одинаковая.

Границу таблице можно задавать аналогично любому другому элементу, это мы рассматривали на прошлом занятии. Единственное, если задать свойство border только тегу <table>, то граница будет задана только внешним границам таблицы. Это происходит потому, что свойство border не наследуется. Чтобы задать границы для ячеек, нужно задать их тегу <td>. Здесь уже можно задавать границы ячейкам любого стиля.

|  |
| --- |
| /\*Внешняя граница таблицы\*/  table {  border: 1px solid #000;  }  /\*Границы для ячеек таблицы\*/  td {  border: 1px solid #000;  } |

Ширину и высоту таблицы задаем при помощи свойств CSS width и height.

|  |
| --- |
| table {       width: 400px;       height: 200px; } |

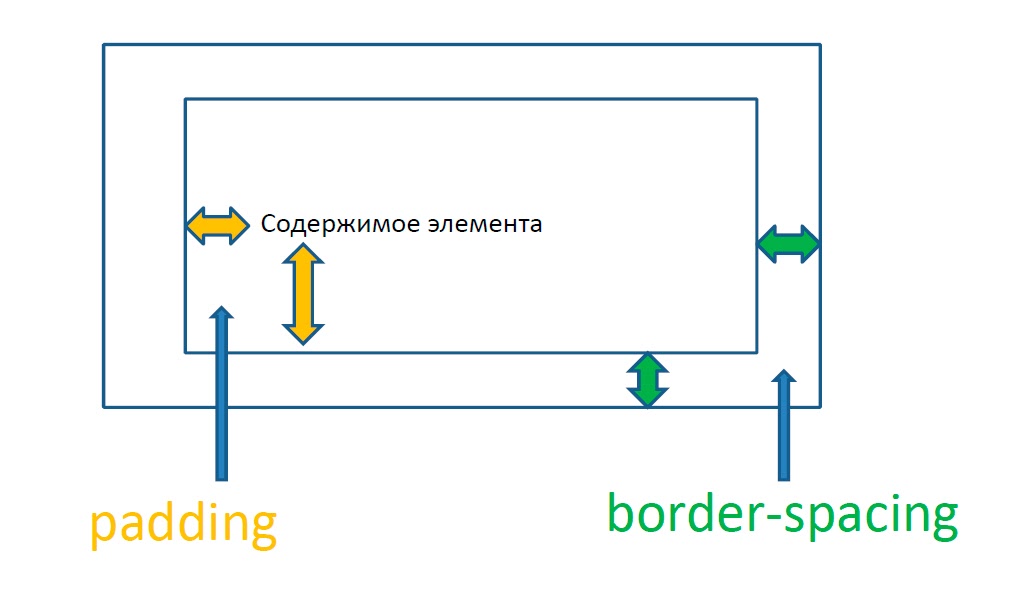
По умолчанию ширина и высота таблицы определяются содержимым ее ячеек. Если не задать ширину, то она будет равна ширине самого широкого ряда.

Ширину таблицы и ее столбцов можно задать с помощью свойства width. Если для таблицы задана ширина table {width: 100%;}, то она будет равна ширине блока-контейнера, а ширина столбцов установится в соответствии с шириной содержимого ячеек. Чаще всего ширину таблицы и столбцов задают в px или %.

Высота таблицы обычно не задается. Высотой строк таблицы можно управлять, добавив верхний и нижний padding или же задав высоту с помощью свойства height.

Для задания отступов элементу в CSS существуют свойства padding и border-spacing.

* border-spacing – внешние отступы.
* padding – внутренние отступы.



Внешние отступы (border-spacing) можно задать либо всем 4 сторонам, либо отдельно. В этом случае первое значение будет действовать для левой и правой сторон, а второе – для верхней и нижней. Внутренние отступы (padding) можно задавать и для каждой границы отдельно.

(1px 2px) – px: верхняя и нижняя, 2px: левая и правая.

(1px 2px 3px) – 1px: верхняя, 2px: левая и правая, 3 нижняя.

(1px 2px 3px 4px) – 1px: верхняя, 2px: правая, 3px: нижняя, 4px: левая.

Для выравнивания по горизонтали внутри ячейки существует устаревший атрибут align, который заменяется на свойство CSS text-align.

|  |
| --- |
| td {      text-align: center; (right | justify | left) } |

Для вертикального выравнивания внутри ячейки у тега <td> существует устаревший атрибут valign. Для его замены в CSS существует свойство vertical-align.

|  |
| --- |
| td {      vertical-align: top; (bottom | baseline | middle) } |

Это свойство в отношении ячеек таблицы может принимать четыре значения:

* top – выравнивание по верхнему краю.
* bottom – выравнивание по нижнему краю.
* baseline – базовая линия ячейки выравнивается по базовой линии первой текстовой строки.
* middle – выравнивание по середине (по умолчанию).

Есть ещё одно полезное свойство для оформления таблиц. Если для всех ячеек задать атрибут cellspacing=”0” или CSS свойство border-spacing=”0”, то границы соседних ячеек будут двойными. Для того, чтобы убрать двойные границы, в CSS существует свойство border-collapse.

|  |
| --- |
| table {        border-collapse: collapse; } |

Свойство border-collapse может принимать два значения. По умолчанию установлено separate, при котором рамка двойная, а при значении collapse рамка становится одинарной.

**Основные теги для верстки**

При верстке сайта с помощью слоев самым часто используемым HTML-тегом является <div>, который как раз и формирует слой на веб-странице. Он является блочным тегом. Второй тег, который используется при верстке, – это строчный тег <span>. Сами по себе эти теги ничего на экране не отображают, и оформляются они стилями CSS.

|  |
| --- |
| <div>Блочный элемент</div> <span>Строчный элемент</span> |

**Блочный элемент** создает разрыв строки перед тегом и после него. Он образуют прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя, если для него не задано значение width.

Блочные элементы могут содержать внутри себя элементы любого типа. Нельзя размещать блочные элементы внутри строчных, за исключением элемента <img>. Для блочных элементов можно задавать margin и padding. Свойства width и height устанавливают ширину и высоту области содержимого элемента. Фактическая ширина элемента складывается из ширины полей (внутренних отступов), границ и внешних отступов.

**Строчные элементы** не создают блоки, они отображаются на одной строке с содержимым рядом стоящих тегов. Строчные элементы являются потомками блочных элементов. Они игнорируют верхние и нижние margin и padding, но, если для элемента задан фон, он будет распространяться на верхний и нижний padding, заходя на соседние строки текста.

Ширина и высота строчного элемента зависит только от его содержания, задать его размеры с помощью CSS нельзя. Можно увеличить расстояние между соседними элементами по горизонтали с помощью горизонтальных полей и отступов.

Давайте вспомним, какие основные элементы HTML относятся к блочным, а какие к строчным.

Блочные элементы

|  |  |  |
| --- | --- | --- |
| * <h1>...</h1> * <h2>...</h2> * <h3>...</h3> * <h4>...</h4> * <h5>...</h5> * <h6>...</h6> * <li>…</li> * <legend>…</legend> * <nav>…</nav> * <ol>…</ol> | * <article>…</article> * <aside>…</aside> * <body>…</body> * <blockquote>…</blockquote> * <div>…</div> * <fieldset>…</fieldset> * <footer>…</footer> * <form>...</form> * <header>…</header> * ... | * <output>…</output> * <p>…</p> * <pre>…</pre> * <section>…</section> * <ul>…</ul> |

Строчные элементы

|  |  |  |
| --- | --- | --- |
| * <a href=”#”>...</a> * <i>...</i> * <b>...</b> * <em>...</em> * <strong>...</strong> * <font>...</font> * <img> * <input> * ... |  |  |

В этом списке тег картинки <img> является замещаемым строчным элементом, т.е. при помощи замещаемых элементов указывается, что в данном месте должен быть какой-то сторонний объект, в данном случае картинка. Замещаемому элементу можно задать ширину и высоту, но он все равно будет являться строчным.

**Практика**

Создание стилей для меню сайта

|  |  |
| --- | --- |
| <ul class="menu">     <li>Главная</li>     <li><a href="catalog.html">Каталог</a></li>     <li><a href="contacts.html">Контакты</a></li> </ul> <h3>Каталог</h3> <ul class="menu">     <li>Товар 1</li>     <li class="new">Товар 2</li>     <li>Товар 3</li> </ul> | .menu li a:hover {     color: #1896FF; } .menu li a:active {     color: red; } .menu .new:after {     content: " new";     color: blue; } |

**Практическое задание**

Добавление контента на нашей странице, создание первых блоков и эффектов на ссылки сайта.

1. При наведении на меню сайта цвет ссылок должен меняться на произвольный.
2. На всех страницах создать блок с классом header, содержащий:
   1. Логотип вашего интернет магазина.
   2. Меню сайта.
3. Создать блок с классом footer и разместить в него содержимое футера.
4. При вводе текста в поле «Email» на странице контактной информации поменять цвет текста и цвет поля.
5. При вводе текста в текстовые поля каждое слово должно начинаться с большой буквы, даже если пользователь будет вводить с маленькой буквы.
6. На странице товара каталога:
   1. Под заголовком «Технические характеристики» разместить таблицу размером не меньше 2\*3 (есть задание \*).
   2. Записать в таблицу произвольные технические характеристики вашего товара.
   3. Если технических характеристик нет, добавьте таблицу с любыми параметрами по вашему выбору.
7. \* При наведении на изображения в разделе просмотра товара, подробного описания необходимо сделать так, чтобы рамка исчезала, но сама картинка при этом не должна смещаться, т.е должна оставаться неподвижной.
8. \* На странице просмотра товара каталога в каждом параграфе сделать первую букву первого слова другим шрифтом и размером.
9. \* Создать таблицу размером не менее 4\*5.
10. \* Объединить несколько ячеек как по горизонтали, так и по вертикали.
11. \* Задать высоту для верхней строки таблицы, отличную от всех остальных.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

**Дополнительные материалы**

[Популярно о псевдоэлементах :Before и :After](https://habrahabr.ru/post/154319/).

[Псевдоклассы и псевдоэлементы](https://ru.wikibooks.org/wiki/CSS/%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE-%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_%D0%B8_%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE-%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D1%8B).

[Дополнительный материал о таблицах](http://html5book.ru/html-table/).

[div & span](http://technologyweb.org/div-%D0%B8-span/).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/samcss/psevdoelementy>.
2. <http://htmlbook.ru/samcss/psevdoklassy>.
3. <http://html5book.ru/osnovy-css/>.
4. <http://html5book.ru/psevdoklassy/>.
5. <http://html5book.ru/html-table/>.
6. <http://htmlbook.ru/css/border-collapse>.
7. <http://technologyweb.org/div-%D0%B8-span/>.

Формирование блочной модели

Свойство display. Обтекаемые элементы. Позиционирование блоков.

[Свойство display](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.c713xnuw3xem)

[Значения свойства display](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.4rzceh7izf7j)

[Схлопывания](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.xtt1gta7pthg)

[Формирование блочной модели](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.ytqvmv91w6f)

[Обтекаемые элементы](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.c0c8fg8cgey1)

[Позиционирование блоков](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.zcci5jphygix)

[Абсолютное позиционирование](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.l1awt1b5noos)

[Относительное позиционирование](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.gv7rsfpjtezo)

[Фиксированное положение](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.xcivrctf30gc)

[Значение по умолчанию](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.v8b169j0qxaa)

[Совмещенное значение](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.57hrlof347ji)

[Z-index](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.qj3h4xfn7yjt)

[Практика](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.vg3uy1csazbk)

[Создание основной структуры сайта](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.69nq3b6y1irv)

[Практическое задание](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.pyvx3fvejl8j)

[Дополнительные материалы](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1DO9O1oLgxCWt8-PacbPDdWxfKw7EGx5TxGBFDq-XbyM/edit#heading=h.6rhrclqn1f50)

**Свойство display**

При помощи CSS можно изменить тип элемента, т.е. блочный тег можно сделать строчным, а строчный – блочным. Для этого существует CSS-свойство display. Вернемся к предыдущему примеру и для элементов <div> зададим значение свойства display: inline;, а для <span> – значение display: block;

|  |
| --- |
| div, span {         border: 1px solid #000;         width: 400px;         height: 50px; }  div {         display: inline; }  span {         display: block; } |

В этом случае получается, что элементы поменялись местами, <div> стал строчным элементом, и ему теперь невозможно задать ни ширину, ни высоту, а <span> стал блочным, и ему теперь можно задать ширину и высоту.

Значения свойства display

* none (скрыть).
* block (блочный).
* inline (строчный).
* inline-block (строчно-блочный).
* table-cell (ячейка таблицы).
* flex (гибкий).

**Блочный элемент** (display: block;) создает разрыв строки перед тегом и после него. Он образует прямоугольную область, по ширине занимающую всю ширину веб-страницы или блока-родителя, если для него не задано значение width.

Блочные элементы могут содержать внутри себя элементы любого типа. Нельзя размещать блочные элементы внутри строчных, за исключением элемента <img>. Для блочных элементов можно задавать margin и padding.

Свойства width и height устанавливают ширину и высоту области содержимого элемента. Фактическая ширина элемента складывается из ширины полей (внутренних отступов), границ и внешних отступов.

**Строчные элементы** (display: inline;) не создают блоки, они отображаются на одной строке с содержимым рядом стоящих тегов. Строчные элементы являются потомками блочных элементов. Они игнорируют верхние и нижние margin и padding, но если для элемента задан фон, он будет распространяться на верхний и нижний padding, заходя на соседние строки текста.

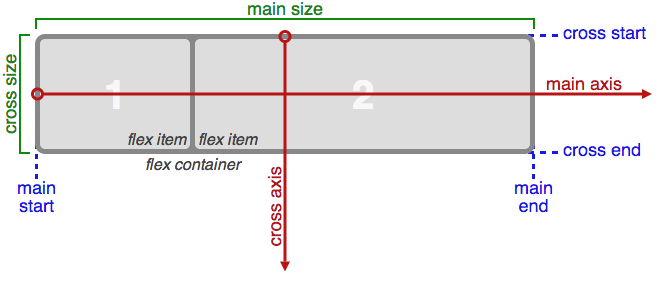
Ширина и высота строчного элемента зависит только от его содержания, задать размеры с помощью CSS нельзя. Можно увеличить расстояние между соседними элементами по горизонтали с помощью горизонтальных полей и отступов.

Существует еще одна группа элементов, которые браузер обрабатывает как **строчно-блочные** (display: inline-block;). Такие элементы являются встроенными, но для них можно задавать поля, отступы, ширину и высоту.

Современные браузеры (IE8+) позволяют описывать таблицу любыми элементами, если поставить им соответствующие значения display. Это хорошо для семантической верстки и позволяет избавиться от лишних тегов.

Для блока возможно задать значение ячейки таблицы (table-cell). Внутри ячеек свойство vertical-align выравнивает содержимое по вертикали. CSS не требует, чтобы вокруг table-cell была структура таблицы: table-row и т.п. Может быть просто одинокий DIV, это допустимо. При этом он ведёт себя как ячейка TD, то есть подстраивается под размер содержимого и умеет вертикально центрировать его при помощи vertical-align.

flexbox – это целый модуль, а не просто единичное свойство, он объединяет в себе множество свойств. Некоторые из них должны применяться к контейнеру (родительскому элементу, так называемому flex-контейнеру), в то время как другие применяются к дочерним элементам или flex-элементам.



Если обычный layout основывается на направлениях потоков блочных и inline-элементов, то flex-layout основывается на «направлениях flex-потока». Ознакомьтесь со схемой из спецификации, разъясняющей основную идею flex-layout-ов.

В основном элементы будут распределяться либо вдоль главной оси (от main-start до main-end), либо вдоль поперечной оси (от cross-start до cross-end).

* main axis – главная ось, вдоль которой располагаются flex-элементы. Обратите внимание: она не обязательно должна быть горизонтальной, все зависит от свойства justify-content (см. ниже).
* main-start | main-end – flex-элементы размещаются в контейнере от позиции main-start до позиции main-end.
* main size – ширина или высота flex-элемента в зависимости от выбранной основной величины. Основная величина может быть шириной либо высотой элемента.
* cross axis – поперечная ось, перпендикулярная главной. Ее направление зависит от направления главной оси.
* cross-start | cross-end – flex-строки, заполняются элементами и размещаются в контейнере от позиции cross-start и до позиции cross-end.
* cross size – ширина или высота flex-элемента в зависимости от выбранной размерности. Это свойство совпадает с width или height элемента в зависимости от выбранной размерности.

Элементы в контейнере поддаются выравниванию при помощи свойства justify-content вдоль главной оси. Это свойство принимает целых пять разных вариантов значений:

* flex-start (default): гибкие элементы выравниваются по началу главной оси.
* flex-end: элементы выравниваются по концу главной оси.
* center: элементы выравниваются по центру главной оси.
* space-between: элементы занимают всю доступную ширину в контейнере, крайние элементы вплотную прижимаются к краям контейнера, а свободное пространство равномерно распределяется между элементами.
* space-around: гибкие элементы выравниваются таким образом, что свободное пространство равномерно распределяется между элементами. Но стоит отметить, что пространство между краем контейнера и крайними элементами будет в два раза меньше, чем между элементами в середине ряда.

Мы также имеем возможность выравнивания элементов по cross-оси. Применив свойство align-items, которое принимает пять разных значений, можно добиться интересного поведения. Это свойство позволяет выравнивать элементы в строке относительно друг друга.

* flex-start: все элементы прижимаются к началу строки.
* flex-end: элементы прижимаются к концу строки.
* center: элементы выравниваются по центру строки.
* baseline: элементы выравниваются по базовой линии текста.
* stretch (default): элементы растягиваются, полностью заполняя строку.
* Еще одно свойство, похожее на предыдущее, – align-content. Оно отвечает за выравнивание целых строк относительно гибкого контейнера. Оно не будет давать эффекта, если гибкие элементы занимают одну строку.

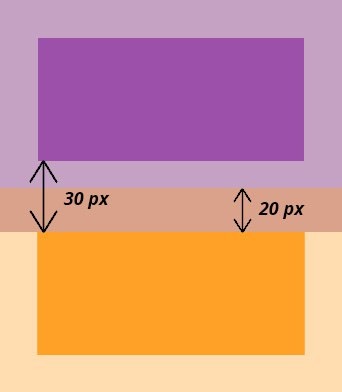
Свойство принимает шесть разных значений:

* flex-start: все линии прижимаются к началу cross-оси.
* flex-end: все линии прижимаются к концу cross-оси.
* center: Flex-элементы выравниваются по центру flex-контейнера.
* space-between: линии распределяются от верхнего края до нижнего, оставляя свободное пространство между строками, крайние же строки прижимаются к краям контейнера.
* space-around: линии равномерно распределяются по контейнеру.
* stretch (default): линии растягиваются, занимая все доступное пространство.

Одно из основных свойств – flex-basis. С его помощью мы можем указывать базовую ширину гибкого элемента. По умолчанию имеет значение auto. Это свойство тесно связано с flex-grow и flex-shrink, о которых будет рассказано чуть позже. Принимает значение ширины в px, %, em и остальных единицах. По сути, это не строго ширина гибкого элемента, а своего рода отправная точка, относительно которой происходит растягивание или усадка элемента. В режиме auto элемент получает базовую ширину относительно контента внутри него.

Схлопывания

Когда два или более вертикальных margin соприкасаются, они сливаются, при этом ширина общего отступа равна ширине большего из исходных отступов.



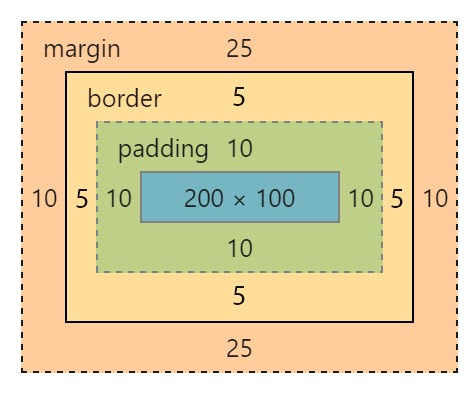
Слияние выполняется только для блочных элементов в нормальном потоке документа. Внешние вертикальные отступы строчных, плавающих и абсолютно позиционированных элементов не сливаются. Чтобы получить желаемый промежуток, можно задать, например, для верхнего элемента padding-bottom, а для нижнего элемента – margin-top.

Существуют исключения для схлопывания:

* С блоками, которым присвоено float.
* С основными элементами (html, body).
* Для блоков, которым присвоено свойство и значение position:absolute.
* Для строчных элементов.

**Формирование блочной модели**

На первый взгляд может показаться, что width – это окончательная ширина элемента, height – его окончательная высота. На самом деле это не так, width и height – не окончательные размеры элемента. Для того чтобы вычислить размеры, необходимо учитывать следующие моменты.



Если внимательно ознакомиться с данной схемой, можно сделать вывод, что ширина блока складывается из следующих свойств:

|  |
| --- |
| margin-left + border-left + padding-left + width + padding-right + border-right + margin-right |

Соответственно, высота из следующих:

|  |
| --- |
| margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom |

Внутренний отступ или поле элемента (padding) добавляет отступы внутри элемента, между его основным содержимым и его границей. Если для элемента задать фон, он распространится также и на поля элемента. Внутренний отступ не может принимать отрицательных значений, в отличие от внешнего отступа.

Внешний отступ (margin) добавляет отступы за границами элемента, создавая тем самым промежутки между элементами. Они всегда остаются прозрачными, и через них виден фон родительского элемента. Значения padding и margin задаются в следующем порядке: верхнее, правое, нижнее и левое.

Граница или рамка элемента задается с помощью свойства border. Если цвет рамки не задан, она принимает цвет основного содержимого элемента, например текста. Если рамка имеет разрывы, то сквозь них будет проступать фон элемента.

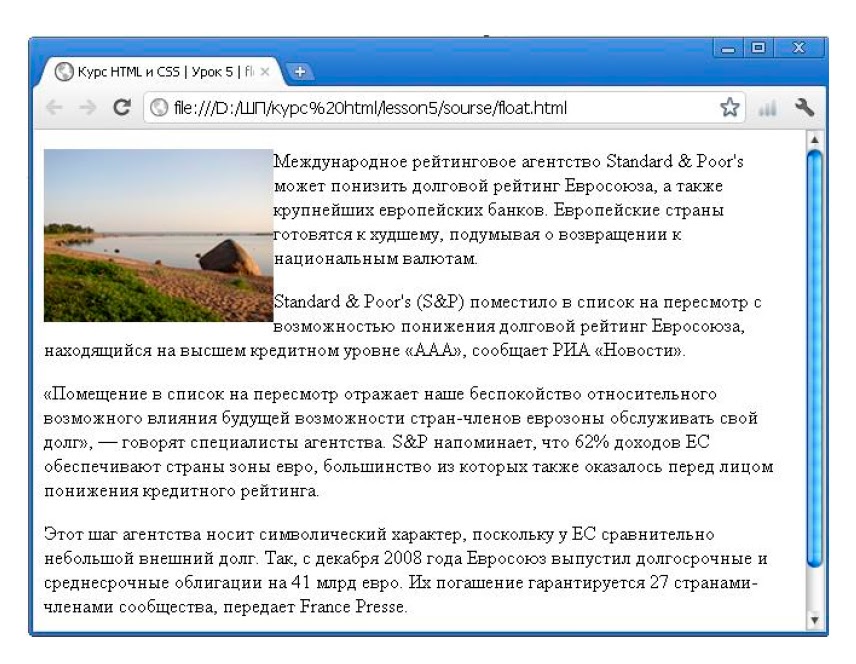
Внешние, внутренние отступы и рамка элемента не являются обязательными, по умолчанию их значение равно нулю. Тем не менее, некоторые браузеры добавляют к этим свойствам положительные значения по умолчанию на основе своих таблиц стилей.

Обтекаемые элементы

Обтекаемые, или «плавающие» элементы используются для реализации обтекания текстом изображений, создания врезок и даже создания многостолбцовых компоновок.

Также обтекаемые элементы активно используются при верстке веб-страниц, с их помощью можно заменить табличную верстку на верстку слоями. Для того чтобы задать обтекание, в CSS существует только одно свойство float, которое может принимать всего два значения – left и right.

В следующем примере картинке, т.е. тегу <img />, задано свойство float: left;

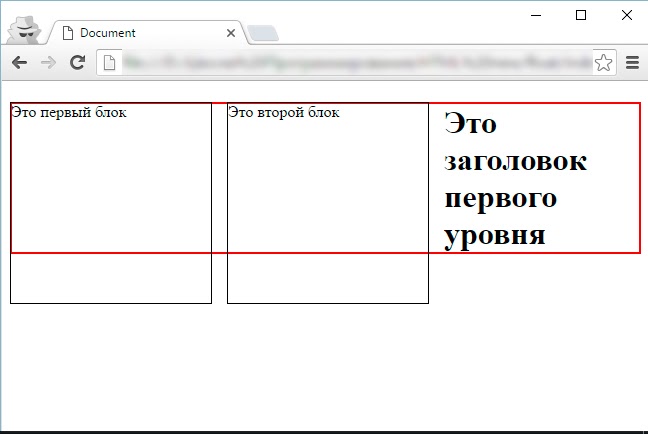


В этом случае картинка займет положение слева и позволит любым элементам, будь они строчные или блочные, обтекать себя справа.

Далее рассмотрим следующий пример. Создадим два элемента <div> и один заголовок первого уровня <h1>.

|  |  |
| --- | --- |
| **HTML** | **CSS** |
| <div>Это первый блок</div> <div>Это второй блок</div> <h1>Это заголовок первого уровня</h1> | div {        width: 200px;        height: 200px;        margin-right: 15px;        border: 1px solid #000;        float: left; } h1 {        border: 1px solid #f00; } |

У обоих элементов <div> задано свойство float: left;, т.е. они должны занимать левое положение и позволять обтекать себя справа. Посмотрим на работу этого примера в браузере.



Разберемся, что произошло. Элементы <div> находятся на одной линии по горизонтали, что ожидаемо, т.к. у них задано свойство float: left;. Первый <div> занял положение слева, позволил обтекать себя справа. Второй <div>, соответственно, в свою очередь также позволил обтекать себя справа. Заголовок первого уровня находится справа второго элемента <div>, но его рамка обрамляет также оба элемента <div>. Это происходит потому, что у свойства float есть особенность: элементы, которым задано это свойство, начинают притягивать к себе все близлежащие элементы и заставляют их тоже участвовать в обтекании. Но с этим можно бороться.

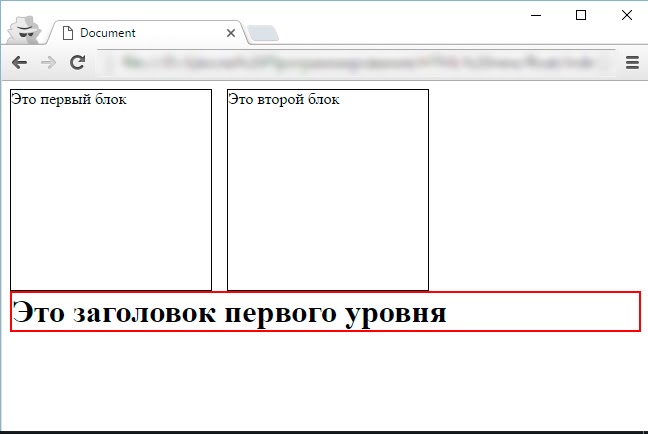
Рассмотрим две ситуации.

1. Заголовок не должен участвовать в обтекании и должен находиться под элементами <div>.

В этом случае необходимо применить запрет на обтекание. Для этого в CSS существует свойство clear. Оно может принимать три значения: left, отменяющее обтекание с левого края, right – с правого края и both, которое отменяет обтекание с обеих сторон. Добавим свойство clear со значением both для заголовка первого уровня.

|  |  |
| --- | --- |
| **HTML** | **CSS** |
| <div>Это первый блок</div> <div>Это второй блок</div> <h1>Это заголовок первого уровня</h1> | div {         width: 200px;         height: 200px;         margin-right: 15px;         border: 1px solid #000;         float: left; } h1 {         border: 1px solid #f00;         clear: both; } |

Если запустить данный код в браузере, заголовок уже не будет участвовать в обтекании, а будет находиться под элементами <div>.

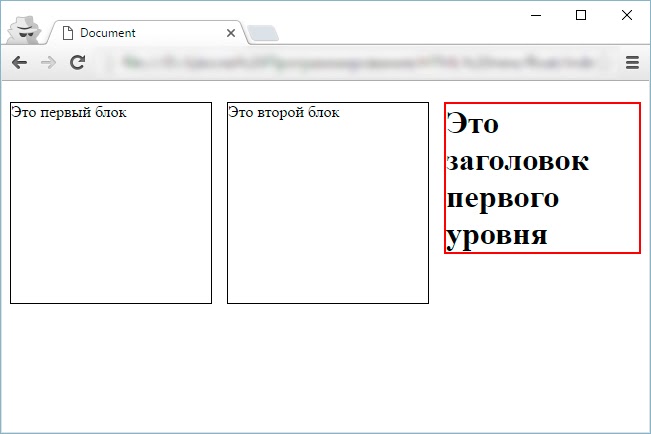


2. Заголовок остается на том же месте, где он сейчас находится, но рамка должна обрамлять только сам заголовок.

Для решения этой задачи поможет CSS-свойство overflow. Оно определяет, как будет вести себя блочный элемент в случае его переполнения, и при значении hidden отображает только содержимое этого элемента.

|  |  |
| --- | --- |
| **HTML** | **CSS** |
| <div>Это первый блок</div> <div>Это второй блок</div> <h1>Это заголовок первого уровня</h1> | div {        width: 200px;        height: 200px;        margin-right: 15px;        border: 1px solid #000;        float: left; } h1 {        border: 1px solid #f00;        overflow: hidden; } |

Если запустить этот пример в браузере, заголовок остается на том же месте, и рамка теперь обрамляет только элемент <h1>.



**Позиционирование блоков**

Идея, лежащая в основе позиционирования, довольно проста. Оно позволяет точно определить, где появятся блоки относительно другого элемента или окна браузера. По умолчанию все элементы располагаются последовательно один за другим в том порядке, в котором они определены в HTML-документе ({position: static}). Свойство не наследуется.

Блочный элемент (p, div, h1 и др.) занимает 100% ширины родительского элемента (по умолчанию – body). Поэтому блочные элементы отображаются один под другим в соответствии с разметкой страницы.

Строчный элемент (em, strong, span и др.) занимает ширину, которая соответствует ширине содержимого внутри него. Поэтому строчные элементы отображаются рядом друг с другом.

Свойство position вместе со значениями top, right, bottom и left отображает элемент с нарушением обычного порядка, смещая его на заданное расстояние. При позиционировании элементов можно использовать как положительные, так и отрицательные значения. Таким образом, существуют 4 вида позиционирования.

Абсолютное позиционирование

При абсолютном позиционировании элемент не существует в потоке документа, и его положение задается относительно краев браузера. Задать этот тип можно через значение absolute свойства position. Координаты указываются относительно краев окна браузера, называемого «видимой областью».

Для режима характерны следующие особенности:

* Ширина слоя, если она не задана явно, равна ширине контента плюс значения полей, границ и отступов.
* Слой не меняет свое исходное положение, если у него нет свойств right, left, top и bottom.
* Свойства left и top имеют более высокий приоритет по сравнению с right и bottom. Если left и right противоречат друг другу, значение right игнорируется. То же самое касается и bottom.
* Если left задать отрицательное значение, слой уйдет за левый край браузера, полосы прокрутки при этом не возникнет. Это один из способов спрятать элемент от просмотра. То же относится и к свойству top, только слой уйдет за верхний край.
* Если left задать значение больше ширины видимой области или указать right с отрицательным значением, появится горизонтальная полоса прокрутки. Подобное правило работает и с top, только речь пойдет о вертикальной полосе прокрутки.
* Одновременно указанные свойства left и right формируют ширину слоя, но только если width не указано. Стоит добавить свойство width, и значение right будет проигнорировано. Аналогично произойдет и с высотой слоя, только участвуют свойства top, bottom и height.
* Элемент с абсолютным позиционированием перемещается вместе с документом при его прокрутке.

Относительное позиционирование

relative (относительное позиционирование) – элемент будет смещаться относительно его определенного в настоящее время положения, и при этом его место будет оставаться незаполненным. Добавление свойств left, top, right и bottom изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения. Положительное значение left определяет сдвиг вправо от левой границы элемента, отрицательное – сдвиг влево. Положительное значение top задает сдвиг элемента вниз, отрицательное – сдвиг вверх.

Свойства bottom и right производят обратный эффект. При положительном значении right сдвигает элемент влево от его правого края, при отрицательном – вправо. При положительном значении bottom элемент поднимается вверх, при отрицательном опускается вниз.

Для относительного позиционирования характерны следующие особенности:

* Этот тип позиционирования неприменим к элементам таблицы вроде ячеек, строк, колонок     и др.
* При смещении элемента относительно исходного положения место, которое занимал элемент, остается пустым и не заполняется ниже- или вышележащими элементами.

Фиксированное положение

Фиксированное положение слоя задается значением fixed свойства position и по своему действию похоже на абсолютное позиционирование. Но, в отличие от него, привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы. Еще одна разница с absolute заключается в том, что при выходе фиксированного слоя за пределы видимой области справа или снизу от нее не возникает полос прокрутки.

Применяется такой тип позиционирования для создания меню, вкладок, заголовков – в общем, любых элементов, которые должны быть закреплены на странице и всегда видны посетителю.

Значение по умолчанию

Если для элемента свойство position не задано или его значение static, элемент выводится в потоке документа как обычно. Иными словами, элементы отображаются на странице в том порядке, как они идут в исходном коде HTML.

Свойства left, top, right, bottom, если определены, игнорируются.

Совмещенное значение

position:relative + position:absolute

Назначив родительскому блоку относительное позиционирование (position: relative), мы сможем позиционировать любые дочерние элементы относительно его границ. Если у элемента есть позиционированный предок, то position: absolute работает относительно него, а не относительно документа. Нужно пользоваться таким позиционированием с осторожностью, т.к. оно может перекрыть текст. Этим оно отличается от float.

Z-index

Любые позиционированные элементы на веб-странице могут накладываться друг на друга в определенном порядке, имитируя тем самым третье измерение, перпендикулярное экрану. Каждый элемент может находиться как ниже, так и выше других объектов веб-страницы. Их размещением по z-оси и управляет z-index. Это свойство работает только для элементов, у которых значение position задано как absolute, fixed или relative.

В качестве значения используются целые числа (положительные, отрицательные и ноль). Чем больше значение, тем выше находится элемент по сравнению с теми элементами, у которых оно меньше. При равном значении z-index на переднем плане находится тот элемент, который в коде HTML описан ниже. Хотя спецификация разрешает использовать отрицательные значения z-index, но такие элементы не отображаются в браузере Firefox до версии 2.0 включительно.

Кроме числовых значений применяется auto – порядок элементов в этом случае строится автоматически, 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.

**Практика**

Создание основной структуры сайта

|  |  |
| --- | --- |
| <div class="container">     <div class="header"></div>     <div class="content"></div>     <div class="footer"></div> </div> | .container {     width: 800px;     margin: 0 auto; } .header {     background-color: #2118FF;     height: 100px; } .content {     background-color: #E1E0E1;     height: 400px; } .footer {     background-color: #905BAE;     height: 100px; } |

**Практическое задание**

Создание блочной структуры сайта, позиционирование элементов.

1. Главная страница:
   1. Создать родительский блок с классом container, задать значение ширины и расположить блок по центру экрана.
   2. Задать блоку с классом header значение  высоты.
   3. Отодвинуть логотип от левого края.
   4. Из вертикального меню сделать горизонтальное.
   5. Разместить меню справа от логотипа.
   6. Прижать меню к правому краю.
2. Footer (на всех страницах):
   1. Задать высоту.
   2. Поменять цвет фона.
   3. Прижать текст «Все права защищены» к правому краю и отодвинуть от верхней и правой границы.
3. Страница просмотра товара каталога:
   1. Разместить краткое описание товара справа от картинки.
   2. Добавить кнопку «Купить».
   3. Присвоить наведение и нажатие на кнопку «Купить».
4. Страница каталога:
   1. Создать блоки для картинки и ссылки на просмотр товара каталога.
   2. Расположить элементы каталога горизонтально.
5. Доделать то, что не успели в прошлых уроках.
6. \* Расположить элементы на ваше усмотрение.
7. \* Страница контактов:
   1. Ширина карты 100%.
   2. Высота карты 374px.
8. \* Добавить фоновые изображения для header на каждой странице.

На данном этапе уже можно приступить к расположению элементов в соответствии с макетом.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

**Дополнительные материалы**

[Свойство display](https://learn.javascript.ru/display).

[Статья про display](http://technologyweb.org/%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE-display/).

[Формирование блочной модели](https://ru.wikibooks.org/wiki/CSS/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C).

[Позиционирование за 10 шагов](http://dreamhelg.ru/2011/02/css-position-in-10-steps/).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/css/display>.
2. <https://learn.javascript.ru/display>.
3. <http://htmlbook.ru/css/z-index>.
4. <http://www.pvsm.ru/css3/73795>.
5. <http://html5book.ru/css3-flexbox/>.
6. <http://htmlbook.ru/samlayout/blochnaya-verstka/blochnaya-model>.
7. <https://ru.wikibooks.org/wiki/CSS/%D0%91%D0%BB%D0%BE%D1%87%D0%BD%D0%B0%D1%8F_%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C>.
8. <http://htmlbook.ru/samlayout/blochnaya-verstka/pozitsionirovanie-elementov>.

Работа с макетом дизайна в формате PSD

Photoshop и его основные функции.

[Рабочая среда «Начало работы»](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.cm7rewg0t2vk)

[Отключение рабочей среды «Начало работы»](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.7pjsui1hs5jv)

[Обзор рабочей области Photoshop](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.rf28znro29i3)

[Описание рабочей среды](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.vxxet46yvp1m)

[Функции, повышающие удобство использования](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.knl3al8xlp6n)

[Скрытие или отображение всех панелей](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.80c5vmsk78zy)

[Параметры отображения палитр/панелей](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.9wy314kdoca8)

[Перенастройка панели «Инструменты»](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.np8bqr3mg3tr)

[Управление окнами и палитрами/панелями](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.6ktm6ntpj544)

[Упорядочение, стыковка и свободное перемещение окон документа](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.v9p5q7xyl6j5)

[Прикрепление и открепление палитр/панелей](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.tw0s5c3kj7ta)

[Добавление и удаление палитр/панелей](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.7ijc0gryvw1y)

[Изменение размеров палитр/панелей](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.kvjyku6woly2)

[Свертывание и развертывание значков панели](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.gal4zreaq7bo)

[Сохранение и переключение рабочих пространств/сред](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.1908w3uti9h7)

[Сохранение пользовательского рабочего пространства/среды](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.tne0ewex6p6v)

[Удаление пользовательской рабочей среды](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.26t0orbcgb0t)

[Как работать в Photoshop](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.l6d3jbc6xph6)

[Создание нового файла](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.99ojjnmelri9)

[Ориентация по документу](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.bxpn2fi0elde)

[Выбор слоев](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.vtbaoqjfltfn)

[Показать и скрыть](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.88yldc6c2mwn)

[Информация о слоях](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.6kulo4t77j10)

[Экспорт графики для WEB](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.9g8qgusck7ao)

[Горячие клавиши](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.eovocofzwn6u)

[Практическое задание](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.2cp3mwkuh9dt)

[Дополнительные материалы](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1HH-oeA-e1q7cM1nAXJpSWTptyZtv9UMRP4oowhRxwfg/edit#heading=h.vcznvglbycoe)

Рабочая среда «Начало работы»

Рабочая среда Photoshop «Начало работы» позволяет быстро переходить к последним документам, библиотекам и наборам. В этой рабочей среде может отображаться содержимое, соответствующее вашим требованиям (зависит от состояния подписки). Еще одно преимущество среды «Начало работы» – поддержка поиска необходимого ресурса Adobe Stock непосредственно во время работы над проектом.

Рабочая среда «Начало работы» отображается при запуске Photoshop или отсутствии открытых документов.

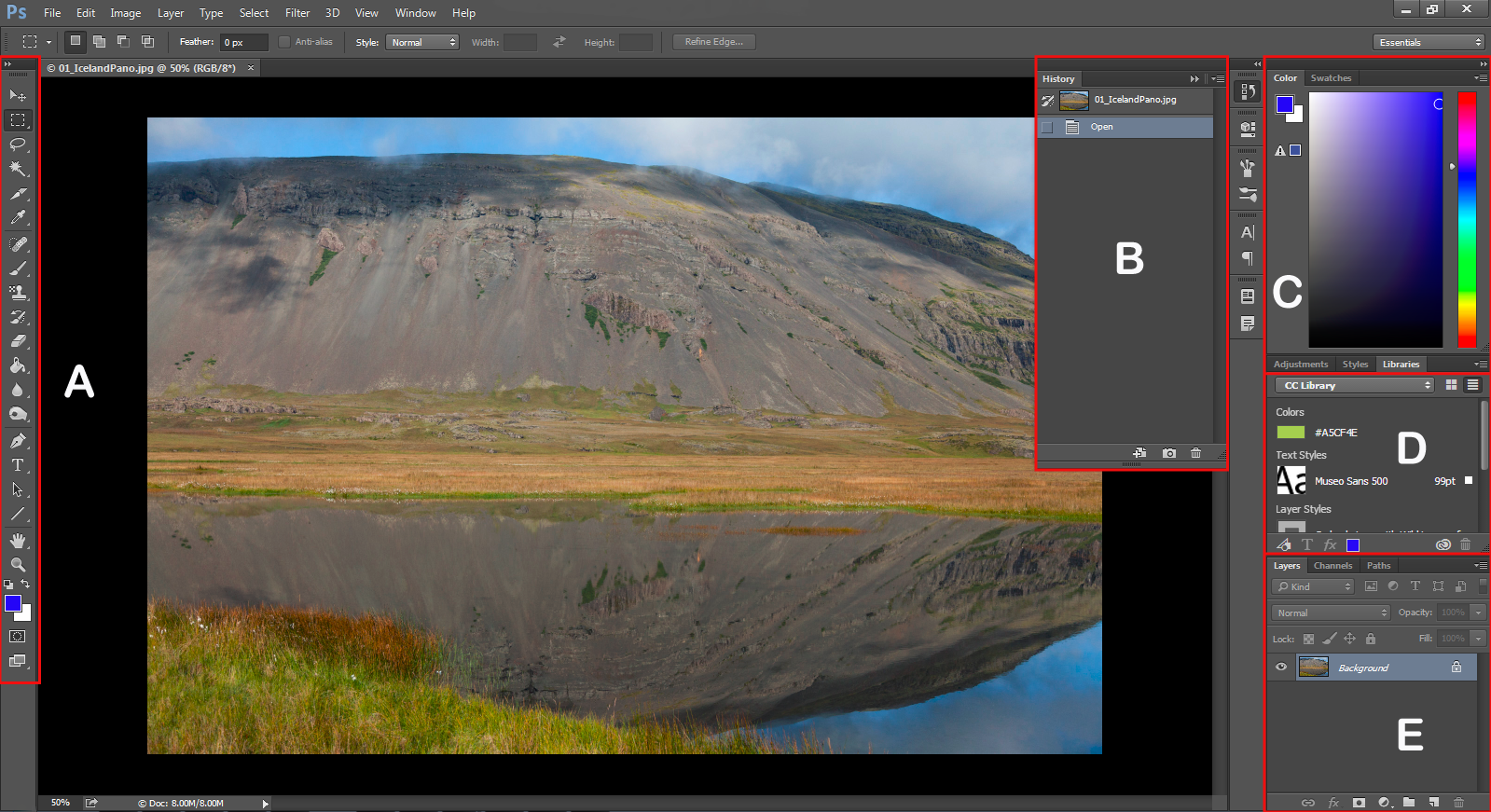
Примечание. При необходимости количество отображаемых последних документов можно изменить. Выберите «Установки» > «Обработка файлов» и укажите требуемое значение (0–100) в поле Длина списка недавних файлов.

Для выхода из рабочей среды «Начало работы» нужно просто нажать клавишу Esc.

Отключение рабочей среды «Начало работы»

1. Выберите «Установки» > «Основные».
2. Снимите флажок «Показывать рабочую среду «Начало работы», если нет открытых документов».

Обзор рабочей области Photoshop



A: Панель «Инструменты».

B: Панель «История».

C: Панель «Цвет».

D: Панель «Библиотеки Creative Cloud».

E: Панель «Слои».

Описание рабочей среды

* Находящаяся в верхней части окна панель приложения содержит переключатель рабочих пространств/сред, меню (только Windows) и другие элементы управления приложением. При работе в некоторых программах Adobe на платформе Mac пользователь может отобразить или скрыть эту панель, используя меню «Окно».
* Панель «Инструменты» содержит инструменты для создания и редактирования изображений, графических объектов, элементов страниц и т.д. Связанные инструменты располагаются в группах.
* На панели «Управление» в строке «Параметры» выводятся настройки выбранного в данный момент инструмента.
* Окно документа отображает находящийся в работе файл. В окна документа можно поместить закладки, в некоторых случаях их можно группировать и закреплять.
* Палитры упрощают отслеживание и изменение обрабатываемого объекта. Палитры можно группировать, собирать в подборки или закреплять.
* Фрейм приложения группирует все элементы рабочей среды в единое интегрированное окно, которое позволяет работать с приложением как с единым целым. При перемещении или изменении размера фрейма приложения или его элементов все элементы внутри него взаимодействуют друг с другом и не перекрываются. Палитры не исчезают при переключении приложений или при случайном щелчке за пределами приложения. При работе с несколькими приложениями можно расположить их рядом на экране или на нескольких мониторах.

Если вы предпочитаете традиционный произвольный интерфейс Mac, можно отключить фрейм приложения.

Функции, повышающие удобство использования

Рабочая среда Photoshop проста в использовании и содержит ряд функций, которые обеспечивают удобство использования:

* Различные уровни яркости: щелкните «Редактирование» > «Установки» (Windows) или Photoshop > «Установки» (Mac OS) и выберите образец «Цветовая тема» в разделе «Интерфейс».
* Чтобы быстро уменьшить яркость, нажмите Shift + 1; чтобы увеличить яркость, нажмите Shift + 2. В Mac OS также необходимо нажать клавишу FN.
* Представления в изображении: получайте актуальные сведения по мере использования избранных инструментов. Представления в изображении показывают размеры выделенного фрагмента, углы трансформирования и т.д. Чтобы изменить размещение представлений, выберите параметр в разделе «Показывать значения трансформирования» в установках интерфейса.
* Максимально доступное пространство экрана: нажмите кнопку в нижней части панели инструментов, чтобы переключиться между стандартным и полноэкранным режимами отображения.

Скрытие или отображение всех панелей

* Чтобы скрыть или отобразить все палитры и панели, включая панели «Инструменты» и «Управление», нажмите клавишу Tab.
* Чтобы скрыть или отобразить все палитры и панели, кроме панелей «Инструменты» и «Управление», нажмите клавиши Shift+Tab.

Можно временно отобразить скрытые панели, если в установках интерфейса выбран параметр «Автоматически показывать скрытые панели». Переместите курсор к краю окна приложения (Windows) или к краю монитора (Mac OS) и наведите его на появившуюся полосу.

Параметры отображения палитр/панелей

Нажмите значок меню палитры/панели в ее правом верхнем углу.

Меню палитры/панели можно открыть, даже если она свернута.

В Photoshop можно изменить размер шрифта для текста панелей и всплывающих подсказок. В настройках интерфейса выберите размер в меню «Размер шрифта меню».

Перенастройка панели «Инструменты»

Инструменты панели «Инструменты» могут располагаться в один или два столбца.

Щелкните двойную стрелку, расположенную в верхней части панели «Инструменты».

Управление окнами и палитрами/панелями

Пользовательское рабочее пространство/среду можно создать, перемещая и изменяя окна документов и палитр/панелей. Можно сохранить несколько рабочих сред и переключаться между ними.

Упорядочение, стыковка и свободное перемещение окон документа

Если открыто более одного файла, окна документов снабжаются закладками.

Для изменения порядка окон документов перетащите закладку окна в новое место в группе.

Для открепления (свободного перемещения или отстыковки) окна документа из группы окон перетащите закладку окна из группы.

**Примечание**. Можно также выбрать меню «Окно» > «Упорядочить» > «Свободно перемещать содержимое окна» для перемещения одного окна документа либо «Окно» > «Упорядочить» > «Свободно перемещать все содержимое окон» для перемещения всех окон документа сразу.

Для прикрепления окна документа к другой группе окон перетащите закладку окна в эту группу.

Чтобы создать группы документов, расположенные каскадно или мозаично, перетащите окно к одной из зон перетаскивания, расположенных вдоль верхнего, нижнего или боковых краев другого окна. Также можно выбрать макет для группы с помощью кнопки «Макет» на панели приложения.

Чтобы переключиться на другой документ в группе со вкладками при перетаскивании выделенной области, перетащите выделенную область на вкладку документа и ненадолго задержите на ней.

Прикрепление и открепление палитр/панелей

Чтобы прикрепить палитру/панель, перетащите ее за вкладку в док и поместите сверху или снизу от других палитр или панелей либо между ними.

Чтобы прикрепить группу палитр/панелей, перетащите ее в док за строку заголовка (пустую одноцветную строку над вкладками).

Чтобы удалить палитру/панель или группу палитр/панелей из дока, перетащите ее из дока, удерживая за вкладку или строку заголовка. Можно перетащить палитру/панель в другой док или сделать плавающей.

Добавление и удаление палитр/панелей

Если из дока удалить все палитры/панели, он исчезнет. Можно создать док, перемещая палитры/панели в правый угол рабочего пространства/среды до появления зоны перетаскивания.

Для удаления палитры/панели щелкните правой кнопкой мыши (Windows) или левой кнопкой мыши с удерживанием клавиши Control (Mac) на ее закладке и выберите «Закрыть» или удалите ее из меню «Окно».

Чтобы добавить палитру/панель, выберите ее в меню «Окно» и закрепите в нужном месте.

Изменение размеров палитр/панелей

Чтобы свернуть или развернуть палитру/панель, группу палитр/панелей или ряд палитр/панелей, дважды щелкните вкладку. Также можно дважды щелкнуть область вкладки (пустое пространство рядом с вкладками).

Для изменения размера палитры/панели потяните за любую из ее сторон. Размеры некоторых палитр/панелей, например палитры «Цвет», нельзя изменить таким способом.

Свертывание и развертывание значков панели

Чтобы уменьшить загромождение рабочего пространства/среды, можно свернуть палитры/панели в значки. В некоторых случаях палитры/панели сворачиваются в значки в рабочем пространстве/среде по умолчанию.

Чтобы свернуть или развернуть все значки панели в столбце, щелкните двойную стрелку в его верхней части.

Чтобы развернуть значок одиночной палитры/панели, щелкните его.

Чтобы изменить размер значков палитры/панели так, чтобы видеть только значки (без подписей), изменяйте ширину дока до исчезновения текста. Чтобы восстановить текст, увеличьте ширину дока.

Чтобы свернуть развернутую палитру/панель в значок, щелкните ее вкладку, значок или двойную стрелку в строке заголовка.

Чтобы добавить плавающую палитру/панель или группу палитр/панелей в док значков, перетащите ее туда за вкладку или строку заголовка (палитры/панели автоматически сворачиваются в значки при добавлении их в док значков).

Чтобы переместить значок палитры/панели (или группу значков палитр/панелей), перетащите значок. Значки панелей можно перетаскивать вниз и вверх в пределах одного дока, в другие доки (где значки будут отображаться в стиле панелей этого дока) либо за пределы дока (где они станут плавающими значками).

Сохранение и переключение рабочих пространств/сред

Если текущий размер и положение панелей сохранены как именованное рабочее пространство/среда, это рабочее пространство/среду можно восстановить позднее, даже если некоторые палитры/панели были перемещены или закрыты. Названия сохраненных рабочих пространств/сред выводятся в переключателе рабочих пространств/сред на панели приложения.

Сохранение пользовательского рабочего пространства/среды

1. Настроив требуемую конфигурацию рабочей среды, выберите Окно > Рабочая среда > Новая рабочая среда.
2. Введите имя рабочей среды.
3. В меню «Захват» выберите один или несколько вариантов.

Удаление пользовательской рабочей среды

* Выберите «Управление рабочими средами» в переключателе рабочих сред на панели приложения, выберите рабочую среду и нажмите «Удалить».
* Выберите «Удалить рабочую среду» в переключателе рабочих сред.
* Выберите «Окно» > «Рабочая среда» > «Удалить рабочую среду», укажите среду, которую нужно удалить, и нажмите «Удалить».

**Как работать в Photoshop**

Создание нового файла

Происходит по нажатию клавиш Ctrl + N, после чего появляется окно с настройками нового документа.

Самые важные пункты для верстальщика: размер новой рабочей области и единицы измерения этих размеров. Ширину и высоту можно задавать вручную, а при копировании изображения из другого макета или вставки скриншота экрана размеры устанавливаются автоматически. В качестве единиц измерения стоит использовать пикселы.

Также размеры рабочей области и размеры изображения можно задать с помощью меню «Размер изображения» (Alt + Ctrl + I) и «Размер холста» (Alt + Ctrl + C).

Ориентация по документу

Tab – показать или скрыть все панели.

Зажать Пробел, «схватить» мышью за документ и перетаскивать – перемещение в рамках масштаба. Зажатие кнопки временно активирует инструмент Hand Tool.

Ctrl + 1 – установка масштаба 100 %.

Зажать H, зажать левую кнопку мыши – документ масштабируется так, чтоб был виден целиком, можно перетащить рамку.

Зажать Alt, крутить колесо мыши – масштабирование в обе стороны.

Выбор слоев

Можно поискать нужный слой в панели слоев, но есть способ быстрее.

Зажать Ctrl и кликнуть на слой. Зажатие кнопки временно активирует инструмент Move Tool. Работает при выборе любого инструмента, кроме Hand Tool (по зажатию Ctrl включается инструмент масштабирования) и самого Move Tool. Чтобы это работало, убедитесь, что настройки инструмента Move Tool (это панель под выпадающими меню, когда инструмент выбран) выставлены следующие: Auto‑Select – галка стоит, в выпадающем списке рядом – Layer.

Показать и скрыть

Показать или скрыть какие‑либо слои просто – кликнуть на иконке «глаз» этого слоя в панели слоев.

Alt + клик по иконке «глаз» в панели слоев – показать только один какой‑либо слой, прочее скрыть.

Информация о слоях

Двойной клик по миниатюре текстового слоя – редактирование слоя. Выставляйте текстовой курсор в нужное место – узнаете шрифт, размер, интерлиньяж, трансформации, кернинг, спейсинг и цвет.

Если у слоя справа есть курсивная надпись «fx» и иконка, открывающая список, – значит, у слоя есть эффекты. Кликайте на открывающую иконку – увидите список эффектов (можно отключить их показ – кликаем на иконки глаза рядом с эффектами), двойной клик по эффекту вызовет панель с настройками эффекта.

Показать и скрыть сетку, направляющие, габариты трансформации, нарезку – Ctrl + H.

Отмена и повтор последнего действия – Ctrl + Z.

Отмена действий последовательно – Ctrl + Alt + Z.

Команда из выпадающего списка Image → Trim – подрезка прозрачных или однотонных пикселей.

Узнать истинный габарит слоя с полупрозрачными эффектами – правый клик по названию слоя → Convert to Smart Object. После этого габарит трансформации показывает размер с эффектами. Обрезать макет до габаритов какого‑либо изображения для его экспорта – инструмент Crop Tool.

F12 – восстановить макет в том виде, в котором он сейчас сохранен на диске.

Экспорт графики для WEB

Большие фотографические (многоцветные) изображения – JPEG, качество 60‒95.

Малюсенькие картинки, вписывающиеся в 256 цветов, не нужна полупрозрачность или она бинарная – PNG‑8.

Любые картинки, где нужна полупрозрачность или отсутствие искажений, – PNG‑24.

Горячие клавиши

Полный список горячих клавиш вы можете найти на официальном сайте:

<https://helpx.adobe.com/ru/photoshop/using/default-keyboard-shortcuts.html>.

Список часто используемых горячих клавиш:

* Ctrl + A – выделить всё.
* Ctrl + C – копировать.
* Ctrl + V – вставить.
* Ctrl + N – новый документ.
* Shift + Ctrl + N – новый слой.
* Ctrl + S – сохранить.
* Shift + Ctrl + S – сохранить как.
* Ctrl + Shift + Alt + S – сохранить для Web.
* Ctrl + Z – вернуть назад.
* Alt + Ctrl + Z – вернуть на несколько шагов назад.
* Shift + Ctrl + Z – вернуть на шаг вперед.
* Ctrl (-) – уменьшение изображения.
* Ctrl (+) – увеличение изображения.
* Ctrl + Alt + 0 – натуральный размер изображения.
* Ctrl + D – снять выделение.

Часто используемые инструменты:

* V – перемещение.
* B – кисть.
* E – ластик.
* H – рука.
* M – выделение.
* P – перо.
* T – текст.

**Практическое задание**

Знакомство с макетом интернет-магазина.

1. Открыть макет интернет-магазина, визуально представить разбиение на блоки.
2. Нарезать макет любым из двух представленных способов.
3. Создать верхнюю часть сайта (как минимум, весь блок слайдера).
4. \* Добавить все эффекты наведения и нажатия на кнопки или ссылки, присутствующие в макете.

**Дополнительные материалы**

1. [Новые возможности Photoshop](http://www.adobe.com/ru/products/photoshop/features.html).
2. [Работа со слоями](https://helpx.adobe.com/ru/photoshop/using/layer-basics.html).
3. [Нарезка макета](https://habrahabr.ru/post/45948/).
4. [Скоростная нарезка макетов](https://geekbrains.ru/posts/cutting_site_template).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <https://helpx.adobe.com/ru/photoshop/using/layer-basics.html>.
2. [https://helpx.adobe.com/ru/photoshop](https://helpx.adobe.com/ru/photoshop/using/layer-basics.html).

Разметка сайта и знакомство с Bootstrap

Создание основной разметки сайта. Заполнение разметки частями нарезанного макета. Приемы позиционирования элементов. Применение Bootstrap.

[Как прижать футер к низу экрана](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.w6z6d3p3gmk)

[Способ первый](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.tvwgwyb86c5e)

[Недостаток данного приема](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.121chbe5vyq9)

[Второй способ](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.v7ho4fmgbghe)

[calc()](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.okotohiokqpv)

[vh](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.9u0wiblgbq1s)

[Недостаток данного приема](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.vnmj51kk1mju)

[Выводы](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.p2kn8pfp1a0w)

[Медиа-запросы](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.c5d9qfauyv0k)

[Синтаксис](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.bl5hyl6jbisq)

[Плюсы](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.5mtsatgh1317)

[Условия для Media Queries](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.i6fib248u2s7)

[Bootstrap](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.hbmtokrrjfb8)

[Основные преимущества](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.94z99bo1s7dc)

[Основные инструменты](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.t54bj9pgi2v6)

[Практическое задание](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.9drxs27lt8vo)

[Дополнительные материалы](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.y937sk8fclye)

[Используемая литература](https://docs.google.com/document/d/1uugCDTye4w7u91PG2AFbzxnVtUuv9JC5O9oHbt9HCXI/edit#heading=h.hdfa4xy0m98n)

# Как прижать футер к низу экрана

Когда пользователь открывает короткие по высоте страницы, сразу бросается в глаза, что информация, которая должна быть в нижней части окна просмотра, «прилипает» к содержимому страницы и находится где-то посередине, а то и вверху. Все полноценно оформленные страницы сайтов имеют «прижатый» (sticky) к низу страницы футер. Это означает, что когда контента между хедером (шапкой, header) и футером (подвалом, footer) мало, футер все равно остается внизу окна браузера, как бы вы ни изменяли размер окна. Если окно не будет умещать в себе контент и футер, появится прокрутка.

Почему нужно прижать footer к низу экрана? Это требование по умолчанию, норма, которую нужно запомнить и выполнять, тем более, что сделать это довольно просто. Иногда дизайнер может сделать макет, в котором не нужно прижимать footer к низу экрана, но в 99% при создании сайтов это потребуется. Кроме этого, прижать футер нужно, чтобы сайт корректно отображался по высоте на разных мониторах (например, на 13-дюймовом нетбуке или на 24-дюймовом настольном мониторе) и при разных разрешениях экрана.

## Способ первый

* Подвал прижат к низу экрана при высоте окна браузера больше высоты страницы независимо от контента.
* Футер находится на положенном ему месте при объеме контента больше, чем высота окна браузера.
* Работает во всех популярных браузерах.

Для примера возьмем простую страницу, состоящую из трех основных блоков: шапки (header), основного содержания (content) и подвала (footer). Сделаем так, чтобы основной блок и шапка заняли всю площадь окна браузера независимо от количества контента, при этом футер прижмем к низу экрана так, чтобы в браузере не появилась вертикальная полоса прокрутки.

|  |
| --- |
| <div class="container">          <div class="header"></div>          <div class="content">Lorem ipsum dolor sit amet, consectetur         adipisicing elit. Adipisci iure tempore similique inventore,         aliquam dicta placeat.</div>  </div>  <div class="footer"></div> |

Основной контейнер (.container) растягиваем на всю высоту экрана браузера (min-height:100%), подвалу жестко указываем высоту (height). При этом общая высота сайта составит «высота экрана + высота подвала».

Отрицательным отступом (margin-top) «въезжаем» в основной блок, чтобы высота сайта составляла только 100% высоты экрана.

|  |
| --- |
| \* { margin: 0; padding: 0; }  html, body { height: 100%; }  .container { min-height: 100%; }  .header {          height: 100px;          background-color: orange;  }  .content { background-color: green; }  .clr { height: 100px; }  .footer {          height: 100px;          background-color: red;          margin-top: -100px;          clear: both;  } |

При таком расположении блоков и при достаточном количестве контента (например, текста) в основном блоке возможны накладки контента в основном блоке на подвал. Для избежания этой потенциальной проблемы добавим в конец основного блока пустой блок высотой не меньше высоты подвала (а лучше немного больше, чтобы был отступ между контентом и подвалом).

При использовании блочной верстки и плавающих основных блоков (колонок) для .footer следует добавить clear: both, чтобы подвал расположился под колонками.

|  |
| --- |
| <div class="container">         <div class="header"></div>          <div class="content">Lorem ipsum dolor sit amet,  consectetur          adipisicing elit. Adipisci iure tempore similique inventore,          aliquam dicta placeat.</div>  <div class="clr"></div> </div>  <div class="footer"></div> |

Теперь, если контента будет много, он будет двигать пустой блок вниз. А это будет опускать и подвал, не давая налезть на него контенту.

Если вы уже немного освоили CSS, может возникнуть вопрос: «Зачем использовать дополнительный элемент, если можно воспользоваться padding-bottom?». Ответ: размер блока равен его ширине и высоте + сумме внутренних отступов + сумме толщин бордюров. Связка min-height: 100% и padding-bottom даст высоту сайта больше высоты экрана. В итоге даже при отсутствии контента вовсе подвал будет за пределами «первого экрана».

### Недостаток данного приема

* В вышеописанном приеме футер поднимали отрицательным отступом вверх. При этом возникает потенциальная проблема с z-слоями.
* Годится только для фиксированного по высоте подвала.

## Второй способ

Особенность данного способа заключается в использовании calc() и единицы измерения vh, которые поддерживаются только современными браузерами.

## calc()

Используется для указания вычисляемого значения свойств, которые в качестве значений используют размер. Это позволяет задавать значения, основанные на сложении или вычитании разных единиц измерений. Например, можно задать 100% – 20px. Если значение не может быть вычислено, оно игнорируется.

Браузеры:

* Firefox до версии 16.0 поддерживает значение -moz-calc.
* Chrome до версии 26.0 поддерживает значение -webkit-calc.
* Safari с версии 6.0 поддерживает значение -webkit-calc.
* Internet Explorer 9.0.

## vh

Относительная единица измерения – 1% от высоты области просмотра.

html

|  |
| --- |
| <div class="content">Lorem ipsum dolor sit amet, consectetur adipisicing elit.   Adipisci iure tempore similique inventore, aliquam dicta placeat.</div> <div class="footer"></div> |

сss

|  |
| --- |
| \* {  margin: 0;  padding: 0;  }  .content {  background-color: blue;  min-height: calc(100vh - 100px);  }  .footer {  background-color: green;  height: 100px;  } |

100vh – высота окна браузера.

100px – высота футера.

### Недостаток данного приема

* Поддержка только современными браузерами.
* Годится только для фиксированного по высоте подвала.

## Выводы

Существует огромное количество способов прижатия футера к низу экрана, мы рассмотрели некоторые из них. На практике вам покажут несколько методов, как разобранных в методичке, так и альтернативных. Прижатый футер – это правило хорошего тона. Сделать качественный сайт – важная часть веб-разработки.

# Медиа-запросы

Наряду с типами носителей в CSS3 включена поддержка различных технических параметров устройств, на основе которых требуется загружать те или иные стили. К примеру, можно определить смартфон с максимальным разрешением 640 пикселей и для него установить одни стилевые свойства, а для остальных устройств другие. Также можно выявить различные характеристики вроде наличия монохромного экрана, ориентации (портретная или альбомная) и др. Все характеристики легко комбинируются, поэтому допустимо задать стиль только для устройств в альбомной ориентации с заданным разрешением экрана.

Возможности медиа-запросов не ограничиваются выявлением мобильных устройств, с их помощью можно создавать адаптивный макет. Такой макет подстраивается под разрешение монитора и окна браузера, меняя при необходимости ширину, число колонок, размеры изображений и текста. Медиа-запросы ограничивают ширину макета, и при достижении этого значения (к примеру, за счёт уменьшения окна или при просмотре на устройстве с указанным размером) применяется другой стиль.

## Синтаксис

Все запросы начинаются с правила @media, после чего следует условие, в котором используются типы носителей, логические операторы и медиа-функции.

## Плюсы

* Четкое отображение страниц на экране с любым разрешением.
* Возможность просмотра группы контента на любом устройстве.
* Отсутствие горизонтальной полосы прокрутки независимо от размера окна.

## Условия для Media Queries

|  |
| --- |
| @media screen and (max-width: XXXpx) { } @media screen and (min-width: XXXpx) { }  @media screen and (min-width: XXXpx) and (max-width: YYYpx) { }  @media screen and (max-device-width: XXXpx) { } |

С их помощью можно отслеживать разрешение экрана пользователя и отображать необходимые стили для каждого разрешения или устройства.

Пример:

|  |
| --- |
| @media screen and (min-width: 1024px) {        .content {         background-color: blue;         }  } |

В результате, если у пользователя экран больше или равен 1024px, задний фон для content будет синим.

# Bootstrap

Bootstrap (также известен как Twitter Bootstrap) – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

## Основные преимущества

* Экономия времени – Bootstrap позволяет сэкономить время и усилия, используя шаблоны дизайна и классы, и сконцентрироваться на других разработках.
* Высокая скорость – динамичные макеты Bootstrap масштабируются на разные устройства и разрешения экрана без каких-либо изменений в разметке.
* Гармоничный дизайн – все компоненты платформы Bootstrap используют единый стиль и шаблоны с помощью центральной библиотеки. Дизайн и макеты веб-страниц согласуются друг с другом.
* Простота в использовании – пользователь с базовыми знаниями HTML и CSS может начать разработку с Twitter Bootstrap.
* Совместимость с браузерами – Twitter Bootstrap совместим с Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer и Opera.
* Открытое программное обеспечение – особенность Twitter Bootstrap, которая предполагает удобство использования посредством открытости исходных кодов и бесплатной загрузки.

## Основные инструменты

* Сетки – заранее заданные размеры колонок, которые можно сразу же использовать. Например, ширина колонки 140px относится к классу .span2 (.col-md-2 в третьей версии фреймворка), который можно использовать в CSS-описании документа.
* Шаблоны – фиксированный или резиновый шаблон документа.
* Типографика – описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т.п.
* Медиа – предоставляет некоторое управление изображениями и видео.
* Таблицы – средства оформления таблиц вплоть до добавления функциональности сортировки.
* Формы – классы для оформления форм и некоторых событий, происходящих с ними.
* Навигация – классы оформления для табов, вкладок, страничности, меню и тулбара.
* Алерты – оформление диалоговых окон, подсказок и всплывающих окон.

# Практическое задание

Создание новых страниц интернет-магазина.

1. Доделать главную страницу, если не доделали.
2. Создание новых страниц по желанию, выбор следующих страниц произвольный.
3. \* Познакомиться с сеткой, присутствующей в макете.

Задачи со \* предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

# Дополнительные материалы

[Несколько способов прижатия футера к низу экрана](http://dimox.name/press_footer_bottom_with_css/).

[Как прижать подвал к низу страницы](http://www.zakharov.ms/footer/).

[Прижатый футер своими руками](https://habrahabr.ru/post/151320/).

[Адаптивный дизайн на основе медиа запросов](https://habrahabr.ru/company/google/blog/173861/).

[Улучшение отзывчивости с помощью медиазапросов CSS](https://developers.google.com/web/fundamentals/design-and-ui/responsive/fundamentals/use-media-queries?hl=ru).

[Адаптивная вёрстка](http://alexdev.ru/1497/).

[Bootstrap от новичка до профессионала](http://bootstrap-3.ru/index.php).

# Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.zakharov.ms/footer/>.
2. <http://htmlbook.ru/css/value/media>.
3. <https://developers.google.com/web/fundamentals/design-and-ui/responsive/fundamentals/use-media-queries?hl=ru>.
4. <http://www.wcdt.com.ua/yak-prytysnuty-footer-do-nyzu-ekrana-za-dopomohoyu-sss/>.
5. <http://alexdev.ru/1497/>.
6. <http://getbootstrap.com/>.

Стандарты web

и вспомогательные инструменты

Загрузка проектов на сервер. Проблема «кроссбраузерности». Стандарты HTML/CSS. Будущее за стандартами HTML5 и CSS3. Грамотная, универсальная верстка. Знакомство с препроцессорами на примере LESS.

[Инструменты разработчика](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.qfb63rro8gxy)

[Ftp-клиент](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.cm7rewg0t2vk)

[Плагины](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.5xka83qwfyr8)

[Валидатор](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.2ccl6ikbss0c)

[Оптимизация изображений](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.5lagnj2fwpnq)

[Генератор CSS-градиентов](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.mdb2cd7hshko)

[Совместимости разных версий браузеров](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.crksfdnfvdb0)

[Проблема «кроссбраузерности»](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.ytqvmv91w6f)

[Условный комментарий](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.ytvxpbx56gla)

[Стандарты HTML/CSS](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.gbflfbkwe021)

[Версии HTML](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.5q05m58x3i8t)

[Версии CSS](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.nd4cbqxgauuj)

[Будущее за стандартами HTML5 и CSS3](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.2pr8w62sxjd)

[Семантические элементы](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.8no6tgv37cvo)

[Сокращенная форма атрибутов в разделе head](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.abvzs4zax1te)

[Атрибуты формы](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.n908dsxcmlf0)

[Canvas](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.6mw54jrh0wo1)

[Что нам дарит CSS3](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.1ugvt44sdok9)

[Тени вокруг блоков и текста](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.1wk105b2scfm)

[Несколько фоновых изображений](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.olpqrqypsj6q)

[Прозрачность](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.wmr67ogu8pju)

[Препроцессоры](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.jrceeyu7innf)

[Что такое препроцессоры](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.jpzh45vz3hda)

[LESS](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.7thng2osi9b0)

[Переменные](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.3lxl72xh512d)

[Операции](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.fkghho8cpn3d)

[Цветовые операции](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.wswa41ddum1x)

[Примеси (mixins)](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.7nbbxh600ne4)

[Заключение](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.vis2gbw5dmq9)

[Практика](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.35yawhiqvtbz)

[Применение препроцессора LESS](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.69nq3b6y1irv)

[Практическое задание](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.8yyjr5jjqvjk)

[Дополнительные материалы](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.lgrp0tj4m2m8)

[Используемая литература](https://docs.google.com/document/d/1Qq__MNsIZZYtooRXpk1bhpHeKS4JWs0uUd83_uN0XG8/edit#heading=h.uvp6qax5r1ok)

**Инструменты разработчика**

Какие инструменты могут понадобиться веб-разработчику:

* ftp-клиент (FileZilla, WinSCP, Total Commander . . .).
* Плагины (FireBug, WebDeveloper).
* Валидатор (http://validator.w3.org/).
* Оптимизация изображений (TinyPNG).
* Генератор CSS-градиентов (CSS3Factory, CSSMatic Gradients).
* Совместимости разных версий браузеров (Can I Use …?).
* Тестер для IE (IETester).

Ftp-клиент

FTP – это протокол передачи файлов. Через него файлы можно скопировать с компьютера на сервер и обратно, отредактировать.

Для использования FTP необходима специальная программа. Их существует большое количество, некоторые из них бесплатные, другие могут быть коммерческими. Практически все программы FTP работают таким образом: появляются 2 окна, где 1-е – ваши файлы на сервере в интернете, а 2-е –  файлы на компьютере. Тем самым можно легко передавать файлы на компьютер/сервер.

Предположим, программа FTP уже выбрана и установлена на компьютер, но для работы необходимо знать FTP-адрес вашего веб-сервера, имя пользователя и пароль. Это важная информация, ее можно получить на хостинге и нужно обязательно записать. Если все данные есть, можно попробовать получить доступ к файлам FTP.

В различных программах действия могут различаться, но в любом случае нужно создать новое соединение с именем пользователя и паролем, что позволит сохранить настройки, тем самым исключая необходимость ввода данных каждый раз при входе в программу.

После создания профиля и ввода всех данных нажмите «Соединиться», после чего программа начнет соединяться с сервером. Если все настройки верны, через несколько секунд (в зависимости от скорости интернета) можно будет увидеть все свои файлы, находящиеся на сайте.

На веб-сервере находится набор папок и файлов. Подключаясь к серверу, программа отправляет запрос через интернет на специальный сервер и получает данные. Существуют протоколы и с более расширенным функционалом, нежели обычный веб-серфинг.

Используемая FTP-программа дает возможность создавать, редактировать и удалять файлы и папки на веб-сервере, а также передавать эти файлы на сервер и обратно.

Плагины

Есть несколько способов открыть:

* Control + SHIFT + I – открывает панель элементов.
* Control + SHIFT + J или F12 – открывает консоль.
* Control + SHIFT + C – позволяет выбрать элемент с помощью мыши в самом документе и открыть его во вкладке «Elements».
* Можно нажать на любой элемент на странице правой кнопкой мыши и выбрать «Inspect Element (Просмотр кода элемента)».
* Можно открыть Web Inspector из меню: Иконка с ключом -> Инструменты -> Инструменты разработчика (Tools -> Developer tools).

Чтобы посмотреть список горячих клавиш, откройте любую вкладку, кроме консоли, и нажмите там «?».

При помощи просмотра кода элемента можно инспектировать любые сайты, четко определять структуру сайта, вносить изменения в код и многое другое.

Валидатор

Чистый валидный код в HTML и CSS повышает скорость загрузки и шансы, что на всех без исключения устройствах и браузерах ваша верстка будет выглядеть так, как вы хотите. Поэтому любой верстке нужно проводить валидацию. Валидаторы W3C – два сервиса, которые помогают держать код в чистоте и порядке.

Есть три варианта проверки: проверить код в файле по указанной ссылке, загрузить файл с десктопа или скопировать код прямо в форму на сайте.

HTML-валидатор производит несколько проверок кода:

* Валидация синтаксиса – проверка на наличие синтаксических ошибок.
* Проверка вложенности тегов – они должны быть закрыты в обратном порядке относительно их открытия.
* Валидация DTD – проверка соответствия вашего кода указанному Document Type Definition. Она включает проверку названий тегов, атрибутов и «встраивания» тегов.
* Проверка на посторонние элементы – выявляет все, что есть в коде, но отсутствует в DTD.

Оптимизация изображений

Оптимизация графики – не менее важный аспект верстки, чем чистота кода. Функция «Save for Web» в Photoshop незаменима, но с png она работает крайне плохо. Поэтому после экспорта png-файлов из фотошопа рекомендуется оптимизировать графику в TinyPNG. После обработки файл теряет 30-70% веса, но на качество это никак не влияет.

Генератор CSS-градиентов

CSS-градиенты позволяют сделать фон из двух и более цветов, плавно переходящих из одного в другой. Они применяются уже достаточно давно и неплохо поддерживаются браузерами. Большинство современных браузеров понимает их без префиксов, для IE9 и старше есть Gradient Filter, также для IE9 можно использовать SVG.

Градиенты могут быть использованы везде, где есть картинки: в фонах, в качестве буллетов списков, в content или border-image.

Совместимости разных версий браузеров

Сайт будет отображаться и работать во всех популярных браузерах одинаково, когда отображение параметров страниц при использовании разных браузеров, их версий и модификаций близко к исходному дизайнерскому и функциональному виду сайта. Необходимо пользоваться специальными программами, чтобы сайт выглядел во всех браузерах одинаково. Инструмент, который приходится использовать ради IE, – тестировщик IETester. Он позволяет проверять, как выглядит сайт во всех версиях IE, начиная с 5.5.

**Проблема «кроссбраузерности»**

В процессе работы любой веб-разработчик сталкивается с таким понятием, как кроссбраузерность. Оно означает, что любой сайт должен выглядеть одинаково во всех браузерах. Главным виновником этой проблемы является Internet Explorer (IE). Рассмотрим пути решения проблемы кроссбраузерности:

1. Необходимо проверять, как выглядит сайт во всех популярных браузерах, с некоторой периодичностью во время верстки. Тогда вы сможете оперативно устранять несоответствия.
2. Некоторые теги HTML, например заголовки, параграфы, списки, изначально имеют определенный набор свойств и значений. Они могут определяться каждым браузером по-разному. Чтобы вид HTML страницы не зависел от того, с помощью какого браузера ее просматривают, используется сброс стилей CSS. Обычно, если используется сброс стилей, их определяют в отдельном файле, который подключают на веб-страницу при помощи тега <link />.
3. Существует возможность воспользоваться хаками. Хак – исправление ошибки или добавление новой функции посредством использования другой недокументированной или некорректно реализованной особенности. Возникает необходимость добавлять хаки в основном для IE, но ими пользоваться не рекомендуется, т.к. это противоречит спецификации CSS. Чтобы исправить несоответствия отображения в IE, лучше воспользоваться следующим пунктом.
4. Объявить условный комментарий. Он представляет собой обычный HTML-комментарий, где в квадратных скобках указывается условие, которое понимает только браузер IE, а все остальные браузеры игнорируют.

Условный комментарий

|  |
| --- |
| <!--[if gte IE 6]>         <link href="css/style\_ie.css" rel="stylesheet" type="text/css" />  <![endif]--> |

То, что находится между if и endif, IE выполнит. В данном случае подключится файл стилей CSS. В нем можно писать только те стили CSS, которые будут относиться к IE. Для того чтобы точно определять, к каким версиям IE применять стили, можно использовать дополнительные логические операторы:

* lt – версия меньше указанной.
* lte – версия меньше или равна указанной.
* gt – версия больше указанной;
* gte – версия больше или равна указанной.

При помощи других логических операторов (& – и, | – или, ! – не) выполняется группировка условных комментариев для различных версий IE:

* [if (IE 6) & (IE 7)] – 6-я версия И 7-я версия.
* [if (IE 6) | (IE 7)] – 6-я версия ИЛИ 7-я версия.
* [if !(IE 8)] – НЕ 8-я версия.

**Стандарты HTML/CSS**

Версии HTML

Стандарт HTML зародился в конце 80-х – начале 90-х годов. Но только в 1995 году был принят стандарт HTML версии 2.0. Затем он активно развивался до 2000 года, остановилось развитие на версии 4.01 Strict. Как раз в это время начал развиваться стандарт XHTML, который основывается на языке представления данных XML и предъявляет к документам более строгие правила оформления. Существуют три версии стандарта XHTML: версии 1.0, 1.1 и 2.0. Но версия 2.0 не достигла статуса рекомендации, и к 2010 деятельность по разработке стандарта XTHML прекращена. Группа разработчиков переведена на разработку стандарта HTML5, которая продолжается в настоящее время.

Версии CSS

CSS версии 1 была принята в 1996 году, а в 1998 году она была расширена версией 2. Рабочий проект CSS версии 2.1 появился в 2006 году, а был принят организацией w3c в 2011 году. В нем были исправлены ошибки версии 2 и изменены некоторые части спецификации. В настоящее время разрабатывается CSS версии 3, которая имеет множество нововведений и дополнений.

**Будущее за стандартами HTML5 и CSS3**

За один урок невозможно рассказать о нововведениях и преимуществах новых стандартов HTML5 и CSS3, поэтому рассмотрим то, что можно применять уже сегодня. HTML5 и CSS3 уже поддерживаются всеми современными браузерами, включая IE, начиная с 9-й версии. Но существует проблема: 9-ю версию IE можно установить только на ОС Windows Vista или Windows 7. Компания Microsoft отказалась от поддержки Windows XP, но у многих пользователей он еще установлен, и пройдет еще немало времени, пока XP не исчезнет с компьютеров пользователей. Такие пользователи не могут обновить IE дальше 8 версии. Новые теги HTML5 можно включить в документ при помощи JavaScript, а в случае с CSS3 пользователь, просматривающий сайт через IE меньше 9 версии, не увидит эффектов применения стилей CSS3.

При помощи HTML5 можно заменить основные элементы в структуре макета специальными тегами и тем самым более четко представить структуру сайта.

Семантические элементы

<section>является способом разделения страницы или статьи на тематические разделы. Не используется многократно, обычно содержит заголовок.

<header> образует содержимое верхней части («шапки») страницы или ее секции. Объединяет вводную информацию и навигационные элементы, может располагаться в любой части страницы. В HTML-документе может содержаться одновременно несколько элементов <header>. Не является обязательным.

<footer> формирует содержимое нижней части («подвала») страницы или ее секции. Предназначен для размещения информации о веб-ресурсе, например сведений об авторских правах, ссылок на условия использования, контактной информации, ссылок на связанное содержимое и т.п. В одном веб-документе может быть несколько элементов <footer>.

<aside> группирует содержимое, связанное с окружающим его контентом напрямую, но которое можно счесть отдельным. Чаще всего элемент позиционируется как боковая колонка.

<hgroup> используется для группировки элементов <h1>-<h6> в случае, когда заголовок имеет сложную структуру – например, имеет уточняющие подзаголовки, альтернативные заголовки.

<nav> предназначен для создания блока навигации веб-страницы или всего веб-сайта, при этом не обязательно должен находиться внутри <header>. На странице может быть несколько элементов <nav>.

<article> используется для группировки записей: публикаций, статей. Представляет собой независимый, обособленный блок, предназначенный для многократного использования.

<main> группирует основной контент веб-страницы. Содержимое элемента должно быть уникальным на странице и не должно отображаться где-либо еще на сайте.

<section> является способом разделения страницы или статьи на тематические разделы.

Возможность вставки аудио и видео без использования Flash появилась благодаря новым тегам, добавленным в HTML5.

Сокращенная форма атрибутов в разделе head

* <!DOCTYPE html>
* <meta charset=”UTF-8″>
* <script src=”script.js”></script>
* <link rel=”stylesheet” href=”styles.css”>

Атрибуты формы

Появились богатые возможности по работе с формами,  масса новых типов полей, например e-mail, tel, url. Благодаря этим типам больше не нужно писать код на JavaScript для проверки формы. Теперь с помощью HTML5 форма автоматически проверяется на корректность заполнения, сразу выдавая все сообщения об ошибках пользователю. Причем все это сделано очень красиво и аккуратно, к тому же внешний вид можно будет изменить через CSS. Также появился элемент datalist, позволяющий создать список вариантов, которые пользователь сможет выбрать при наборе текста. Такой функционал часто используется при наборе поискового запроса, когда при вводе первых символов сразу появляются возможные варианты. Никогда еще сложная функциональность не создавалась так легко.

Также упрощена реализация drag and drop. Пример его использования – перетащили мышкой картинку товара в корзину, и он автоматически туда добавился. Раньше это была достаточно сложная задача, теперь же она делается в течение нескольких минут.

Canvas

Позволяет рисовать прямо на веб-странице через JavaScript, генерировать картинки без перезагрузки страницы. Раньше без использования PHP это было невозможно.

Что нам дарит CSS3

Закругленные уголки у блоков задаются при помощи свойства border-radius.

Можно задавать скругления как всем четырем сторонам вместе, так и каждой в отдельности, тогда отсчет у уголков производится от левого верхнего угла и идет по часовой стрелке. Также возможно задать значения скругления через слеш, тогда первое значение будет задавать скругление по горизонтали, а второе – по вертикали. Таким образом из блока можно сделать эллипс.

Тени вокруг блоков и текста

Для блоков:

|  |
| --- |
| box-shadow: 2px 3px 4px #ccc; |

Для текста:

|  |
| --- |
| text-shadow: 2px 3px 4px #ccc; |

* 2px – смещение по оси X.
* 3px – смещение по оси Y.
* 4px – радиус размытия.
* #ccc – цвет тени.

Несколько фоновых изображений

Чтобы задать несколько картинок для одного элемента в качестве фонового изображения, их необходимо перечислить через запятую для свойства background:

|  |
| --- |
| background:  url(img/HTML5.png) top center no-repeat,              url(img/css3.png) top right no-repeat; |

Прозрачность

При помощи свойства opacity можно задавать элементу полупрозрачность. Значение указывается в интервале от 0 до 1, где 0 – элемента не будет видно, но при этом он не исчезнет из потока документа, как в случае со свойством display: none. А при значении 1 элемент будет четко виден:

|  |
| --- |
| opacity: 0.1 .... 1; |

Появилась возможность управлять элементами на странице. Если задать для многострочного поля ввода <textarea> свойство resize, пользователь получает возможность растягивать данное поле. При значении horizontal можно растянуть его по горизонтали, vertical – по вертикали, а при значении both – и по горизонтали, и по вертикали.

**Препроцессоры**

Что такое препроцессоры

CSS-препроцессоры – это «программистский» подход к CSS. Они позволяют использовать при написании стилей  свойственные языкам программирования приемы и конструкции: переменные, вложенность, наследуемость, циклы, функции и математические операции. Синтаксис препроцессоров похож на обычный CSS. Код, написанный на языке препроцессора, не используется прямо в браузере, а преобразуется в чистый CSS-код с помощью специальных библиотек.

LESS

* Динамический язык стилей.
* Продукт с открытым исходным кодом.
* Может работать на стороне клиента или на стороне сервера под управлением Node.js или Rhino.

Переменные

Синтаксис переменных:

|  |
| --- |
| @название\_переменной: значение\_переменной; |

Создав переменную один раз, можно использовать ее в любом месте кода.

Например:

|  |
| --- |
| @color\_red: #f00; background-color: @color\_red;  color: @color\_red;  border-color: @color\_red; |

Во всех местах, где указана переменная, LESS заменит строку @color\_red на #f00. Теперь, если понадобится изменить цвет, не нужно искать все его объявления в файле, достаточно просто изменить значение переменной в одном месте.

Область видимости переменных:

* Переменные можно объявлять как «снаружи» правил, так и «внутри». В случае «внутреннего» объявления переменная будет доступна только внутри правила, в котором она объявлена.
* Если переменная объявлена и «внутри» правила, и «снаружи», LESS применит «внутреннее» значение.
* Таким образом можно «переопределять» глобальные переменные в локальном контексте.

Важно отметить: вопреки тому, что вы только что прочитали, переменные в LESS больше похожи на константы. Это значит, что они, в отличие от переменных, могут быть определены только один раз.

Операции

Можно использовать операции умножения, деления, сложения и вычитания.

|  |
| --- |
| @width: 100px; .class\_1 {          width: @width; } .class\_2 {          width: @width / 3; } .class\_3 {          height: 100px + 20px; } |

Цветовые операции

На практике есть немало случаев, когда мы начинаем с базового цвета и нуждаемся в слегка затемненном или осветленном его варианте.

|  |
| --- |
| @color-button: #ccc; .button {         width: 150px;          height: 75px;         background:@color-button;          border:5px solid @color-button - #222; } |

Этот код создает кнопку с немного затемненной рамкой. Это частая ситуация и определение лишь одного цвета – большая помощь.

Есть гораздо больше возможностей для работы с цветами. LESS позволяет манипулировать ими на канальном уровне:

* Осветлять – lighten.
* Затемнять – darken.
* Насыщать – saturate.
* Обесцвечивать – desaturate.
* Вращать цвета – spin.

Примеси (mixins)

Примеси в LESS избавят вас от набора излишнего кода. Вам когда-нибудь приходилось создавать закругленную рамку, в которой только верхние углы скруглены?

|  |
| --- |
| .border\_top {             -webkit-border-top-left-radius: 6px;              -webkit-border-top-right-radius: 6px;              -moz-border-radius-topleft: 6px;              -moz-border-radius-topright: 6px;               border-top-left-radius: 6px;               border-top-right-radius: 6px; } .block { background: #333; .border\_top; } |

Благодаря такому синтаксису мы можем использовать любой элемент в качестве примеси.

Заключение

Препроцессоры – очень удобное изобретение для всех разработчиков. Это сокращение кода, возможность создания большого функционала сайтов на простом CSS.

Использовать препроцессоры стало намного проще, чем раньше. Для этого нужно лишь установить программу, которая будет следить за файлами, предназначенными для препроцессора, и при их изменении будет компилировать содержимое этих файлов в чистый CSS-код. Для более продвинутых пользователей есть специальные сборщики проектов, которые предлагают полный контроль и расширенные настройки.

Разумеется, как и в любой другой области, на рынке препроцессоров есть конкуренция. Они очень похожи, имеют минимальные отличия. Какой лучше выбрать, какой синтаксис лучше подходит – выбирать вам!

**Практика**

Применение препроцессора LESS

|  |  |
| --- | --- |
| <header></header> <main></main> <footer>     <div class="block\_1"></div>     <div class="block\_2"></div>     <div class="block\_3"></div> </footer> | @width: 1400px; @height: 300px; \* {   margin:0;   padding:0; } header {   width:@width;   height:200px;   background-color: bisque;   margin:0 auto; } main {   width:@width;   height:@height \* 2;   background-color: cadetblue;   margin:0 auto; } footer {   width:@width;   height:@height;   background-color: blueviolet;   margin:0 auto;   .block\_1, .block\_2, .block\_3 {     width:@width / 3;     float: left;     height:@height;   }   .block\_1 {     background-color: aqua;   }   .block\_2 {     background-color: #3074ff;   }   .block\_3 {     background-color: #42ffba;   } } |

**Практическое задание**

**Финальный проект**

1. Установить все перечисленные инструменты к себе на компьютер.
2. При помощи валидатора проверить свой сайт на ошибки.
3. Переделать структуру вашего сайта под HTML 5 при помощи новых тегов.
4. Добавить в форму регистрации новые типы полей и атрибутов HTML 5.
5. Добавить на сайт некоторые эффекты CSS 3.
6. Закачать проект на хостинг.
7. Выслать ссылку на ваш сайт в Интернете.
8. \* Переделать стили с использованием препроцессорного языка.

Пожалуйста, пройдите короткий опрос, чтобы оценить время выполнения домашнего задания: https://forms.gle/cBoYMPvbABa36jb7A

**Дополнительные материалы**

[Немного о кроссбраузерности](http://www.a3s.biz/nemnogo-o-krossbrauzernosti/).

[Кроссбраузерность веб-сайта](http://www.ldi.ru/krossbrauzernost-veb-sayta-problemy-sovmestimosti.html).

[Новые атрибуты html5](http://webformyself.com/vvedenie-v-formy-html5-i-novye-atributy/).

[Использование html5](http://ruseller.com/lessons.php?rub=43&id=2020).

[HTML5 для начинающих](http://html-5.ru/formy-html5).

[Примеры CSS3](http://www.designonstop.com/tutorials/my-tuts/20-vpechatlyayushhix-primerov-vozmozhnostej-i-funkcij-css3-texnologij.htm).

[Новое в CSS3](https://habrahabr.ru/post/153925/).

[СSS Препроцессоры](https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter_1/css-reprocessors.html).

[Начинаем писать CSS](http://frontender.info/starting-css/).

**Используемая литература**

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C>.
2. <http://htmlbook.ru/html5>.
3. <https://ru.wikipedia.org/wiki/CSS#.D0.A3.D1.80.D0.BE.D0.B2.D0.B5.D0.BD.D1.8C_3_.28CSS3.29>.
4. <http://htmlbook.ru/css3>.
5. <http://html5book.ru/html5-semantic-elements/>.
6. <https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter_1/css-reprocessors.html>.
7. <http://cmsart.ru/joomla/joomla-articles/joomla-1-7/chto-takoe-ftp-i-dlya-chego-on-nuzhen>.