



## Урок 1 Создаем свою первую веб-страницу

**HTML** (от англ. **Hyper Text Markup Language** — «язык гипертекстовой разметки») — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере.

**Гипертэкст** (англ. hypertext) — термин, обозначающий систему из текстовых страниц, имеющих перекрёстные ссылки. Примерами гипертекста являются энциклопедии, компьютерные сети, веб-сайты, в которых можно переходить с одной страницы на другую и выполнять поиск по ключевым словам.

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

**Тег, тэги** (более правильное название — **дескриптор**) — элемент языка разметки гипертекста.

Теги бывают парные, например: `<i></i>` - где `<i>` - открывающий тег, а `</i>` - закрывающий тег и бывают одиночными, например `<br>`.

Текст, содержащийся между начальным и конечным тегом, отображается и размещается в соответствии со свойствами, указанными в начальном теге. Например, текст заключённый между начальным тегом `<small>` и конечным тегом `</small>` (от англ. small — маленький), отображается с меньшим размером, чем основной текст, а текст между тегами `<big>` и `</big>` отображается с большим размером.

Элемент `<!DOCTYPE>` предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях. Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать `<!DOCTYPE>`.

В данный момент для свежих версий HTML5 + актуальна запись: `<!DOCTYPE html>`

Тег `<html>` является контейнером, который заключает в себе все содержимое веб-страницы, включая теги `<head>` и `<body>`. Открывающий и закрывающий теги `<html>` в документе необязательны, но хороший стиль диктует неперемное их использование. Как правило, тег `<html>` идет в документе вторым, после определения типа документа, устанавливаемого через элемент `<!DOCTYPE>`. Закрывающий тег `</html>` должен всегда стоять в документе последним.

```
<html>
```

Все содержимое веб-страницы

```
</html>
```

Тег `<head>` предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Также внутри контейнера `<head>` находятся метатеги, которые используются для хранения информации, предназначенной для браузеров и поисковых систем. Содержимое тега `<head>` не отображается напрямую на веб-странице, за исключением тега `<title>` устанавливающего заголовок окна веб-страницы.



**<meta charset="utf-8">** - указывает кодировку документа. UTF-8 (от англ. Unicode Transformation Format, 8-bit — «формат преобразования Юникода, 8-бит») — распространённый стандарт кодирования символов, позволяющий более компактно хранить и передавать символы Юникода используя переменное количество байт (от 1 до 4), и обеспечивающий полную обратную совместимость с 7-битной кодировкой ASCII. Кодировка UTF-8 сейчас является доминирующей в веб-пространстве.

Элемент **<body>** предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера **<body>**. К такой информации относится текст, изображения, теги и т.д.

Открывающий и закрывающий теги **<body>** на веб-странице не являются обязательными, однако хорошим стилем считается их использование, чтобы определить начало и конец HTML-документа.

## Урок 2 Изучаем первые теги и оформляем с их помощью веб-страницу

**Lorem** и нажать клавишу TAB – команда создания «рыбного» текста на английском языке.

**Loremru** и нажать клавишу TAB – команда создания «рыбного» текста на русском языке.

**<p></p>** - тег HTML. Определяет текстовый абзац. Тег **<p>** является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

**<b></b>** - тег HTML. Устанавливает жирное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

**<i></i>** - тег HTML. Устанавливает курсивное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

**<ul></ul>** - тег HTML. устанавливает маркированный список.

**<ol></ol>** - тег HTML. устанавливает нумерованный список.

Каждый элемент списка должен оборачиваться тегами **<li></li>**.

## Урок 3 Подключаем css файл и делаем текст красивым

**CSS** (англ. Cascading Style Sheets «каскадные таблицы стилей») — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML).

**<link rel="stylesheet" href="style.css" type="text/css">** - устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. Тег **<link>** размещается всегда внутри контейнера **<head>**.

**rel** - определяет отношения между текущим документом и файлом, на который делается ссылка. В данном случае указано, что нужно относиться к подключаемому файлу, как к таблице стилей.



**href** - путь к связываемому файлу.

**type** - MIME-тип данных подключаемого файла.

Формат записи CSS стилей: **селектор { свойство: параметр; свойство: параметр; }**

В данном случае мы записали два свойства для селектора.

**background-color** – свойство CSS. определяет фоновый цвет элемента.

**color** – свойство CSS. определяет цвет текста внутри элемента.

## Урок 4 Классы и идентификаторы

Селекторы могут быть простыми: **тег, класс, идентификатор** или составными: состоящие из нескольких вложенных элементов.

**class** – в HTML атрибут тэга – в CSS вызывается через . (точку)

**id** – в HTML – в CSS вызывается через # (шарп)

Класс в HTML может быть назначен множеству элементов, идентификатор в HTML может быть назначен только одному элементу.

## Урок 5 Добавляем картинки на веб-страницу

****

Тег **<img>** предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут **src**. Через атрибут **alt** – задаётся альтернативный текст, который отображается, когда картинка недоступна на странице (неверный адрес, не успела загрузиться и т.д.)

**width** – свойство CSS. Определяет ширину изображения или элемента. Может задаваться в пикселях (px), процентах от ширины экрана или в относительных единицах.

**height** - свойство CSS. Определяет высоту изображения. Может задаваться в пикселях (px), процентах или в относительных единицах.

!!! Чтобы не нарушить пропорции изображения, указывают только один из параметров (width или height) или же задают второму параметру значение auto.

**border** - свойство CSS. Определяет обводку изображения. Свойство состоит из трех параметров:

толщина обводки, тип линии (solid – сплошная, dotted – точки, dashed – штрихи), цвет.

**border: 1px solid red;** - в примере обводка будет выполнена линией со следующими характеристиками: линия толщиной один пиксель, сплошная и красного цвета.

**border-radius** – свойство CSS. Определяет скругление углов обводки. Может задаваться в пикселях (px), процентах или в относительных единицах.

## Урок 6 Учимся работать с блочными элементами

**<div></div>** - тег HTML. Блочный элемент, предназначенный для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с



помощью стилей `css`. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут `class` или `id` с именем селектора. Как и при использовании других блочных элементов (например, тег `p`), содержимое тега `<div>` всегда начинается с новой строки. После него также добавляется перенос строки.

**float** – свойство `CSS`. Определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с других сторон.

Чаще всего используется конструкция: **float: left;** - выравнивающая блоки по левому краю, при этом границы блока сжимаются до размеров содержимого.

**clear** - свойство `CSS`. Устанавливает, с какой стороны элемента запрещено его обтекание другими элементами. Если задано обтекание элемента с помощью свойства `float`, то `clear` отменяет его действие для указанных сторон.

Чаще всего используется конструкция: **clear: both;** - которая отменяет обтекание элемента одновременно с правого и левого края. Это значение рекомендуется устанавливать, когда требуется снять обтекание элемента, но неизвестно точно с какой стороны.

**box-sizing** - свойство `CSS`. Применяется для изменения алгоритма расчета ширины и высоты элемента. Согласно спецификации `CSS` ширина блока складывается из ширины контента (`width`), значений отступов (`margin`), полей (`padding`) и границ (`border`). Аналогично обстоит и с высотой блока. Свойство `box-sizing` позволяет изменить этот алгоритм, чтобы свойства `width` и `height` задавали размеры не контента, а размеры блока.

Чаще всего используется конструкция: **box-sizing: border-box;** - свойства `width` и `height` включают в себя значения полей и границ, но не отступов (`margin`).

## Урок 7 Продолжаем тему блочных элементов и создаем структуру сайта

Чаще всего логические блоки на сайте называют:

**header** – шапка сайта

**aside** – колонка на сайте. Бывает еще `left-aside`, `right-aside`

**content** – блок с основным контентом сайта

**footer** – нижняя часть или (подвал сайта)

**padding** - свойство `CSS`. Устанавливает значение полей внутри элемента и вокруг его содержимого. Поле называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое.

Свойство `padding` позволяет задать величину поля сразу для всех сторон элемента или определить ее только для указанных сторон.

Примеры записи свойств:

**padding: 10px;** - поля будут установлены одновременно от каждого края элемента.

**padding: 10px 20px;** - первое значение устанавливает поля от верхнего и нижнего края (10px), второе — от левого и правого (20px).



**padding: 10px 20px 30px 40px;** - поочередно устанавливаются поля от верхнего (10px), правого (20px), нижнего (30px) и левого (40px) края.

Также возможно отдельное написание для каждой стороны по отдельности:

**padding-top: 10px;** - отступ внутри блока сверху

**padding-right: 20px;** - отступ внутри блока справа

**padding-bottom: 30px;** - отступ внутри блока снизу

**padding-left: 40px;** - отступ внутри блока слева

**text-align** - свойство CSS. Определяет горизонтальное выравнивание текста в пределах элемента. Бывают: center – выравнивание по центру, justify – выравнивание по ширине, left – выравнивание по левому краю, right – выравнивание по правому краю.

Пример: **text-align: center;** - выравнивание текста по центру. Текст помещается по центру горизонтали окна браузера или контейнера, где расположен текстовый блок. Строки текста словно нанизываются на невидимую ось, которая проходит по центру веб-страницы. Подобный способ выравнивания активно используется в заголовках и различных подписях, вроде подрисовочных, он придает официальный и солидный вид оформлению текста. Во всех других случаях выравнивание по центру применяется редко по той причине, что читать большой объем такого текста неудобно.

### Размещение фонового изображения для блока:

**Background** - универсальное свойство CSS, позволяет установить одновременно до пяти характеристик фона. Значения могут идти в любом порядке, браузер сам определит, какое из них соответствует нужному свойству.

**background: url(..img/1.jpg);** - в данном примере, с помощью этого свойства мы устанавливаем фоновую картинку.

**background-size: cover;** - свойство CSS, которое масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.

**background-position: center;** - задает начальное положение фонового изображения, в данном примере изображение будет масштабироваться от центра. По умолчанию начальное положение располагается в левом верхнем углу.

## Урок 8 Изучаем гиперссылки

Тег **<a>** является одним из важных элементов HTML и предназначен для создания гиперссылок.

**href** – атрибут HTML тега **<a>**, задает адрес документа, на который следует перейти. В качестве значения используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным (например, <https://ya.ru>) и относительным (например, </second.html>). Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где



прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта.

**title** - атрибут HTML тега `<a>`, добавляет всплывающую подсказку к тексту ссылки.

`<a href="https://ya.ru">Яндекс</a>` - в примере ссылка ведет на страницу **ya.ru**, по протоколу **https**, а на странице сайта отображается текст ссылки: **Яндекс**.

`<a href="/second.html#ancor123">Якорь</a>` - пример анкорной (якорной ссылки). Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. Закладка указывается как `id` - элемента. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

### Про протоколы передачи данных:

**HTTP** (Hyper Text Transfer Protocol) — это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером.

**HTTPS** (Hyper Text Transfer Protocol Secure) — это расширение протокола HTTP. Оно позволяет существенно снизить риск перехвата персональных данных посетителей (логины, пароли, номера банковских карт и т. д.), а также избежать подмены контента, в том числе рекламы, при загрузке сайта.

**FTP** (File Transfer Protocol) — протокол передачи файлов.

### Псевдоклассы, применяемые в CSS для стилизации состояний элементов:

**:hover** - определяет стиль элемента при наведении на него курсора мыши, но при этом элемент еще не активирован, иными словами кнопка мыши не нажата.

**:active** - определяет стиль для активной ссылки. Активной ссылка становится при нажатии на нее.

**:visited** - применяется к ссылкам, уже посещенным пользователем, и задаёт для них стилевое оформление.

## Урок 9 Верстка меню в виде кнопок

**margin** - свойство CSS. Устанавливает значение полей вокруг элемента и позволяет задавать расстояния взаимодействия с окружающими элементами. Свойство `margin`, аналогично свойству `padding` позволяет задать величину поля сразу для всех сторон элемента или определить ее только для указанных сторон.

Примеры записи свойств:

**margin: 10px;** - поля будут установлены одновременно от каждого края элемента.

**margin: 10px 20px;** - первое значение устанавливает поля от верхнего и нижнего края (10px), второе — от левого и правого (20px).

**margin: 10px 20px 30px 40px;** - поочередно устанавливается поля от верхнего (10px), правого (20px), нижнего (30px) и левого (40px) края.



Также возможно отдельное написание для каждой стороны по отдельности:

**margin -top: 10px;** - отступ снаружи блока сверху

**margin -right: 20px;** - отступ снаружи блока справа

**margin -bottom: 30px;** - отступ снаружи блока снизу

**margin -left: 40px;** - отступ снаружи блока слева

**transition** – универсальное CSS свойство, которое позволяет одновременно задать значения таких свойств, как transition-property, transition-duration, transition-timing-function и transition-delay. Устанавливает эффект перехода между двумя состояниями элемента, они могут быть определены с помощью псевдокласса :hover

**transition: 2s;** - в примере мы устанавливаем плавный переход состояний псевдоклассов.

## Урок 10 Адаптивные страницы и медиа запросы

В данном видеоуроке мы настроим правильное отображение блоков страницы сайта при различных разрешениях экрана, чтобы наша страница красиво отображалась на мониторах компьютеров и на мобильных устройствах.

Медиа-запросы (media queries) – это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств. Иными словами, это конструкции, которые позволяют определять на основании некоторых условий какие стили необходимо использовать на веб-странице, а какие нет.

Создание медиа-запроса начинается с ключевого слова @media после которого указывается одно или несколько условий. В качестве условия можно указывать тип устройства или требования к определённой характеристике. Требование к определённой характеристике записывается в круглых скобках.

Комбинирование нескольких условий выполняется с помощью логических операторов.

После составления @media, стили, указанные в нём, будут применяться только в том случае, когда итоговый результат вычисления условий является истинной.

Пример медиа-запроса с комбинированием нескольких условий:

```
@media (min-width: 992px) and (max-width: 1199.98px) { ... }
```

В @media можно указывать определённые типы устройств:

- all – для всех;
- print – для принтеров и в режиме предварительного просмотра страницы перед печатью;
- screen – для устройств с экранами;
- speech – для программ чтения с экрана.

## Урок 11 Верстка веб-страниц с помощью сеточной системы



Модульная сетка — это система организации объектов в макете, основанная на колонках, рядах и отступах между ними. По направляющим сетки дизайнер выравнивает все элементы сайта: текстовые блоки, картинки, кнопки, фактоиды.

В данном видеоуроке мы познакомимся с интересным и универсальным подходом в верстке веб-страниц – сеточной системой из 12-ти колонок.

## Урок 12 Mobile first & Desktop first

Mobile first, в отличие от Desktop first это стратегия разработки сайта, согласно которой изначально сайт создается под мобильные устройства, с последующей адаптацией под планшеты и ПК с помощью медиа-запросов.

## Урок 13 Семантические теги

Семантические теги — это теги, которые предназначены для того, чтобы компьютерные программы (поисковые системы, сборщики информации, речевые браузеры и т.д.) понимали, какой тип информации заложен в данных тегах.

### Основные семантические теги:

#### <article>

Значение: независимая, отделяемая смысловая единица, например комментарий, твит, статья, виджет ВК и так далее.

Особенности: желателен заголовок внутри.

Типовые ошибки: путают с тегами <section> и <div>.

#### <section>

Значение: смысловой раздел документа. Неотделяемый, в отличие от <article>.

Особенности: желателен заголовок внутри.

Типовые ошибки: путают с тегами <article> и <div>.

#### <aside>

Значение: побочный, косвенный для страницы контент.

Особенности: может иметь свой заголовок. Может встречаться несколько раз на странице.

Типовые ошибки: считать <aside> тегом для «боковой панели» и размечать этим тегом основной контент, который связан с окружающими его элементами.

#### <nav>

Значение: навигационный раздел со ссылками на другие страницы или другие части страниц.

Особенности: используется для основной навигации, а не для всех групп ссылок. Основной является навигация или нет — на усмотрение верстальщика. Например, меню в подвале сайта можно не оборачивать в <nav>. В подвале обычно появляется краткий список ссылок (например, ссылка на главную, копия и условия) — это не является





основной навигацией, семантически для такой информации предназначен `<footer>` сам по себе.

Типовые ошибки: многие считают, что в `<nav>` может быть только список навигационных ссылок, но согласно спецификации там может быть навигация в любой форме.

#### `<header>`

Значение: вводная часть смыслового раздела или всего сайта, обычно содержит подсказки и навигацию. Чаще всего повторяется на всех страницах сайта.

Особенности: этих элементов может быть несколько на странице.

Типовые ошибки: использовать только как шапку сайта.

#### `<main>`

Значение: основное, не повторяющееся на других страницах, содержание страницы.

Особенности: должен быть один на странице, исходя из определения.

Типовые ошибки: включать в этот тег то, что повторяется на других страницах (навигацию, копирайты и так далее).

#### `<footer>`

Значение: заключительная часть смыслового раздела или всего сайта, обычно содержит информацию об авторах, список литературы, копирайт и так далее. Чаще всего повторяется на всех страницах сайта.

Особенности: этих элементов может быть несколько на странице.  
Тег `<footer>` не обязан находиться в конце раздела.

Типовые ошибки: использовать только как подвал сайта.