

Привет! Сегодня с вами познакомимся с HTML и CSS. С его помощью мы сможем создать собственный сайт. Также, ознакомимся с основными тегами, научимся оформлять простую страницу и добавлять текст и картинки на наш сайт. А с помощью CSS мы сможем изменить дизайн сайта.

## Введение HTML

**HTML** — это язык гипертекстовой разметки, с помощью которого создают структуру веб-страниц. Он позволяет пользователю создавать и структурировать разделы, параграфы, заголовки, ссылки и блоки для веб-страниц и приложений.

HTML — это специальный код для браузеров.

В нем с помощью тегов указывается, что показать на странице:

- Текст
- Картинки
- Таблицы
- Видеоролики
- Музыку

**HTML-теги** — это ключевые слова или символы, заключенные в угловые скобки.

Теги бывают двух видов: парные и одиночные (их также называют пустыми). Парные теги состоят из открывающего и закрывающего тега. Символ "/" после угловой скобки указывает на то, что тег закрывающий. Одиночные теги состоят только из открывающего тега.

Теги не чувствительны к регистру, поэтому могут быть написаны как прописными, так и строчными буквами: `<P>` означает то же самое, что и `<p>`.

### Тег `<!DOCTYPE html>`

Элемент `<!DOCTYPE html>` предназначен для указания типа текущего документа. Это необходимо, чтобы браузер понимал, как следует использовать текущую веб-страницу, поскольку HTML существует в нескольких версиях, кроме того, имеется XHTML (EXtensible HyperText Markup Language), похожий на HTML, но различающийся с ним по синтаксису.

### Тег `<html>...</html>`

Необходимо сообщить браузеру, что вы будете "говорить" с ним на языке HTML, это делается тэгом `<html>`.

Прежде чем напечатать что-либо, нужно поставить тэг "`<html>`" в первой строке файла.

Этот элемент — контейнер, который заключает в себе все содержимое веб-страницы, включая элементы `head` и `body`

### Тег `<head>...</head>`

Следующее, что необходимо, это "head" (голова или "шапка"). Этот элемент предназначен для хранения других элементов, цель которых — помочь браузеру в работе над

преобразованием веб-страницы. Также внутри могут находиться метатеги, которые хранят информацию для поисковых систем (ключевые слова, описание). Элемент не отображается на веб-странице.

## Тег <body>...</body>

"body" (тело) – предназначен для хранения содержания веб-страницы, отображаемого в окне браузера.

### Пример:

Ваш документ теперь должен выглядеть так:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

## Кодировка

В разделе <head> обязательно указываем кодировку **<meta charset=utf-8">**.

Данная запись указывает браузеру кодировку, в которой была написана данная страница - формат документа и раскладку клавиатуры, в данном случае это кириллица для Windows. Без этого ваша страница в браузере может отображаться непонятными символами.

## Кодировка. Основы viewport

Типичный сайт, оптимизированный для мобильных устройств, содержит следующий мета-тег:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

## Тег <title> ... </title>

Если вы хотите дать название документу, чтобы оно появлялось в верхней строке браузера, это нужно сделать в разделе "head", используя тег <title>

### Пример:

```
<title>Моя первая web-страница</title>
```

## Заголовки <h1> ... </h1>

Тэги <h1>, <h2>, <h3>, <h4>, <h5> и <h6> указывают браузеру создавать заголовки (h для "heading")

<h1> это заголовок первого уровня - самый крупный шрифт.

<h2> - заголовок второго уровня - шрифт меньшего размера.

<h6> - заголовок шестого уровня - самого низкого в этой иерархии, и самый маленький шрифт.

## Тег <p> ... </p>

Это параграф. Используется для отображения основного текста на веб-странице.

Чтобы на странице выводилось "Ура! Это моя первая страница." в разделе body напечатайте следующее:

```
<p>Ура! Это моя первая страница. </p>
```

### Пример:

Ваш HTML-документ должен теперь иметь такой вид:

```
<html>

<head>
<title>Моя первая web-страница</title>
</head>

<body>
<h2> Это заголовок</h2>
<p>Ура! Это моя первая страница. </p>
</body>

</html>
```

### Жирный шрифт

Тэг <b> информирует браузер, что весь текст между <b> и </b> должен быть напечатан жирным шрифтом

- "b" это сокращение для "bold"

### Курсив

Тэг <i> информирует браузер, что весь текст между <i> и </i> должен быть напечатан курсивом

- "i" это сокращение для "italic"

### Подчеркивание

Тэг <u> информирует браузер, что весь текст между <u> и </u> должен быть подчеркнутым

- "u" это сокращение от слова "underlined"

### Новая строка

Есть теги, которые являются одновременно открывающими и закрывающими.

Эти тэги содержат команды, которые не связаны с конкретными буквами, являются изолированными командами

Пример - тег <br>, который создает перевод строки

### Выравнивание по центру

Тег <center> выравнивает текст по центру.

Пример:

```
<center>Текст</center>
```

### Изображения

Изображения очень важны для повышения привлекательности сайта. С их помощью можно наметить пользователю о том, какую информацию он сможет получить на этой странице. Вы можете вставить любое изображение в веб-страницу с помощью тега <img>.

Тег <img> является пустым тегом, так как он не имеет закрывающего тега.

Как и любой другой тег он имеет свои атрибуты.

С помощью атрибута "src" - от английского "source" (источник), можно указать адрес картинки, которую Вы хотите поставить на сайт.

Для того чтобы теперь вставить её в нашу страничку к ней нужно указать путь.

Делается это так:

```

```

Где foto.jpg это название Вашей фотографии в данном случае, так как фото лежит рядом с html документом, путь к ней мы не указываем.

## Горизонтальная линия

Тег <hr> рисует горизонтальную линию. Тег <hr> относится к блочным элементам, линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

## Добавление комментариев

Тег <!-- --> добавляет комментарий в код документа.

Текст комментария не отображается на странице. Разрешается внутри комментария добавлять другие теги.

Пример:

```
<!-- текст -->
```

## Введение CSS

CSS расшифровывается как «каскадные таблицы стилей». Этот язык отвечает за внешний вид HTML-страницы. Синтаксис языка достаточно прост: он состоит из селекторов и свойств. С помощью селекторов можно сказать браузеру какие именно элементы мы хотим оформить. Свойства описывают как именно мы хотим оформить эти элементы. CSS файл имеет расширение .css

Подключаем файл style.css к нашей странице

```
<head>
  <meta charset="UTF-8">
  <title>Sample document</title>
  <link rel="stylesheet" href="style.css">
</head>
```

В CSS существует огромное количество свойств. Их можно разбить на следующие группы:

- оформление текста;
- работа с размерами и отступами;
- позиционирование элементов;
- создание сеток;
- декоративные: цвета, фон, тени;
- другие.

## CSS-правила

Весь CSS-код состоит из повторяющихся блоков следующего вида:

```
селектор {
свойство1: значение;
свойство2: значение;
}
```

## Простейшие селекторы

**Простейшие селекторы** — это селекторы по именам тегов. С их помощью можно задать стили для всех абзацев на странице, для всех ссылок, заголовков первого уровня и так далее. Такие селекторы содержат имя тега без символов `<` `>`.

**Например:**

```
p{
  свойство: значение;
}
h1{
  свойство: значение;
}
```

## Селектор по class

**Класс** — это один из атрибутов тегов.

Выглядит он вот так: `<p class="first"></p>`

В CSS существует возможность выбирать элементы по классу. Делается это с помощью такого селектора: `.имя_класса`.

**Например:**

```
.first {
  свойство: значение;
}
```

Имена классов могут состоять из латинских символов, цифр и знаков `-` и `_`. Имя класса должно начинаться с латинской буквы.

## Селектор по id

Атрибут `id` (идентификатор), а селектор записывается с помощью символа `#`, например, `#some-id`

На значение `id` распространяются те же ограничения, что и на имя класса. Также `id` должен быть уникальным на странице.

**Например:**

```
#first {
  свойство: значение;
}
```

## Цвет фона и фоновые изображения

### Цвет фона background-color

Свойство устанавливает цветную заливку для любого HTML-элемента. Для блочных элементов цвет фона распространяется на всю ширину и высоту блока элемента, для строчных — только на область их содержимого. Не наследуется.

**Например:**

```
body {
  background-color: red;
}
#container {
```

```
background-color: #ccc;  
}
```

## Фоновое изображение **background-image**

Свойство устанавливает изображение в качестве фона для элемента.

**Например:**

```
body {  
background-image: url(flower.png);  
}
```

## Повтор фоновых изображений **background-repeat**

Свойство определяет, каким образом будет повторяться фоновый рисунок. Не наследуется.

<b>repeat</b>	Весь фон страницы будет заполнен фоновым рисунком. Если при этом задать <code>backgroundposition</code> , то повтор будет осуществляться с указанной позиции. Значение по умолчанию.
<b>no-repeat</b>	Фоновое изображение не будет повторяться.
<b>repeat-x</b>	Фоновый рисунок повторяется от левого до правого края веб-страницы по верхнему краю страницы.
<b>repeat-y</b>	Фоновый рисунок повторяется от верхнего до нижнего края веб-страницы по левому краю страницы.

## Позиционирование фоновых изображений **background-position**

Свойство управляет точным расположением фонового изображения. Можно определить начальную позицию фонового изображения в виде горизонтальной и вертикальной координат посредством ключевых слов, точных абсолютных и процентных значений.

## Размер изображения **background-size**

Свойство позволяет масштабировать фоновое изображение по вертикали и горизонтали (`background-image`). Оно описывает, как изображение будет растягиваться и обрезаться, чтобы полностью закрыть собой фоновую область.

<b>cover</b>	Масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.
<b>contain</b>	Масштабирует изображение с сохранением пропорций таким образом, чтобы оно целиком поместилось внутри блока.

## Форматирование шрифта с помощью CSS свойств

Семейство шрифтов **font-family**.

Свойство используется для выбора начертания шрифта. Поскольку невозможно предсказать, установлен тот или иной шрифт на компьютере посетителя вашего сайта, рекомендуется прописывать все возможные варианты однотипных шрифтов. В таком случае браузер будет проверять их наличие, последовательно перебирая предложенные варианты.