

# Введение в HTML

# Зачем нужен HTML?

**HTML (Hypertext Markup Language)** - придуман для создания структуры страницы и структурирования ее контента.

**ВАЖНО!**

Это не язык программирования!

# HTML - теги

`<имя_тега>` — тег.

Часто используемые теги:

`<html>` `<body>` `<div>` `<h1>` `<nav>` `<ul>` `<ol>` `<li>` `<p>` `<a>`  
`<strong>` `<img>` `<video>` `<audio>` `<form>` `<input>`  
`<button>` `<label>`

# Парные теги

**<p>** <!-- открывающий тег -->

**Содержимое тега**

**</p>** <!-- закрывающий тег -->

# Одиночные теги

`<br>` `<!-- перенос строки -->`

`` `<!-- картинка -->`

Альтернативная запись одиночного тега.

`<br />` - с закрывающим слешем перед закрывающей скобкой. В стандарте HTML5 использование закрывающего слеша в одиночных тегах необязательно.

# Атрибуты тегов

`<p class="text-red">Моя собака кушает сосиски!</p>`

`<div id="logo">Логотип</div>`

Тег может иметь несколько атрибутов. Некоторые теги имеют обязательные атрибуты.

Атрибуты содержат дополнительную информацию об элементе, которая не будет отображаться на странице.

# Атрибут должен всегда иметь следующее:

`<имя-тега атрибут-1="значение-1" атрибут-2="значение-2">`

- Пробел между ним и именем элемента (или предыдущим атрибутом, если элемент уже имеет один или несколько атрибутов).
- Имя атрибута, за которым следует знак равенства.
- Значение атрибута, заключенное в кавычки.

## На заметку:

Простые значения атрибутов, которые не содержат пробелов или символов " ' ` = < > могут быть без кавычек. Но рекомендуется указывать все значения атрибутов в кавычки. Это делает код более читаемым и понятным.

# Вложенные теги

Вы также можете помещать элементы в другие элементы - это называется **вложенностью**.

```
<!-- Хорошо! -->
```

```
<p>Моя собака <strong>часто</strong> ест сосиски!</p>
```

```
<!-- Плохо! -->
```

```
<p>Моя собака <strong>часто есть сосиски!</p></strong>
```



# Категории тегов по типу содержания

**Metadata content** - метаданные для браузеров, поисковиков и т. д. теги которые находятся в `<head></head>`.

**Flowcontent** - потоковый контент, теги которые находятся в `<body></body>`.

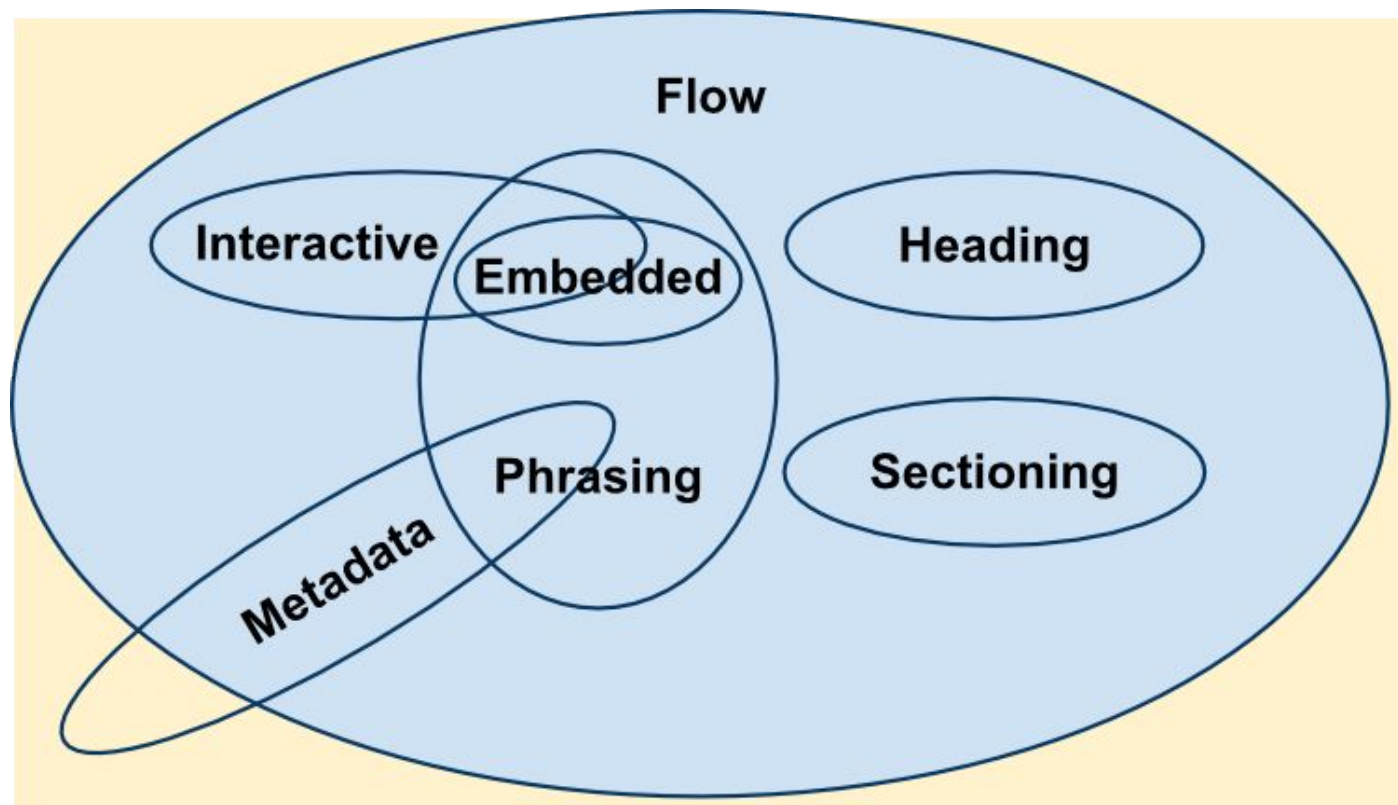
**Sectioning content** - крупные смысловые разделы документа.

**Heading content** - заголовки.

**Phrasing content** - фразовый контент, сам текст документа и мелкие текстовые элементы, которые мельче абзаца.

**Embedded content** - встраиваемый контент (изображения, видео, аудио и т. д.).

**Interactive content** - интерактивные элементы, то, с чем взаимодействует пользователь.



# С чего начинается HTML страница

Каждый HTML-документ должен начинаться с декларации типа документа или «доктайпа».

Для текущей версии HTML:

```
<!DOCTYPE html>
```

Для старой версии HTML 4.01 доктайп задается так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

# Базовая структура страницы

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Заголовок страницы</title>
```

```
  </head>
```

```
  <body>
```

```
    <!-- Контент. -->
```

```
  </body>
```

```
</html>
```

# Более сложная структура страницы

```
<!DOCTYPE html>
<html lang="ru">
  <head>
    <meta charset="utf-8">
    <meta name="description" content="Описание страницы" >
    <meta name="keywords" content="Ключевые слова" >
    <title>Заголовок страницы</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <p>Здесь текст, который будет выведен на экран.</p>
    <script src="scripts.js"></script>
  </body>
</html>
```

# Валидность кода

Markup Validation Service

<http://validator.w3.org/nu/>

Валидация зависит от того, какой указан тип документа **<!DOCTYPE>**

Частые ошибки:

- Не правильная вложенность тегов.
- Незакрытые теги.
- Отсутствуют обязательные атрибуты.
- Блочные элементы внутри строчных и т.д.

# Блочные элементы

К блочным элементам относятся теги: `<p>`, `<h1>`, `<div>`, `<ul>` и так далее.

Особенности блочных тегов:

- Блочные элементы всегда переносятся на новую строку.
- Блочным элементам можно задавать ширину, высоту, внутренние и внешние отступы.
- По умолчанию ширина равна ширине окна.

Блочный тег который может использоваться чаще всего это `<div>` представляет собой прямоугольник. Может создавать структуру и сетки на странице.

# Строчные элементы

К строчным элементам относятся теги: `<a>`, `<span>`, `<strong>`, `<em>` и так далее.

Особенности строчных тегов:

- Строчные элементы всегда идут друг за другом в одной строке.
- Строчным элементам можно задавать только отступы слева и справа.
- Ширина и высота равна содержимому элемента изменить с помощью стилей нельзя.

Основная задача строчных элементов это оформление текста.



# Смысловые разделы

Тег `<article></article>`. Представляет самостоятельную, независимую часть документа, страницы, например комментарий, статья, виджет погоды и т.д. Желателен заголовок внутри.

Тег `<section></section>`. Смысловой раздел документа. В отличие от `<article>` его нельзя отделить от страницы. Как правило, но не всегда, разделы имеют заголовок

Тег `<nav></nav>` определяет отдельную секцию документа, назначение которой обозначение ссылок навигации на другие страницы или другие части страниц.

Тег `<aside></aside>` представляет собой часть документа, чье содержимое только косвенно связано с основным содержимым документа. Чаще всего представлен в виде боковой панели, сносок или меток.

# Теги для работы с текстом

## Тег абзаца

Тег `<p></p>`. По умолчанию абзацы начинаются с новой строки и имеют вертикальные отступы, которыми можно управлять с помощью стилей.

## Теги выделения полужирным

Тег `<strong></strong>` делает текст полужирным и определяет **важность** отмеченного текста.

Тег `<b></b>` предназначен для выделения текста полужирным без придания ему особой важности.

## Теги выделения курсивом

Тег `<em></em>` выделяет текст курсивом, на который необходимо сделать *акцент*.

Тег `<i></i>` выделяет текст курсивом не делает акцент на важности.

# Теги для работы с текстом

## Теги перенос строки и разделитель

Тег `<hr>` используется для того, чтобы создать горизонтальную линию-разделитель. Линии можно задать стили через css.

Тег `<br>` переносит текст на новую строку. Можно использовать при оформлении блока контактов. Плохая практика использовать для разметки абзацев.

## Теги индекса

Теги используются для написания единиц измерений и простых формул.

Тег `<sup></sup>` отображает текст в виде верхнего индекса.

Тег `<sub></sub>` отображает текст в виде нижнего индекса.

# Теги для работы с текстом

## Тег преформатированный текст

Браузер удаляет пробелы и переносы строк если не использовать для этого теги или специальные символы.

Тег `<pre>/pre>` оставляет преформатированный текст без изменений.

## Тег вставки кода

Тег `<code>/code>` необходим для вставки кода в текст. И если вы пишете технические статьи или документацию.

# Теги для работы с текстом

## Теги цитаты

Тег `<blockquote></blockquote>` тег выделяет цитату как отдельный блок текста с отступами. Используется для длинных цитат, которые могут состоять из нескольких абзацев.

Тег `<q></q>` используется для коротких цитат в предложениях. Текст внутри этого тега автоматически обрамляется кавычками.

Тег `<cite></cite>` используется для того, чтобы выделить источник цитаты, название произведения или автора цитаты.

# Теги для работы с текстом

## Тег аббревиатуры

Тег `<abbr title="Расшифровка"></abbr>` указывает, что последовательность символов является аббревиатурой. С помощью атрибута `title` дается расшифровка сокращения. При наведении на текст появляется подсказка.

## Тег контактной информации

Тег `<address></address>` используется для написания контактной информации. Внутри можно использовать другие теги разметки текста. По умолчанию текст внутри выделен курсивом.

## Тег выделения текста

Тег `<mark></mark>` обычный текст выделяет визуально. Фоновый цвет тега желтый.

# Теги для работы с текстом

## Заголовки и подзаголовки

В html для выделения заголовков используется ряд тегов от `<h1>` до `<h6>` Порядок идет от самого главного до менее значимого заголовка. В реальной работе очень редко используются заголовки ниже третьего уровня.

`<h1>`Заголовок страницы 1 уровня`</h1>`

`<h2>`Заголовок страницы 2 уровня`</h2>`

`<h3>`Заголовок страницы 3 уровня`</h3>`

### НА ЗАМЕТКУ!

На странице сайта рекомендуется размещать как минимум один заголовок первого уровня также стоит правильно использовать заголовки на сайте по причине особого внимания к ним со стороны поисковых систем.

# Теги для работы с текстом

## Неупорядоченный или маркированный список

Список данного типа создается с помощью тега `<ul></ul>` и может содержать в себе элементы списка `<li></li>`.

```
<ul>
  <li>Молоко</li>
  <li>Хлеб</li>
</ul>
```

Также у тега `<ul>` есть атрибут `type` который задает вид маркера.

```
<ul type="disc">...</ul>
```

```
<ul type="circle">...</ul>
```

```
<ul type="square">...</ul>
```



# Теги для работы с текстом

## Упорядоченный или нумерованный список

Список данного типа создается с помощью тега `<ol></ol>` и может содержать в себе элементы списка `<li></li>`.

```
<ol>  
  <li>Первый пункт</li>  
  <li>Второй пункт</li>  
</ol>
```

У тега `<ol>` есть атрибут `start` задающий число с которого начинается нумерация.

```
<ol start="5">...</ol>
```

# Теги для работы с текстом

## Многоуровневый список

Списки можно вкладывать друг в друга и уровень вложенности не ограничен. В многоуровневом списке можно использовать как упорядоченные, так и неупорядоченные списки.

```
<ul>  
  <li>Мото</li>  
  <li>Автомобили  
    <ul>  
      <li>Легковые</li>  
      <li>Грузовые</li>  
    </ul>  
  </li>  
  <li>Водный транспорт</li>  
</ul>
```

# Теги для работы с текстом

## Список определений

Для создания списка определений необходимо использовать три тега

`<dl>...</dl>` - открывает и закрывает список

`<dt>...</dt>` - содержит название термина

`<dd>...</dd>` - содержит определение термина

Например:

`<dl>`

`<dt>Сингулярность</dt>`

`<dd>Этот термин довольно многолик</dd>`

`<dt>Нейронная сеть</dt>`

`<dd>Совокупность нейронов головного и спинного мозга</dd>`

`</dl>`

# Тег ссылка

Для добавления ссылки на странице используют тег `<a>...</a>`

`<a href="http://google.com">Найди меня!</a>`

У ссылки есть обязательный атрибут `href` в котором задается адрес ссылки.

Виды адресов:

1. Абсолютные.
2. Относительные.
3. Указывающие на страницу.
4. На файл.
5. Изображение.
6. Содержащие якорь.

# Тег ссылка

## Дополнительные атрибуты

Атрибут **title** - добавляет ссылке подсказку которая показывается при наведении на ссылку.

```
<a href="http://google.com" title="Поисковая система" >Найди меня!</a>
```

Атрибут **target** - управляет открытием документа или страницы. Чаще всего используется со значением **\_blank** если открывать страницу нужно в новом окне.

```
<a href="http://google.com" target="_blank" >Найди меня!</a>
```

Менее популярные значения атрибута: **\_self**, **\_parent**, **\_top**

# Тег изображение

Для вставки изображения на страницу необходимо использовать тег `<img>` с атрибутом `src` в котором нужно указать адрес картинки.

```

```

## Дополнительные атрибуты изображения

`alt` - альтернативный текст для изображения

`width` - ширина изображения. Может быть px и процентах

`height` - высота изображения. Может быть px и процентах

```

```

# Теги таблиц

Для создания таблицы необходимо использовать три тега

`<table>...</table>` - открывает и закрывает таблицу

`<tr>...</tr>` - добавляет строку в таблицу

`<td>...</td>` - добавляет ячейку в строку таблицы

В каждой строке должно быть одинаковое количество ячеек.

Например:

```
<table>
  <tr>
    <td>Ячейка 1</td>
    <td>Ячейка 2</td>
  </tr>
</table>
```

# Теги таблиц

В таблице мы можем добавлять заголовки столбцам или строкам. Для обозначения заголовка используется тег `<th>` он располагается внутри `<tr>` и аналогичен `<td>`. Текст внутри `<th>` выравнивается **по центру** и показывается **жирным**.

```
<table>
  <tr>
    <th>Магазин</th>
    <th>Количество товара</th>
  </tr>
  <tr>
    <td>Пятерочка</td>
    <td>345</td>
  </tr>
</table>
```



# Теги таблиц

В таблицах также можно задавать заголовки используя тег `<caption>...</caption>`. Заголовок отображается **сверху** таблицы и выравнивается **по центру**. Позицию и выравнивание можно изменить стилями.

```
<table>
  <caption>Учет по магазинам</caption>
  <tr>
    <th>Магазин</th>
    <th>Количество товара</th>
  </tr>
  <tr>
    <td>Пятерочка</td>
    <td>345</td>
  </tr>
</table>
```

# Теги таблиц

Кроме всего мы можем объединять строки и столбцы у таблиц. Это делается с помощью добавления атрибутов тегам `<th>` и `<td>`.

`colspan` - объединяет столбцы

`rowspan` - объединяет строки

```
<table>
  <tr>
    <td colspan="2">Ячейка на два столбца</td>
  </tr>
  <tr>
    <td>Ячейка на 1 столбец</td>
    <td>Ячейка на 1 столбец</td>
  </tr>
</table>
```

# Теги формы

Для добавления формы на страницу используется тег `<form>...</form>` у этого тега есть два важных атрибута `action` и `method`.

`action` - задает адрес куда отправлять данные формы.

`method` - задает метод передачи данных. Обычно это `get` или `post`. По умолчанию `get`.

```
<form action="https://itproger.com/login/" method="post">
```

```
  <!-- Поля формы. -->
```

```
</form>
```

# Теги формы

Большая часть полей в форме задаются с помощью тега `<input>` у тега есть два обязательных атрибута `type` и `name`.

`type` - задает тип поля и влияет на то как поле отображается в браузере. По умолчанию тип `text`.

`name` - необходим для того чтобы считать информацию с поля на сервере.

Дополнительные атрибуты

`id` - уникальный идентификатор поля. Используется для связи поля и подписи также для удобного выбора поля в коде JavaScript.

`value` - в этом атрибуте хранится значение поля.

# Теги формы

Для создания подписи к полю существует тег `<label></label>` используется с атрибутом `for` который связывает подпись и поле.

```
<form action="https://itproger.com/login/" method="post">  
  <label> Имя <input type="text" name="user-name"></label>  
</form>
```

```
<form action="https://itproger.com/login/" method="post">  
  <label for="user-name-id">Имя</label>  
  <input id="user-name-id" type="text" name="user-name">  
</form>
```

# Теги формы

Значения атрибута **type** у тега **<input>**

**password** - значение поля с данным атрибутом будет отображаться звездочками

```
<input type="password" name="user-password">
```

**checkbox** - создает поле галочку.

```
<input type="checkbox" name="user-policy" checked>
```

**radio** - переключатель. Тип поля для выбора варианта. Обычно это группа полей которая связана именем. У поля переключатель всегда должен быть атрибут **value**

```
<label> Есть машина <input type="radio" name="user-car" value="yes" checked></label>
```

```
<label> Нет машины <input type="radio" name="user-car" value="no"></label>
```

# Теги формы

**file** - добавляет поле для загрузки файлов. При вставке поля в форму требует обязательно наличие у формы атрибута **enctype** со значением **multipart/form-data**.

```
<input type="file" name="avatar">
```

**hidden** - делает поле скрытым от пользователя. При этом данные с этого поля передаются на сервер.

```
<input type="hidden" name="token">
```

**button** - создает кнопку для отправки формы. В обычных случаях не требует атрибута **name**.

```
<input type="button" value="Отправить!">
```

Можно заменить тегом кнопки.

```
<button>Отправить!</button>
```

# Теги формы

## Раскрывающийся список или селект

Селект `<select>...</select>` так же как и переключатель используется для выбора варианта. Чтобы задать варианты нужно внутрь вставить теги `<option>...</option>`.

```
<select name="color">
  <option value="red">Красный</option>
  <option value="green">Зеленый</option>
</select>
```

На сервер отправляется значение в атрибуте `value` если такой не задан, то отправится текст в `<option>`.



# Теги формы

## Селект с множественным выбором

В отличии от радиокнопок в селектах можно выбирать несколько вариантов из списка. Для этого тегу нужно добавить атрибут **multiple**.

```
<select name="cars[]" multiple>  
  <option value="lada">Lada</option>  
  <option value="bmw" selected>BMW</option>  
  <option value="lexus">Lexus</option>  
</select>
```

# Теги формы

## Многострочное поле ввода

Для передачи большого объема текста используют поле `<textarea></textarea>`. Кроме атрибутов `name` и `id` у этого тега есть свои специфические `rows` и `cols`. Атрибут `value` отсутствует.

`rows` - задает высоту поля в строках

`cols` - задает ширину поля в символах

```
<textarea name="comment" rows="100" cols="100">
```

Очень много текста с добрыми отзывами!

```
</textarea>
```