

HTML



HTML – HyperText Markup Language

Стандартизированный язык разметки документов во всемирной паутине.

HTML интерпретируется браузерами, полученный результат отображается на экране.

HTML был создан в 1991 г. Тимом Бернерсом Ли (сотрудником Европейского института физики частиц (CERN) в Женеве).

HTML - теги

Теги – это структурные и семантические элементы. Вставляются в текст html документа. Теги задают структуру документа.

Примеры тегов:

`<html> <body> <div> <table>`

`<p> <form> <audio> <video>`

`<article> <section> <nav>`

Синтаксис тегов

- Тег начинается с (`<`) и заканчивается с (`>`).
- Теги регистронезависимы, `<html>` значит то же, что и `<HTML>`. Но принято писать теги в **нижнем регистре**.
- Теги могут быть парными и одиночными.
- Теги могут быть вложены друг в друга.
- Теги могут иметь атрибуты.

Парные теги

`<h1>`Заголовок`</h1>`

`<p>` ← открывающий тэг

Этот тэг обозначает параграф.

`</p>` ← закрывающий тэг

Одиночные теги

`
` `<!-- перенос строки -->`

`<hr>` `<!-- разделитель -->`

`` `<!-- изображение -->`

`<input type="email">` `<!-- поле ввода email -->`

`<link rel="stylesheet" href="style.css">`

Вложенные теги

`<p>`Обычный текст и `<i>`курсивный текст`</i><p>`

```
<div>  
    <p>Параграф 1</p>  
    <p>Параграф 2</p>  
</div>
```

Не правильно

`<p>`Обычный текст и `<i></p>` курсивный текст`</i>`

Атрибуты тегов

- Атрибутов может быть несколько, а может и не быть вовсе.
- У некоторых тегов есть обязательные атрибуты.

```
<div class="content">
```

```
<script src="app.js"></script>
```

```
<input type="text" name="username">
```


Структура HTML документа

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <!-- Content -->
</body>
</html>
```

Теги верхнего уровня

<!DOCTYPE> - предназначен для указания типа текущего документа, поскольку HTML существует в нескольких версиях.

<html> - контейнер, содержащий всю страницу.

<head> - хранит в себе заголовок страницы и мета теги. Тут могут подключаться стили и скрипты.

<body> - хранит в себе контент, который будет отображен в окне браузера.

DOCTYPE	Описание
HTML 4.01	
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>	Строгий синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	Переходный синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></code>	В HTML-документе применяются фреймы.
HTML 5	
<code><!DOCTYPE html></code>	Для всех документов.

Теги заголовка документа

<title> - используется для отображения строки текста в левом верхнем углу окна браузера, а также на вкладке.

<meta> - хранит информацию, предназначенную для браузеров и поисковиков. Поисковые системы обращаются к метатегам для получения описания страницы, ключевых слов и других данных.

<head>

<title>Заголовок страницы**</title>**

<meta charset="UTF-8">

<meta name="desctiption" content="Мой сайт">

</head>

Кодировки

Первым стандартом для кодирования символов был **ASCII**.

Он содержал в себе 127 разных символов: числа, латинские буквы и некоторые спец символы.

UTF-8 (юникод) – еще одна из стандартизированных кодировок текста. Содержит в себе символы практически всех письменных языков, русский в том числе. Так же является кодировкой по умолчанию для HTML5.

```
<meta charset="UTF-8">
```

Заголовки

<h1>Заголовок уровня 1**<h1>**

<h2>Заголовок уровня 2**<h2>**

<h3>Заголовок уровня 3**<h3>**

<h4>Заголовок уровня 4**<h4>**

<h5>Заголовок уровня 5**<h5>**

<h6>Заголовок уровня 6**<h6>**

Параграф

`<p>`Этот текст начнется с абзаца`<p>`

`<p>`Новый абзац`<p>`

Форматированный текст

Жирный шрифт

Курсивный шрифт

Шрифт в виде нижнего индекса

Шрифт в виде верхнего индекса

Теорема Пифагора

А ² + В ² = С ²

Важность и акцент

`` - предназначен для акцентирования текста.

`` - предназначен для выделения важного текста.

Визуально эти элементы имеют сходный результат с `<i>` и ``, поэтому часто возникает заблуждение, что теги эквивалентны.

Но они не эквивалентны. В речевом браузере текст в тэгах `` и `` будет выделен по сравнению с остальным текстом.

Гиперссылки

```
<a href="filename">Текст ссылки</a>
```

`filename` — имя файла или адрес страницы в интернете

```
<a href="https://www.google.com">Google</a>
```

```
<a href="index.html">Главная</a>
```

```
<a href="photo/myphoto.html">Мой фотоальбом</a>
```

Составные части гиперссылки

<http://www.domen.ru/test-papka/index.htm>

http://	-протокол доступа
www.domen.ru	-доменное имя сервера
/test-papka/index.htm	-путь к файлу и имя файла web-страницы

Адрес ссылки может быть абсолютным или относительным.

Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка.

```
<a href="https://www.google.com">Google</a>
```

Относительные адреса, построены относительно текущего документа или корня сайта.

```
<a href="photo/myphoto.html">Мой фотоальбом</a>
```

Атрибут target

Открыть в новой вкладке

```
<a href="page.html" target="_blank">Page</a>
```

Открыть в текущем окне (поведение по умолчанию)

```
<a href="page.html" target="_self">Page</a>
```

Открыть в указанном окне

```
<a href="page.html" target="имя_окна">Page</a>
```

Картинки

```

```

Непарный тег, который имеет 2 обязательных атрибута

src – указывает путь к картинке

alt – альтернативный текст для картинки

Путь может быть, как абсолютный, так и относительный.

Пути к файлам

```

```

Картинка в текущей папке

```

```

Картинка в папке images, которая в текущей папке

```

```

Путь относительно корня сайта

```

```

Картинка на уровень выше, относительно текущей папки

Списки

Нумерованный список

1. First item
2. Second item
3. Third item
4. Fourth item

Маркированный список

- Item
- Item
- Item
- Item

Списки

Нумерованный список

```
<ol>  
  <li>Кофе</li>  
  <li>Чай</li>  
  <li>Молоко</li>  
</ol>
```

Маркированный список

```
<ul>  
  <li>Кофе</li>  
  <li>Чай</li>  
  <li>Молоко</li>  
</ul>
```

Таблицы

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Таблица задается при помощи тега `<table>`

`<tr>` - создает строку в таблице

`<td>` - создает ячейку в таблице

`<th>` - создает заголовочную ячейку в таблице

`<td>` должен находиться внутри `<tr>`

`<tr>` находится непосредственно внутри `<table>`

`<table><tr><td> ... </td></tr></table>`

Все нижеперечисленные элементы могут размещаться только внутри контейнера `<table>`

`<caption>` - предназначен для создания заголовка к таблице.

`<tbody>` `<tfoot>` `<thead>`

Группируют строки вверху таблицы, внизу таблицы и в теле таблицы.

Нужны для семантического разделения контента таблицы.

```
<table>
  <caption>Table of users</caption>
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Age</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
    </tr>
  </tbody>
</table>
```

Комментарии

<!-- это HTML комментарий -->

<!-- этот текст не будет выведен на экран -->

<!-- Параграф с важной информацией -->

<p>Только этот параграф будет отображен</p>

<!-- Нужно сюда добавить больше информации -->

<!-- <p>Этот текст не увидит пользователь</p> -->

Зарезервированные спец. символы

Символ копирайта ©

`©`

или

`©`

<iframe>

Тег позволяет загружать в область заданных размеров любые другие независимые документы.

Например, если мы хотим вставить **youtube** видео на сайте, то сервис дает нам готовый html код для этого, а там **<iframe>**

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/kpNbUN2_ebo"  
frameborder="0" allowfullscreen></iframe>
```


Формы

Формы – одни из важнейших элементов любого сайта.

Формы нужны для обмена данными между клиентом и сервером.

HTML формы содержат в себе элементы форм.

Элементы форм могут быть разных типов:

текстовый поля, переключатели, флажки, кнопки и много других

Форма создается при помощи тега `<form>`.

Внутри могут быть любые элементы, включая элементы форм.

```
<form action="handler.php">  
    <input name="fieldname" type="text">  
    <input type="submit">  
</form>
```

Почти все формы содержат в себе:

- адрес программы на веб сервере для обработки данных формы
- элементы формы, куда пользователь вводит данные
- кнопку для отправки формы

Атрибут `method`

```
<form action="handler.php" method="POST"></form>
```

Атрибут задает HTTP метод, которым будет отправлена форма.

Два наиболее используемых метода:

- GET
- POST

Если атрибут не задан, то данные отправляются методом GET.

GET и POST

GET

Данные формы становятся частью URL строки при GET запросе
`/test/demo_form.php?name1=value1&name2=value2`

POST

Данные формы отправляются в теле сообщения POST запроса

`POST /test/demo_form.php HTTP/1.1`

`Host: example.com`

`name1=value1&name2=value2`

Элемент `<input>`

Вид и свойства тега в браузере зависит от значения в атрибуте **type**.
Существует 23 типа учитывая нововведения стандарта HTML5.

Часто используемые типы:

```
<input type="text">
```

```
<input type="email">
```

```
<input type="password">
```

```
<input type="checkbox">
```

```
<input type="radio">
```

Типы input

button	Кнопка.	<input type="button" value="Кнопка"/>
checkbox	Флажки. Позволяют выбрать более одного варианта из предложенных.	<input type="checkbox"/> Пиво <input type="checkbox"/> Чай <input type="checkbox"/> Кофе
file	Поле для ввода имени файла, который пересылается на сервер.	<input type="file" value="Выберите файл"/>
hidden	Скрытое поле. Оно никак не отображается на веб-странице.	
image	Поле с изображением. При нажатии на рисунок данные формы отправляются на сервер.	<input type="image" value="Отправить"/>

Типы input

password	Обычное текстовое поле, но отличается от него тем, что все символы показываются звездочками. Предназначено для того, чтобы никто не подглядел вводимый пароль.	<input type="password"/>
radio	Переключатели. Используются, когда следует выбрать один вариант из нескольких предложенных.	<input type="radio"/> Пиво <input type="radio"/> Чай <input type="radio"/> Кофе
reset	Кнопка для возвращения данных формы в первоначальное значение.	<input type="reset" value="Сбросить"/>
submit	Кнопка для отправки данных формы на сервер.	<input type="submit" value="Отправить"/>
text	Текстовое поле. Предназначено для ввода символов с помощью клавиатуры.	<input type="text"/>

Типы input добавленные в HTML5

color	Виджет для выбора цвета.
date	Поле для выбора календарной даты.
datetime	Указание даты и времени.
datetime-local	Указание местной даты и времени.
email	Для адресов электронной почты.
number	Ввод чисел.
range	Ползунок для выбора чисел в указанном диапазоне.

Типы input добавленные в HTML5

search	Поле для поиска.
tel	Для телефонных номеров.
time	Для времени.
url	Для веб-адресов.
month	Выбор месяца.
week	Выбор недели.

Элемент `<select>`

Создает выпадающий список

```
<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="audi">Audi</option>  
  <option value="fiat">Fiat</option>  
  <option value="citroen">Citroen</option>  
</select>
```

`<option>` - задает вариант, который можно будет выбрать

Атрибуты select и option

```
<select name="cars" multiple>
```

...

```
</select>
```

Позволяет одновременно выбирать несколько элементов списка

```
<option value="audi" selected>Audi</option>
```

Заранее устанавливает данный пункт списка выделенным

Элемент textarea

Создает область, куда можно вводить несколько строк текста.

В отличие от `<input>` можно делать переносы строк.

```
<textarea rows="10" cols="45">
```

```
    Lorem ipsum.
```

```
</textarea>
```

Дополнительные атрибуты

У элементы форм и у самих форм есть еще ряд дополнительных атрибутов. Очень много атрибутов было добавлено в HTML5.

Атрибуты форм

<https://webref.ru/html/form>

Атрибуты input

<https://webref.ru/html/input>

HTML теги

HTML тегов много. Мы разобрали основные и наиболее используемые, к ним также относятся: `<div>` и ``.

`<div>` - является универсальным блочным элементом и предназначен для группирования элементов документа.

`` - универсальный строчный элемент для выделения отдельных строк, символов или других строчных элементов для дальнейшего их оформления с помощью стилей.

Блочные и строчные элементы

То, как браузер отобразит элемент, зависит от типа элемента.

Есть два типа HTML элементов:

- блочные (block-level elements)
- строчные (inline elements)

Блочные элементы

Блочные элементы отображаются с новой строки и занимают всю доступную ширину окна браузера.

Примеры блочных элементов:

- `<div>`
- `<h1>` – `<h6>`
- `<p>`
- `<form>`

Строчные элементы

Строчные элементы не отображаются с новой строки.
Занимаемое место определяется количеством контента.

Примеры строчных элементов:

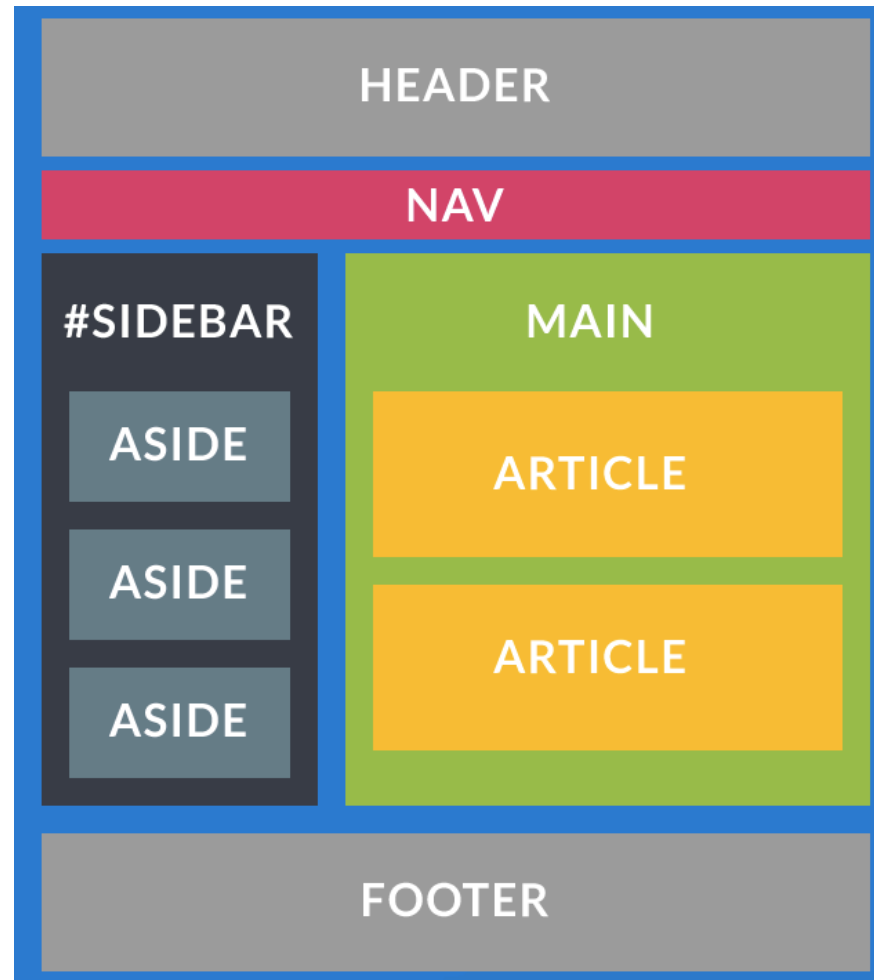
- ``
- `<a>`
- ``
- `<i>`, ``, ``, ``

HTML 5

- Новый «доктайп» `<!DOCTYPE html>`
- Новые семантические элементы
- Много новых атрибутов у элементов форм
- Новые элементы для работы с графикой и мультимедиа
- Ряд новых API (геолокация, локальное хранилище, веб воркеры, кэширование веб приложения, drag&drop)
- Убраны некоторые элементы из стандарта (```<center>` и др.)

Семантика HTML5

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



Графика HTML5

<canvas> - создаёт область, в которой при помощи JavaScript можно рисовать разные объекты. Это API по работе с холстом. Можно создавать рисунки, анимацию, игры и др.

<svg> - это контейнер для svg графики. SVG (Scalable Vector Graphics). Формат описания векторной графики средствами XML. При использовании в HTML5 SVG является частью DOM и операции доступные над DOM доступны над SVG. Отлично подходит для создания сложных интерфейсов, простых анимаций и т.д.

Сравнение canvas и svg

Canvas

- Зависит от разрешения
- Нет обработки событий
- Слабые возможности для вывода текста
- Можно сохранить результат как .png или .jpg
- Отлично подходит для игр

SVG

- Не зависит от разрешения
- Обработка событий
- Отлично для приложений с большой областью рендеринга
- Медленный рендеринг сложных фигур, так как использует DOM

Пример canvas

```
<canvas id="myCanvas" width="200" height="100"  
style="border:1px solid #000000;">  
</canvas>
```

```
1 var c = document.getElementById("myCanvas");  
2 var ctx = c.getContext("2d");  
3 ctx.moveTo(0,0);  
4 ctx.lineTo(200,100);  
5 ctx.stroke();
```



Пример svg

```
<svg width="400" height="180">  
  <rect x="50" y="20" rx="20" ry="20"  
    width="150" height="150"  
    style="fill:red;stroke:black;stroke-  
width:5;opacity:0.5"/>
```

Your browser does not support inline SVG.

```
</svg>
```



Медиа элемент `audio`

```
<audio controls >
```

```
    <source src="music.mp3" type="audio/mp3"/>
```

```
    <b>Audio is not supported in your browser</b>
```

```
</audio>
```

Атрибуты `<audio>`

`autoplay` - звук начинает играть сразу после загрузки страницы.

`controls` - добавляет панель управления к аудиофайлу.

`loop` - повторяет воспроизведение звука после его завершения.

Медиа элемент `video`

```
<video loop>
  <source src="video.mp4" type="video/mp4"/>
  <b>Video is not supported in your browser</b>
</video>
```

Атрибуты `<video>`

poster - картинка, которая будет отображаться, пока видео не доступно или не воспроизводится.

preload - для загрузки видео вместе с загрузкой веб-страницы.

+ атрибуты элемента `<audio>`

Проверить поддержку элементов можно на
<http://caniuse.com/>