

HTML

HTML



HTML (HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Начиная с 2 версии (HTML.2.0)

Практически все страницы содержат описание разметки на языке HTML(XHTML). Язык HTML интерпретируется браузерами (процесс интерпретации - парсинг); В результате интерпретации и выводится содержимое страницы на экраны устройства.

HTML-документ: ТЕГ

HTML - документ состоит из древа элементов и текста. Сами элементы это теги. Все теги помещаются в угловые скобки `< tag >`

Начальный **тег** показывает где будет начинаться элемент открывающий (начальный) и **закрывающий** (конечный), где перед именем тега ставится слеш / : `< tag >` содержимое `< /tag >` . Между тегами содержится содержимое тега, **контент**

DOCTYPE

Синтаксис с указанием, как браузеру правильно интерпретировать документ

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"
<html>
  <head>
    <title>Моя первая веб-страница</title>
  </head>

  <body>
    <h1>Заголовок страницы</h1>
    <p>Основной текст.</p>
  </body>
</html>
```

```
<!DOCTYPE [Элемент верхнего уровня] [Публичность]
"[Регистрация]//[Организация]//[Тип] [Имя]//[Язык]" "[URL]">
```

HTML 5

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

  <body>
    <h1>Заголовок страницы</h1>
    <p>Основной текст.</p>
  </body>
</html>
```

Структура

```
<!DOCTYPE html>
  <html lang="ru">
    <head>
      <title>Document</title>
    </head>
    <body>
      <div></div>
    </body>
  </html>
```

`<html></html>` - **теги** определяющие начало и конец документа.

`<head></head>` - служебная секция. Здесь размещаются команды для **браузеров**, инструкции для поисковых роботов, ссылки на файлы, скрипты.

`<body></body>` - секция в которой выводит всю видимую часть **web-страницы**.

`<html>...</html>`

Является корневым элементом документа. Все остальные элементы содержатся внутри **тегов** `<html>...</html>`. Все, что находится за пределами **тегов**, не воспринимается браузером как код **HTML** и никак им не обрабатывается.

`<head>...</head>`

Тег помогает браузеру разобраться с данными которые включены в `html`- документ. Содержимое `head` не отображается в браузере напрямую, за исключение тега `title`.

- `<link>`
- `<meta>`
- `<script>`
- `<style>`
- `<title>`

`<head> . . . </head>`

Обязательным тегом раздела `<head>` является тег `<title>`. Текст, размещенный внутри этого тега, отображается в строке заголовка веб-браузера. Оптимальная длина Title — от 30 до 65 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.

<meta>

Не обязательным тегом является тег <meta> С его помощью можно задавать различные мета-описание, ключевые слова для поисковых систем. Можно записывать <head> несколько <meta> тегов.
синтаксис:

```
<meta name="description" content="Описание страницы">  
<meta name="keywords" content="Ключевые слова через запятую">
```

Один тег <meta>, содержит в себе только 1 вид атрибута.
Так например в <meta> нельзя поставить атрибут

В интернет-магазине Nice Price вы можете купить айфон 7 по приятным ценам в рассрочку или кредит. Бесплатная доставка по Смоленску и сервисное...

```
16 <meta name="description" content="В интернет-магазине Nice Price вы  
можете купить айфон 7 по приятным ценам в рассрочку или кредит. Бесплатная доставка  
по Смоленску и сервисное обслуживание. Телефон 8 (4812) 56-44-56." />  
17 <meta name="keywords" content="iphone 7, айфон 7" />
```

<meta>

- description — описание веб-страницы,
- keywords — ключевые слова,
- http-equiv — тип отправляемого документа и кодировка (charset), **charset="utf-8"**
- author — автор,
- copyright — авторские права,
- robots — правила индексирования страницы для роботов,
- viewport — данные о настройке области просмотра

* не рекомендуется использовать кодировку ASCII, поскольку они представляют угрозу безопасности.

<meta> - viewport

* <meta name="viewport" content="width=device-width, initial-scale=1.0">

width	положительный целочисленный номер или литерал device-width	Определяет ширину области просмотра в пикселях
height	положительный целочисленный номер или литерал device-height	Определяет высоту области просмотра в пикселях
initial-scale	положительное число между 0.0 и 10.0	Определяет соотношение между шириной устройства и размером области просмотра
maximum-scale	положительное число между 0.0 и 10.0	Определяет максимальное значение зума;
minimum-scale	положительное число между 0.0 и 10.0	Определяет минимальное значение зума
user-scalable	булевское значение (да или нет)	Если весь набор содержит значения нет, то пользователю не доступен зум на веб-странице. По умолчанию задано значение да.

<body>

В теге <body> размещено все видимое содержимое документа

<body>

...

...

...

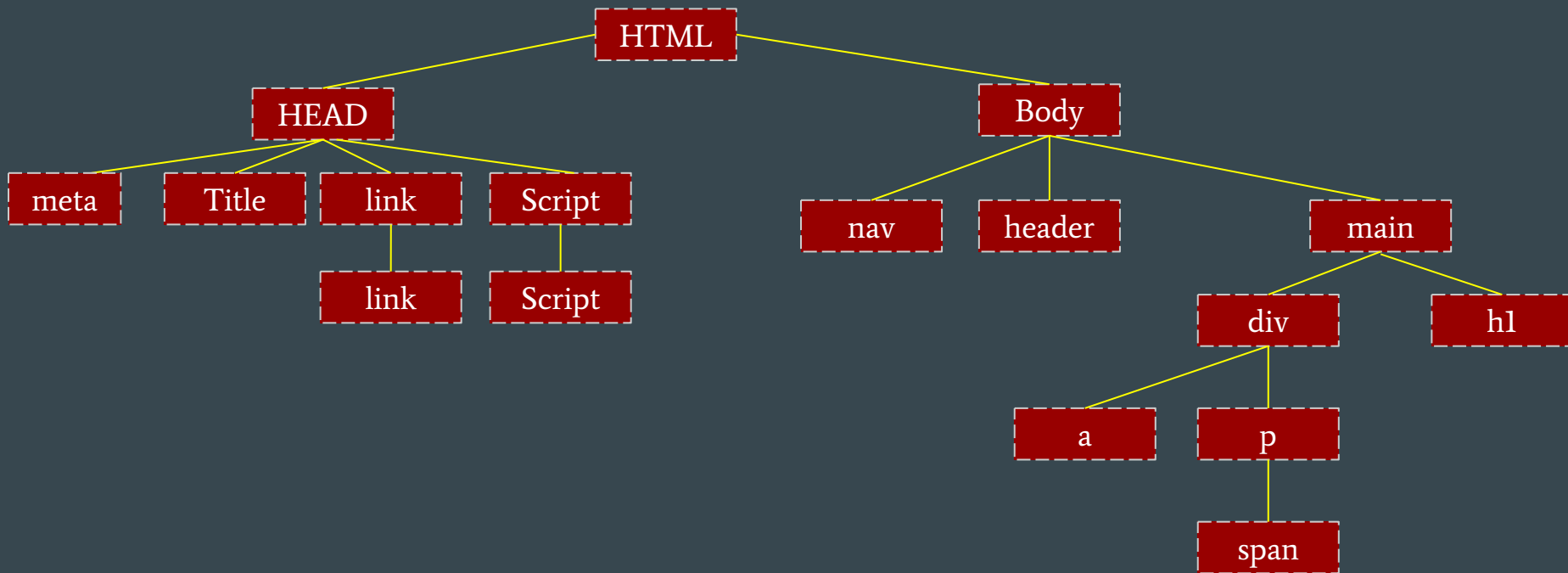
</body>

Вложенность или индентация

```
<body>
  <nav>
    <ul>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
    </ul>
  </nav>
</body>
```

Индентация (стиль отступов) - правила форматирования исходного кода.

Структура



DOM Иерархия

HTML-документ — это как большое семейное древо, с родителями, братьями, детьми, предками и потомками.

Всё это происходит из возможности вкладывать одни элементы HTML внутрь других.

`<p>Случайносгенерированныйтекст, для показания<q>тегов</q>этого урока</p>`

DOM Иерархия

Предок - Элемент, который содержит в себе другие элементы. `<html>` является предком для всех элементов внутри него. В тоже время `<body>` является предком для всех тегов которые выводят содержимое.

Потомок - элемент, расположенный внутри одного или более типов элементов

```
<div> <p> <span></span> </p> </div>
```

```
<div> </div> Предок
```

Родительский элемент — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На схеме `<html>` является родительским только для `<html>` и `<body>` Тег `<p>` является родительским только для ``.

DOM Иерархия

Дочерний элемент — элемент который находится по иерархии ниже предка.

`<div> <p> </p> </div>`

`<p> </p>` - Дочерний элемент `<div>`

Сестринский элемент — элемент, имеющий общий **родительский** элемент с рассматриваемым, так называемые элементы одного уровня. На схеме `<head>` и `<body>` — элементы одного уровня, так же как и элементы `<h1>`, `<h2>` и `<p>` являются между собой сестринскими.

ТЕГИ

Метаданные документа

```
<head> </head>
```

```
<title></title>
```

```
<link>
```

```
<meta>
```

```
<style> </style>
```

Секционные элементы

```
<article> </article>
```

```
<section> </section>
```

```
<nav> </nav>
```

```
<aside> </aside>
```

```
<header> </header>
```

```
<footer> </footer>
```

Заголовки

```
<h1> </h1>
```

```
<h2> - </h2>
```

```
<h3> - </h3>
```

```
<h4> - </h4>
```

```
<h5> - </h5>
```

```
<h6> - </h6>
```

ТЕГИ

Скрипты

`<script></script>`

`<canvas></canvas>`

Формы

`<form></form>`

`<label></label>`

`<input>`

`<textarea>`

Встроенное содержимое

`<picture> </picture>`

``

`<iframe> </iframe>`

`<object> </object>`

`<video> </video>`

`<audio> </audio>`

Семантика текста

`<a> `

` `

`<cite> </cite>`

`<ruby> </ruby>`

`<code> </code>`

`<i> </i>`

` `

`
`

Типы элементов

Теги верхнего уровня - все которые входят в

`<html></html>` - входят все теги

`<head> </head>` - мета данные

`<body></body>` - содержимое документа

Блочные элементы:

`<div>` - относится к универсальным блочным контейнерам и применяется в тех случаях, где нужны блочные элементы без дополнительных свойств.

`<h1>`, ..., `<h6>` - эта группа **тегов** определяет текстовые заголовки разного уровня, которые показывают относительную важность секции, расположенной после заголовка.

`<hr>` - Рисует горизонтальную линию, которая по своему виду зависит от используемых атрибутов. Линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

`<p>` - определяет параграф (абзац) текста.

Типы элементов - Строчные

Строчные элементы

Строчными называются такие **элементы веб-страницы**, которые являются непосредственной частью другого **элемента**, например, текстового абзаца. В основном они используются для изменения вида текста или его логического выделения.

Типы элементов - Строчные

`<a>` - является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов `name` или `href` тег `<a>` устанавливает ссылку или якорь.

`` - определяет жирное начертание шрифта.

`
` - устанавливает перевод строки в том месте, где этот тег находится.

`<i>` - Устанавливает курсивное начертание шрифта.

`` предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG

Типы элементов - Строчные

`` - предназначенный для определения строчного **элемента** внутри документа.

`<sub>` - отображает шрифт в виде нижнего индекса. Текст при этом располагается ниже базовой линии остальных символов строки и уменьшенного размера — H_2O .

`<sup>` - отображает шрифт в виде верхнего индекса. По своему действию похож на `<sub>`, но текст отображается выше базовой линии текста — m^2 .

Типы элементов

Разница между блочными и строчными элементами

Строчные элементы могут содержать только данные или другие **строчные элементы**, а в **блочные** допустимо вкладывать другие **блочные элементы**, **строчные элементы**, а также данные. Иными словами, **строчные элементы** никак не могут хранить **блочные элементы**.

Блочные элементы всегда начинаются с новой строки, а **строчные** таким способом не акцентируются.

Блочные элементы занимают всю доступную ширину, например, окна браузера, а ширина **строчных элементов** равна их содержимому плюс значения отступов, полей и границ.

Спецсимволы

Для отображения символов, которых нет на клавиатуре, применяются специальные знаки, начинающиеся с амперсанда (&) и заканчивающиеся точкой с запятой (;).

Например:

© - ©

™ - ™

¼ - ¼

√ - √

β - β

π - π

½ - ½

¶ - ¶

™ - ™

∫ - ∫

♠ ♠

♣ ♣

♥ ♥

♦ ♦

HTML - комментарии

Для того чтобы комментировать определенную часть кода или сделать пометку, используют тег `<!-- -->`

Комментарии в HTML не отображаются на странице в браузере пользователю, однако их можно увидеть в исходном коде веб-страницы

Практическое применение комментариев:

- Позволяет оставить подсказку, для упрощения поиска по коду
- Для временного отключения кода. Удалять код неудобно, поскольку он может еще пригодиться

Комментарии можно использовать в любом месте страницы, кроме тега `<title>` — внутри него они не работают.

Атрибуты

Чтобы расширить возможности отдельных **тегов** и более гибко управлять содержимым контейнеров и применяются **атрибуты тегов**.

```
<тег атрибут1="значение" атрибут2="значение">  
<тег атрибут1="значение" атрибут2="значение">...</тег>
```

Атрибуты без значений

Допустимо использовать некоторые **атрибуты** у **тегов**, не присваивая им никакого значения.

Порядок атрибутов в тегах

Порядок **атрибутов** в любом **теге** не имеет значения и на результат отображения **элемента** не влияет.

Гиперссылки

Гиперссылки создаются с помощью парного тега `<a>`. Внутри тега помещается текст, который будет отображаться на веб-странице.

ссылка состоит из 2 частей, указателя и адреса

```
<a href="адресная-часть">указатель ссылки</a>
```

Адресная часть ссылки состоит из URI (Uniform Resource Locator) — унифицированный адрес ресурса. При создании адресов для разделения слов между собой рекомендуется использовать дефис, а не символ подчеркивания.

Гиперссылки (атрибуты)

accesskey - активация ссылки с помощью комбинации клавиш.

accesskey="с"

download - предлагает скачать указанный по ссылке файл.

download=""

href - Задаёт адрес документа, на который следует перейти.

href="URL"

title - задаёт описание ссылки при наведении.

title="при наведении"

target - Задаёт возможность открытия ссылки в текущем окне или в новой вкладке

target="_blank" открытия в новой вкладке

target="_self" открытия в текущей вкладке(по умолчанию)

Гиперссылки (Относительный путь)

Относительный путь описывает путь к указанному документу относительно текущего. Путь определяется с учетом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте.

Путь для относительных ссылок имеет два специальных обозначения:

/ - указывает на корневую директорию и говорит о том, что нужно начать путь от корневого каталога документов и идти вниз до следующей папки.

.. / - подняться на одну папку (директорию) выше.

Гиперссылки (Якоря)

Якоря, или внутренние **ссылки**, создают переходы на различные разделы текущей **веб-страницы**, позволяя быстро перемещаться между разделами. Это оказывается очень удобным в случае, когда на странице слишком много текста. Внутренние **ссылки** также создаются при помощи **тега** `<a>` с разницей в том, что **атрибут** `href` содержит имя указателя — так называемый **якорь**, а не **URI-адрес**. Перед именем указателя всегда ставится знак `#`.

`Лето` - **якорь** с указанием `id` **элемента**.

`<p id="p1">...</p>` - **элемент** с соответствующим `id`.

Если нужно сделать **ссылку** с одной страницы сайта на определенный раздел другой страницы, то необходимо задать `id` для этого раздела страницы, а затем добавить его к абсолютному адресу **ссылки**:

`Лето`