**Вёрстка (HTML/CSS) - основы HTML**

<https://webdesign-master.ru/blog/html-css/4.html>

<https://html5book.ru/osnovy-html/>

<https://webref.ru/course/html-basics/syntax>

<https://html5book.ru/examples/html-tags.html> - таблица тегов

<https://validator.w3.org/> - Валидатор

<https://caniuse.com/> - Can I Use (проверка на кроссбраузерность)

**План:**

1. [Базовые принципы верстки](#30j0zll)
2. [Объектная модель документа (DOM](#1fob9te))
3. [Теги.](#3znysh7)

* [Блочные/строчные и парные/непарные теги](#3znysh7)
* [Структура тега. Атрибуты тегов](#_tyjcwt)
* [Основные группы тегов](#3dy6vkm)
* [Семантические элементы html5](#1t3h5sf)

1. [Структура HTML-документа](#17dp8vu)
2. [Основные группы тегов](#3dy6vkm)

* [Текст](https://html5book.ru/html-text/)
* [Списки](https://html5book.ru/html-lists/)
* [Изображения](https://html5book.ru/images-in-html/)
* [Таблицы](https://html5book.ru/html-table/)
* [Формы](https://html5book.ru/html5-forms/)
* [Ссылки](https://html5book.ru/hyperlinks-in-html/)

1. [Спецсимволы](https://html5book.ru/specsimvoly-html/)

# [HTML-комментарии](#_3whwml4)

 **HTML/CSS**

**H**yper **T**ext **M**arkup **L**anguage (**HTML**) — “язык гипертекстовой разметки” — стандартизированный язык разметки документов (веб-страниц). Это простой текст, содержащий много тегов, который образует веб-страницу. Правильное произношение - **Эйч Ти Эм Эль**

*HTML = структура и содержание*

Актуальная версия: HTML5

* **язык** означает, что он может быть прочитан как человеком, так и компьютером;
* **разметка** означает, что написанный вами код помечается с помощью ключевых слов;
* **гипертекст** означает, что он использует HTTP как часть Интернета.

**C**ascading **S**tyle **S**heets (**CSS**) — “каскадные таблицы стилей” — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

*SCC = визуальное оформление*

Актуальная версия: CSS 3.

**Вёрстка** — процесс создания HTML документа согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету.

Верстка должна быть:

* адаптивной;
* кроссбраузерной (поддержку свойств версиями браузеров можно проверить тут: <https://caniuse.com/>)
* семантичной ( <https://gsnedders.html5.org/outliner> )
* валидной ( <https://validator.w3.org/> )
* + простотой, гибкой, компонентной.

**Браузеры.** Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение .html), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса вебстраниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются:

 - Google Chrome

 - Mozilla Firefox

 - Opera

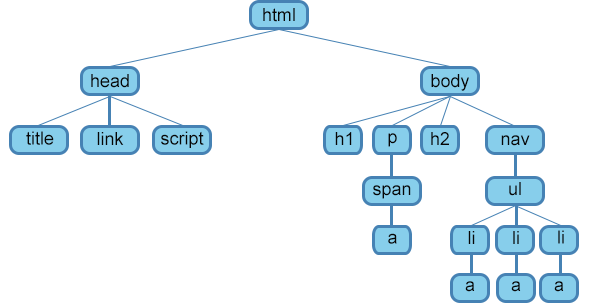
 - Internet Explorer

 - Safari

**Объектная модель документа (DOM)**

Элементы, находящиеся внутри тега <html>, образуют дерево документа, так называемую ***объектную модель документа, DOM*** (*document object model*). При этом элемент <html> является корневым элементом.

HTML-код, который вы пишете, анализируется и преобразуется браузером впоследствии в DOM.



Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые **«родственные отношения»** между элементами.

***Предок*** — элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <html>. В то же время элемент <body> является предком для всех содержащихся в нем тегов: <h1>, <p>, <span>, <nav> и т.д.

.

***Потомок*** — элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <html>.

***Родительский элемент/parent*** — это непосредственный предок (предок первого уровня) элемента.

**Дочерний элемент*/ребенок/child*** — непосредственный потомок

***Сестринский элемент*** — это группа из двух и более элементов, у которых общий родитель. Элементы не обязательно должны быть одного типа, просто у них должен быть общий родитель. Их еще называют элементы одного уровня.

[Подробнее тут](https://html5book.ru/osnovy-html/)

 **Теги**. **Блочные/строчные и парные/непарные теги**

**Tag (тег)** — элемент языка разметки гипертекста.

HTML тег - структурная единица разметки HTML документа. Код HTML складывается из кирпичиков, которые именуются тегами. Каждый тег имеет свою функцию, а изучение языка разметки HTML, в конечном счете, заключается именно в изучении тегов и их свойств в документе.

Существуют ***парные теги*** (включают открывающийся и закрывающийся теги) и ***одиночные***.

<p> Text </p> - парный  
 <br/> - одиночный

Теги могут вкладываться друг в друга. При вложении следует соблюдать порядок их закрытия (***принцип «матрёшки»***).

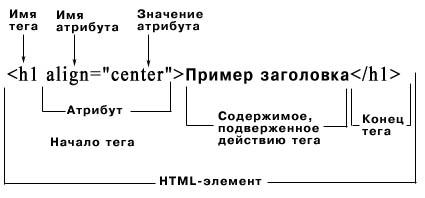
<p><i> Hello! </i></p> - верно  
 <p><i> Hello! </p></i> - неверно

Выделяют две основные категории HTML-элементов, которые соответствуют типам их содержимого и поведению в структуре веб-страницы — ***блочные и строчные*** элементы. С помощью блочных элементов можно создавать структуру веб-страницы, строчные элементы используются для форматирования текстовых фрагментов.

* примеры блочных: <p>, <div>, <form>, <nav>, <ul>, <h1>, ...
* примеры строчных: <a>, <span>, <b>, <em>, <i>, <mark>, …

[Подробнее тут](https://html5book.ru/html-tags/)   
[Таблица тегов](https://html5book.ru/examples/html-tags.html)

# :mortar_board: **Структура тега. Атрибуты тега.**



**Начальный тег** показывает, где начинается элемент, конечный — где заканчивается. **Закрывающий тег** образуется путем добавления слэша / перед именем тега: <имя тега>…</имя тега>. Между начальным и закрывающим тегами находится содержимое тега — контент.

Одиночные теги не могут хранить в себе содержимого напрямую, оно прописывается как значение атрибута, например, тег <input type="button" value="Кнопка"> создаст кнопку с текстом **Кнопка** внутри.

Универсальные атрибуты

* class - определяет имя класса, которое позволяет связать тег со стилевым оформлением;
* id - указывает имя стилевого идентификатора;
* style - применяется для определения стиля элемента с помощью правил CSS;
* title - описывает содержимое элемента в виде всплывающей подсказки.

Каждому элементу можно присвоить несколько значений class и только одно значение id. Множественные значения class записываются через пробел, <div class="nav top">. Значения class и id должны состоять только из букв, цифр, дефисов и нижних подчеркиваний и должны начинаться только с букв или цифр.

**Атрибуты** действуют как дополнительная информация, привязанная к элементу HTML. Они пишутся внутри тега HTML. Таким образом, они также не отображаются в браузере.

Есть [16 атрибутов HTML](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes), которые могут быть использованы в любом элементе HTML. Все они не являются обязательными. Вы в основном будете применять class (который используется для CSS) и title (подсказка, которая появляется при наведении курсора на объект, вроде этого).

Некоторые элементы HTML содержат **обязательные** атрибуты. Например, при вставке изображения вы должны указать его расположение с помощью атрибута src:

О **атрибутах** можно почитать [тут](https://html5book.ru/html-attributes/) или [тут](https://webref.ru/html/attr)

 **Основные группы тегов**

- [служебные теги](https://html5book.ru/osnovy-html/#part2) (doctype, head, title, ...);

- [теги для группировки контента](https://html5book.ru/html5-semantic-elements/) (div, header, main, footer, section, article, ...);

- [списки](https://html5book.ru/html-lists/) (ol, ul, dl);

- [ссылки](https://html5book.ru/hyperlinks-in-html/) (a);

- [таблицы](https://html5book.ru/html-table/) (table, thead, tbody, tr, th, td, ...);

- [текстовые](https://html5book.ru/html-text/) (h1, p, q, strong, ...);

- [изображения](https://html5book.ru/images-in-html/) (img, …);

- формы (form, input, select, button, ...).

[Полный список тегов](https://html5book.ru/examples/html-tags.html)

**Семантические элементы html5**

[**https://developer.mozilla.org/en-US/docs/Web/HTML/Element**](https://developer.mozilla.org/en-US/docs/Web/HTML/Element)

[**https://html5book.ru/html5-semantic-elements/**](https://html5book.ru/html5-semantic-elements/)

Целью тегов HTML является передача **смысла** документу. Не беспокойтесь о том, как ваша веб-страница выглядит. Сосредоточьтесь на значении каждого тега, который вы будете использовать.

В зависимости от написанного вами содержимого, вы можете выбрать подходящий элемент, соответствующий смыслу текста.

**1. Структурные элементы: организация страницы** Структурные элементы позволяют организовать основные части вашей страницы. Они обычно содержат другие элементы HTML.  
  
Вот что типичная веб-страница может в себя включать:  
  
 <header> в качестве первого элемента страницы, который может включать в себя логотип и слоган;  
 <nav> в качестве списка ссылок, которые ведут на разные страницы сайта; На странице может быть несколько элементов <nav>. Не заменяет теги <ul> или <оl>, он просто их обрамляет.

<nav class="main-nav">  
 <ul>  
 <li><a href="#">Item 01</a></li>  
 <li><a href="#">Item 02</a></li>  
 <li><a href="#">Item 03</a></li>  
 <li><a href="#">Item 04</a></li>  
 </ul>  
 </nav>

<h1> в качестве заголовка страницы;  
 <article> в качестве основного содержимого страницы, вроде статьи блога;  
 <footer> в качестве последнего элемента страницы, расположенного внизу.

<aside> - группирует содержимое, связанное с окружающим его контентом напрямую, но которое можно счесть отдельным (т.е., удаление этого блока не повлияет на понимание основного содержимого). Чаще всего элемент позиционируется как боковая колонка.

<section> - группирует тематическое содержимое и обычно содержит заголовок. Не является блоком-оберткой, для этих целей уместнее использовать элемент <div>. В качестве содержимого может выступать оглавление, разделы научных публикаций, программа мероприятия.

### <address> - Используется для определения контактной информации автора/владельца документа или статьи. В браузере обычно отображается курсивом.

<main> - представляет основное содержимое документа (содержимое элемента <body>). Элемент <main> не может быть потомком таких элементов как <article>, <aside>, <footer>, <header> или <nav>.

<form> - основу любой формы составляет элемент <form>...</form>. Он не предусматривает ввод данных, так как является контейнером, удерживая вместе все элементы управления формы – поля. Атрибуты этого элемента содержат информацию, общую для всех полей формы, поэтому в одну форму нужно включать поля, объединенные логически.

**2. Текстовые элементы: определение контента** Внутри структурных элементов вы обычно находите текстовые элементы, призванные определить цель вашего содержимого.  
  
Вы, в основном, будете использовать:  
  
 <p> для абзацев;  
 <ul> для (неупорядоченных) списков;  
 <ol> для (упорядоченных) списков;  
 <li> для отдельных пунктов списка;  
 <blockquote> для цитат.

**3. Строчные элементы: различный текст**  
 Поскольку текстовые элементы могут быть длинными, но с разным содержанием, строчные элементы позволяют различать части текста.  
  
Есть много строчных элементов, но вы обычно столкнётесь со следующими:  
  
 <strong> для важных слов;  
 <em> для выделенных слов;  
 <a> для ссылок;  
 <small> для менее важных слов;  
 <abbr> для аббревиатур, вроде W3C.  
Просто читая этот код HTML, вы можете легко понять, что означает каждый элемент HTML.

<mark> - текст, помещенный внутрь тега <mark>, выделяется по умолчанию желтым цветом (цвет фона и цвет шрифта в выделенном блоке можно изменить, задав определенные css-стили). С помощью данного тега можно отмечать важное содержимое, а также ключевые слова.

<time> - определяет время (24 часа) или дату с возможным указанием времени. Текст, заключенный в данный тег, не имеет стилевого оформления браузером.

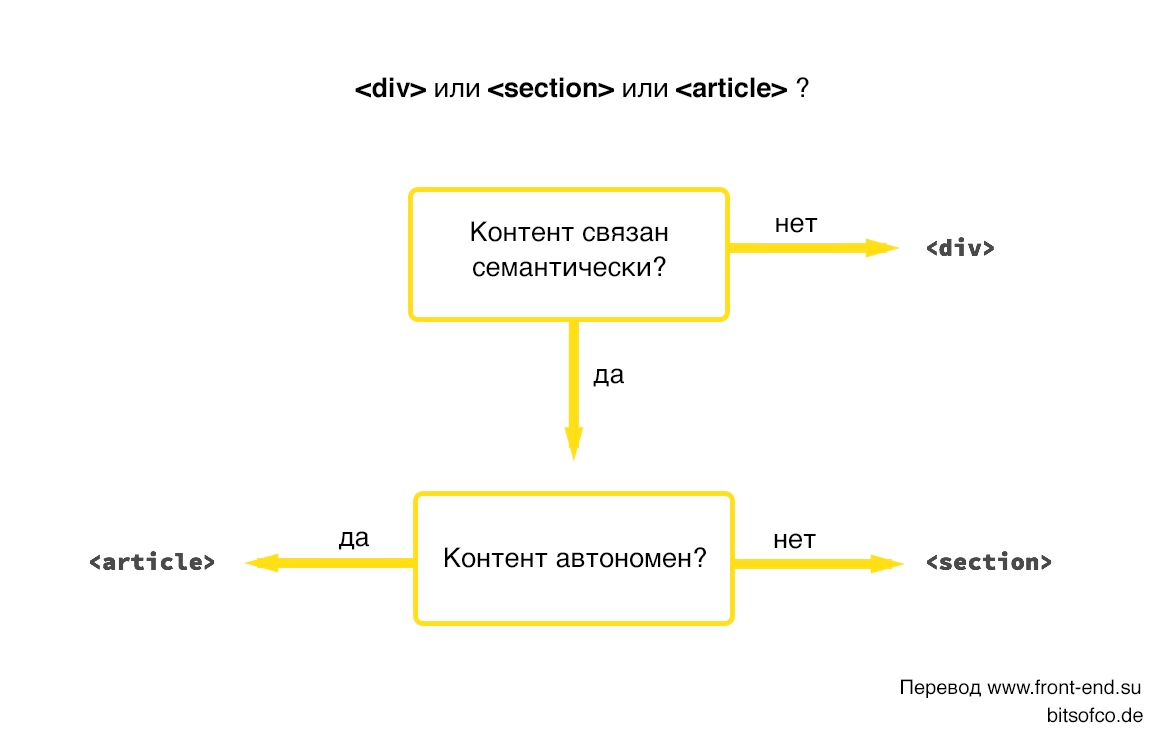
## **4.Общие элементы**

Когда ни один семантический элемент не подходит для вашего содержимого, но вы всё ещё хотите вставить элемент HTML (в целях группирования или стилизации), то можете остановиться на одном из двух **общих** элементов:

* <div> для блочных элементов;
* <span> для строчных элементов.

Хотя эти элементы HTML на самом деле не несут какого-либо *смысла*, они пригодятся когда мы начнём использовать CSS.

Хорошая статья по поводу выбора: <article>, <section> или <div> читаем [тут](http://front-end.su/2015/11/01/div-or-section-or-article/)



 **Структура HTML-документа**

#### 

#### **<!doctype html> - тип документа (доктайп)**

Строчка с **DOCTYPE *(Document Type Definition)*** - данная конструкция всегда указывается в начале документа для правильного "понимания" браузером того, какая версия HTML используется при построении документа.

Ввиду того, что HTML постоянно развивается, он имеет несколько версий, как и любой программный продукт. Текущая версия HTML - пятая и приведенный в примере доктайп является актуальным.

Главная цель: сделать так, чтобы ваш сайт правильно работал и одинаково ровно и красиво смотрелся во всех браузерах.

DTD представляет собой XML-документ, определяющий, какие теги, атрибуты и их значения действительны для конкретного типа HTML. В принципе, углубляться в изучение типов документа нет никакого смысла, ибо с выходом HTML5 данная конструкция стала стандартом. Просто вставляйте ее в начало документа каждый раз, когда начинаете верстать макет сайта.

<!-- Объявление формата/типа документа -->  
<!DOCTYPE html>   
<html>

<head>

<!-- Определяем кодировку символов документа -->

<meta charset="utf-8">

<!-- Устанавливаем параметры отображения страницы на мобильных устройствах -->  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
 <!-- Задаем заголовок документа -->   
 <title>Title</title>  
  
 <!-- Подключаем внешнюю таблицу стилей -->   
 <link media="all" rel="stylesheet" href="css/main.css">  
   
 <!-- Подключаем скрипты -->   
 <script src="js/main.js"></script>  
  
 </head>  
  
 <body>  
 <!-- Основная часть документа -->   
 </body>  
</html>

#### 

#### 

#### **<html> - начало документа**

<html>  
Первый тег, который мы встречаем после доктайпа, это <html>. Тег <html> является обязательным, так как содержит всю структуру документа и является оболочкой для остальных элементов.

#### **Тег <head>**

Тег <head> предназначен для хранения метаинформации HTML документа, то есть информации, которая не отображается в самом документе, но является важной и во многом определяет, как документ будет выглядеть и как себя вести.

Данный тег обязателен в документе.

#### **Тег <title> - заголовок документа**

<title>Заголовок</title>

Заголовок <title> является обязательным тегом, содержащим текстовую метаинформацию, которая отображается в заголовке браузера или вкладки. Тег <title> должен находиться в теге <head>. Также, содержимое данного тега используется поисковыми системами для отображения документа в результатах выдачи.

#### 

#### **Метатег <meta charset="UTF-8" >**

<meta charset="UTF-8" >

Метатег, содержащий информацию о кодировке документа. Очень желательно указывать данный тег во всех создаваемых документах для правильного отображения. Отсутствие данного тега может привести к тому, что вместо слов, на странице будут отображены неведомые символы и текст перестанет быть разборчивым и человекопонятным.

Метатег - специализированный тег, предназначенный для предоставления структурированных данных о странице. Метатеги чаще всего используются в теге <head>. Метатеги не являются обязательными в структуре HTMLдокумента.

Рекомендую во всех HTML документах изначально использовать кодировку UTF-8, как в примере выше.

#### **Фавикон (favicon)**

**<link rel="icon" href="favicon.ico" >**

Подключает к документу файл с изображением фавиконки. Фавиконка (favicon) - миниатюрный значок, отображаемый рядом с названием документа во вкладке браузера. Фавиконка - это графический файл, размером 16 x 16 (или 32 x 32) пикселей, который может иметь различные форматы, такие, как png, jpg, ico, gif. Традиционно используется формат ico. Анимированные фавиконки - это gif файлы, содержащие анимацию. Наблюдать анимированный фавикон можно, например, ВКонтакте, когда приходит новое сообщение.

#### **CSS стили документа**

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

Подключает к документу CSS файл со стилями оформления HTML. войство href конструкции <link > указывает расположение внешнего файла.

#### **Тег <script>**

Тег <script> содержит код или ссылку на файл javaScript и чаще всего используется внутри тега <head>, хотя [инструмент оптимизации скорости загрузки страниц](http://developers.google.com/speed/pagespeed/insights/) от Google, рекомендует данный тег использовать в конце документа, перед закрывающим тегом </body>.

# :mortar_board: **HTML-комментарии**

Комментарий в HTML-коде задается так:

<!-- любой текст -->

Текст внутри комментария не отображается браузером на странице. Комментарии обычно используются в следующих случаях:

* Для комментирования кода. Всегда полезно оставить подсказку.
* Для временного отключения кода. Удалять код неудобно, так как его надо будет восстанавливать, а закомментировать и потом раскомментировать — самое лучшее решение.

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

 **Понятие «layout»**

**Layout** - это шаблон(каркас) веб-страницы.

