

## Элемент <!DOCTYPE>

Элемент <!DOCTYPE> предназначен для указания типа текущего документа. Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях. Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать <!DOCTYPE>.

Существует несколько видов <!DOCTYPE>, они различаются в зависимости от версии языка, на которого ориентированы.

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

## Тег <html>

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>. Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует неизменное их использование. Как правило, тег <html> идет в документе вторым, после определения типа документа, устанавливаемого через элемент <!DOCTYPE>. Закрывающий тег <html> должен всегда стоять в документе последним.

## Тег <head>

Тег <head> предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Также внутри контейнера <head> находятся метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Содержимое тега <head> не отображается напрямую на веб-странице, за исключением тега <title> устанавливающего заголовок окна веб-страницы.

## Тег <body>

Элемент <body> предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <body>. К такой информации относятся текст, изображения, теги, скрипты JavaScript и т.д.

## **Тег <script>**

Тег <script> предназначен для описания скриптов, может содержать ссылку на программу или ее текст на определенном языке. Скрипты могут располагаться во внешнем файле и связываться с любым HTML-документом. Такой подход позволяет использовать одни и те же общие функции на многих веб-страницах и ускоряет их загрузку, т.к. внешний файл кэшируется при первой загрузке, и скрипт вызывается быстрее при последующих вызовах.

<script> может располагаться в заголовке или теле HTML-документа в неограниченном количестве. В большинстве случаев местоположение скрипта никак не сказывается на работу программы. Однако скрипты, которые должны выполняться в первую очередь, обычно помещают в заголовок документа.

## **Тег <style>**

Тег <style> применяется для определения стилей элементов веб-страницы. Тег <style> необходимо использовать внутри контейнера <head>. Можно задавать более чем один тег <style>.

## **Тег <link>**

Устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. В отличие от тега <a>, тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

## **Тег <title>**

Определяет заголовок документа. Элемент <title> не является частью документа и не показывается напрямую на веб-странице. Допускается использовать только один тег <title> на документ и размещать его в контейнере <head>.

## Тег <a>

Тег <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. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

## Тег <b>

Устанавливает жирное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

## Тег <br>

Тег <br> устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега абзаца <p>, использование тега <br> не добавляет пустой отступ перед строкой.

## Тег <div>

Элемент <div> является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут class или id с именем селектора.

## Тег <h1>

HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег <h1> представляет собой наиболее важный заголовок первого уровня, а тег <h6> служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги <h1>,...,<h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

## Тег <hr>

Рисует горизонтальную линию, которая по своему виду зависит от используемых параметров, а также браузера. Тег <hr> относится к блочным элементам, линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

## Тег <i>

Устанавливает курсивное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

## Тег <img>

Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут `src`. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег <img> в контейнер <a>.

## Тег <p>

Определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

## Тег <canvas>

Создает область, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства. При помощи тега <canvas> можно создавать рисунки, анимацию, игры и др.

## Тег <audio>

Добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег [<source>](#). Внутри контейнера <audio> можно написать текст, который будет выводиться в браузерах, не работающих с этим тегом.

## Тег <video>

Добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег <source>.

## **Тег <ol>**

Тег <ol> устанавливает нумерованный список. Каждый элемент списка должен начинаться с тега <li>. Если к тегу <ol> применяется таблица стилей, то элементы <li> наследуют эти свойства.

## **Тег <ul>**

Тег <ul> устанавливает маркированный список. Каждый элемент списка должен начинаться с тега <li>. Если к тегу <ul> применяется таблица стилей, то элементы <li> наследуют эти свойства.

## **Тег <li>**

Тег <li> определяет отдельный элемент списка. Внешний тег <ul> или <ol> устанавливает тип списка — маркированный или нумерованный.

## Тег <table>

Элемент <table> служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>. Внутри <table> допустимо использовать следующие элементы: <caption>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> и <tr>.

Таблицы с невидимой границей долгое время использовались для верстки веб-страниц, позволяя разделять документ на модульные блоки. Подобный способ применения таблиц нашел воплощение на многих сайтах, пока ему на смену не пришел более современный способ верстки с помощью слоев.

## Тег <tr>

Тег <tr> служит контейнером для создания строки таблицы. Каждая ячейка в пределах такой строки может задаваться с помощью тега <th> или <td>.

## Тег <td>

Предназначен для создания одной ячейки таблицы. Тег <td> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

## Тег <form>

Тег <form> устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

## Тег <input>

Тег <input> является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем. Главным образом <input> предназначен для создания текстовых полей, различных кнопок, переключателей и флажков. Хотя элемент <input> не требуется помещать внутрь контейнера <form>, определяющего форму, но если введенные пользователем данные должны быть отправлены на сервер, где их обрабатывает серверная программа, то указывать <form> обязательно.

Основной атрибут тега <input>, определяющий вид элемента — type. Он позволяет задавать следующие элементы формы: текстовое поле (text), поле с паролем (password), переключатель (radio), флажок (checkbox), скрытое поле (hidden), кнопка (button), кнопка для отправки формы (submit), кнопка для очистки формы (reset), поле для отправки файла (file) и кнопка с изображением (image). Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики. Кроме того, в HTML5 добавлено еще более десятка новых элементов.

## Тег <select>

Тег <select> позволяет создать элемент интерфейса в виде раскрывающегося списка, а также список с одним или множественным выбором, как показано далее. Конечный вид зависит от использования атрибута `size` тега <select>, который устанавливает высоту списка. Ширина списка определяется самым широким текстом, указанным в теге <option>, а также может изменяться с помощью стилей. Каждый пункт создается с помощью тега <option>, который должен быть вложен в контейнер <select>. Если планируется отправлять данные списка на сервер, то требуется поместить элемент <select> внутрь формы. Это также необходимо, когда к данным списка идет обращение через скрипты.

## Тег <option>

Тег <option> определяет отдельные пункты списка, создаваемого с помощью контейнера <select>. Ширина списка определяется самым широким текстом, указанным в теге <option>, а также может изменяться с помощью стилей. Если планируется отправлять данные списка на сервер, то требуется поместить элемент <select> внутрь формы. Это также необходимо, когда к данным списка идет обращение через скрипты.

## Тег <textarea>

Поле <textarea> представляет собой элемент формы для создания области, в которую можно вводить несколько строк текста. В отличие от тега <input> в текстовом поле допустимо делать переносы строк, они сохраняются при отправке данных на сервер.

Между тегам `<textarea>` и `</textarea>` можно поместить любой текст, который будет отображаться внутри поля.

Вся информация взята с сайта

<http://htmlbook.ru/html>