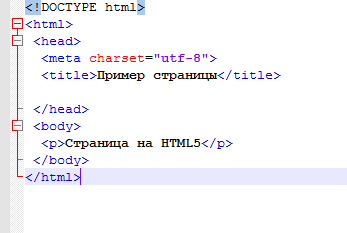
***Структура HTML-кода***

Если открыть любую веб-страницу, то она будет содержать в себе типичные элементы, которые не меняются от вида и направленности сайта. В примере  показан код простого документа, содержащего основные теги.



Скопируйте содержимое данного примера и сохраните его в папке под именем example.html. После этого запустите браузер и откройте файл через пункт меню Файл > Открыть файл (Ctrl+O). В диалоговом окне выбора документа укажите файл example.html. В браузере откроется веб-страница, показанная на рис.

Элемент <!DOCTYPE> предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, ведь HTML существует в нескольких версиях.

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

|  |  |
| --- | --- |
| HTML 5 | |
| <!DOCTYPE html> | В этой версии HTML только один доктайп. |

# Теги

Чтобы браузер при отображении документа понимал, что имеет дело не с простым текстом, а с элементом форматирования и применяются теги. Общий синтаксис написания тегов следующий.

**<тег атрибут1="значение" атрибут2="значение">**

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

Как видно из данного примера, теги бывают двух типов — одиночные и парные (контейнеры). Одиночный тег используется самостоятельно, а парный может включать внутри себя другие теги или текст. У тегов допустимы различные атрибуты, которые разделяются между собой пробелом. Впрочем, есть теги без всяких дополнительных атрибутов. Условно атрибуты можно подразделить на обязательные, они непременно должны присутствовать, и необязательные, их добавление зависит от цели применения тега.

В примере 1 показан типичный HTML-документ с тегами и текстом.

**Пример 1. Теги в документе**

<!DOCTYPE HTML >

**<html>**

**<head>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**<title>**Lorem ipsum**</title>**

**</head>**

**<body>**

**<p>Hello!**  **</p>**

**</body>**

**</html>**

В данном примере используется одиночный тег **<meta>**, а парных тегов сразу несколько: **<html>**, **<head>**, **<title>**, **<body>** и **<p>**.

# *Парные теги*

Парные теги, называемые по-другому контейнеры, состоят из двух частей — открывающий и закрывающий тег. Открывающий тег обозначается как и одиночный — **<тег>**, а в закрывающем используется слэш — **</тег>**. Допускается вкладывать в контейнер другие теги, однако следует соблюдать их порядок. Так, на рис.1 демонстрируется, как можно и нельзя добавлять один контейнер внутрь другого.

Рис. 3.1а

*а*

Рис. 3.1б

*б*

*Рис. 1. Вложение тегов, а — правильное, б — неверное*

Если связать открывающий и закрывающий тег между собой скобкой, как показано на рис. 1, то несколько скобок обозначающих разные контейнеры, не должны пересекаться между собой (рис. 1а). Любое пересечение условных скобок (рис. 1б) говорит о том, что правильная последовательность тегов нарушена.

Не все контейнеры требуют обязательно закрывающего тега, иногда его можно и опустить. Тем не менее, закрывайте все требуемые теги, так вы приучитесь сводить к нулю возможные ошибки.

# *Правила применения тегов*

Для тегов любого типа действуют определенные правила их использования. Причем, некоторые правила обязательны для выполнения, а другие являются рекомендациями, т.е. их можно выполнять, а можно и нет.

### Атрибуты тегов и кавычки

Согласно спецификации HTML все значения атрибутов тегов следует указывать в двойных ("пример") или одинарных кавычках ('пример'). Отсутствие кавычек не приведет к ошибкам, браузеры во многих случаях достаточно корректно обрабатывают код и без кавычек, за исключением текста, содержащего пробелы (пример 1).

**Пример 1. Использование кавычек в атрибутах тегов**

<!DOCTYPE HTML >

**<html>**

**<head>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**<title>**Кавычки в атрибуте alt**</title>**

**</head>**

**<body>**

**<p><img** src="images/arena.png" alt="Вид заголовка" width="400" height="101"**></p>**

**<p><img** src="images/arena.png" alt=**Вид заголовка** width="400" height="101"**></p>**

**</body>**

**</html>**

В данном примере строка 8 написана правильно, со всеми кавычками, а в строке 9 у атрибута alt кавычки отсутствуют. Из-за этого браузер в качестве значения alt возьмет только первое слово («Вид»), а слово «заголовка» будет воспринято как ошибочное значение. Поэтому всегда приучайтесь указывать значения атрибутов тегов в кавычках.

### Теги можно писать как прописными, так и строчными символами

Любые теги, а также их атрибуты нечувствительны к регистру, поэтому вы вольны выбирать сами, как писать — **<BR>**, **<Br>** или **<br>**. В любом случае рекомендуется придерживаться выбранной формы записи на протяжении всех страниц сайта. Заметим также, что текст, полностью набранный прописными символами, читается хуже, чем текст со строчными символами или смешанный.

### Переносы строк

Внутри тега между его атрибутами допустимо ставить перенос строк. В примере 2 показана одна и та же строка, но оформленная разными способами.

**Пример 2. Переносы строк в коде тега**

<!DOCTYPE HTML >

**<html>**

**<head>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**<title>**Кавычки в атрибуте alt**</title>**

**</head>**

**<body>**

**<p><img** src="images/arena.png" alt="Вид заголовка в IE" width="400" height="101"**></p>**

**<p><img** src="images/arena.png"

 alt="Вид заголовка в браузере IE"

 width="400"

 height="101"**></p>**

**</body>**

**</html>**

В данном примере первый тег **<img>** набран в одну строку, включая все его атрибуты, а второй тег **<img>** разбит на несколько строк.

Хотя ошибки при переносе текста в подобном случае и не возникнет, рекомендуем писать теги в одну строку, иначе ухудшается восприятие кода и его становится сложнее править.

### Неизвестные теги и атрибуты

Если какой-либо тег или его атрибут был написан неверно, то браузер проигнорирует подобный тег и будет отображать текст так, словно тега и не было. Опять же, следует избегать неизвестных тегов, поскольку код HTML не пройдет валидацию.

### Порядок тегов

Существует определенная иерархия вложенности тегов. Например, тег **<title>** должен находиться внутри контейнера **<head>** и нигде иначе. Чтобы не возникло ошибки, следите за тем, чтобы теги располагались в коде правильно.

Если теги между собой равноценны в иерархии связи, то их последовательность не имеет значения. Так, можно поменять местами теги **<title>** и **<meta>**, на конечном результате это никак не скажется.

### Закрывайте все теги

Существует три состояния закрывающего тега: обязателен, не требуется или не обязателен. Обязательный закрывающий тег должен присутствовать всегда, иначе это приведет к ошибке при отображении документа. Для некоторых тегов вроде **<br>**закрывающего тега нет в принципе. Необязательный закрывающий тег говорит о том, что разработчик может его как добавить, так и опустить, к ошибке это не приведет. Однако рекомендуем закрывать все подобные теги, включая необязательные, это дисциплинирует, создает более стройный и строгий код, который легко модифицировать.

# Атрибуты тегов

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

### Для атрибутов тегов используются значения по умолчанию

Когда для тега не добавлен какой-либо допустимый атрибут, это означает, что браузер в этом случае будет подставлять значение, заданное по умолчанию. Если вы ожидали получить иной результат на веб-странице, проверьте, возможно, следует явно указать значения некоторых атрибутов.

### Атрибуты без значений

Допустимо использовать некоторые атрибуты у тегов, не присваивая им никакого значения, как показано в примере 3.

**Пример 3. Атрибуты без значений**

<!DOCTYPE HTML >

**<html>**

**<head>**

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"**>**

**<title>**Добавление формы**</title>**

**</head>**

**<body>**

**<form** action="self.php"**>**

**<p><input** type="text"**></p>**

**<p><input** type="submit" disabled**></p>**

**</form>**

**</body>**

**</html>**

В данном примере используется атрибут disabled, у которого явно не задано значение. Подобная запись называется «сокращенный атрибут тега».

### Порядок атрибутов в тегах

Порядок атрибутов в любом теге не имеет значения и на результат отображения элемента не влияет. Поэтому теги вида <img src="/images/title.gif" width="438" height="118"> и <img height="118" width="438" src="/images/title.gif"> по своему действию равны.

### Формат атрибутов

Каждый атрибут тега относится к определенному типу (например: текст, число, путь к файлу и др.), который обязательно должен учитываться при написании атрибута. Так, в примере 3.3 упоминается тег **<img>**, он добавляет на веб-страницу рисунок, а его атрибут width задает ширину изображения в пикселах. Если поставить не число, а нечто другое, то значение будет проигнорировано и возникнет ошибка при валидации документа.

# Типы тегов

Каждый тег HTML принадлежит к определенной группе тегов, например, табличные теги направлены на формирование таблиц и не могут применяться для других целей.

Условно теги делятся на следующие типы:

* теги верхнего уровня;
* теги заголовка документа;
* блочные элементы;
* строчные элементы;
* универсальные элементы;
* списки;
* таблицы;
* фреймы.

# *Теги верхнего уровня*

Эти теги предназначены для формирования структуры веб-страницы и определяют раздел заголовка и тела документа.

### <html>

Тег **<html>** является контейнером, который заключает в себе всё содержимое веб-страницы, включая теги **<head>** и **<body>**. Открывающий и закрывающий теги **<html>** в документе необязательны, но хороший стиль диктует непременное их использование.

### <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.

### <body>

Тег **<body>** предназначен для хранения содержания веб-страницы, отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера **<body>**. К такой информации относится текст, изображения, таблицы, списки и др.

Набор тегов верхнего уровня и порядок их вложения показан в примере 1.

**Пример 1. Теги верхнего уровня**

**<html>**

**<head>**

...

**</head>**

**<body>**

...

**</body>**

**</html>**

В данном примере показано, что контейнер **<html>** определяет «каркас» всей веб-страницы, внутри него вначале задается тег **<head>**, затем идет контейнер **<body>**, в нем хранится содержательная часть документа, которая и отображается в браузере. Теги **<html>** и **<body>** хотя и не относятся к обязательным тегам (т. е. их можно не размещать в коде), все же стоит добавлять всегда. Это позволяет получить четкую и понятную структуру документа.

Заметьте, что в примере не упоминается **<!DOCTYPE>**, поскольку этот обязательный элемент кода веб-страницы не является тегом, а предназначен для браузеров, чтобы сообщить им, как интерпретировать текущий документ.

# *Теги заголовка документа*

К этим тегам относятся элементы, которые располагаются в контейнере **<head>**. Все эти теги напрямую не отображаются в окне браузера, за исключением тега **<title>**, который определяет название веб-страницы.

### <title>

Используется для отображения строки текста в левом верхнем углу окна браузера, а также на вкладке. Такая строка сообщает пользователю название сайта и другую информацию, которую добавляет разработчик.

### <meta>

Метатеги 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. Хотя тег **<meta>**всего один, он имеет несколько атрибутов, поэтому к нему и применяется множественное число.

Так, для краткого описания содержимого веб-страницы используется значение description атрибута name, как показано в примере 1.

**Пример 1. Использование description**

<!DOCTYPE HTML>

**<html>**

**<head>**

**<title>**HTML**</title>**

**<meta** name="description" content="Сайт об HTML и создании сайтов"**>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

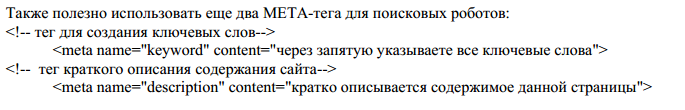
**</head>**

**<body>**

**<p>**...**</p>**

**</body>**

**</html>**



Описание сайта, заданное с помощью тега **<meta>** и значения description, обычно отображается в поисковых системах или каталогах при выводе результатов поиска. Значение keywords также предназначено в первую очередь для повышения рейтинга сайта в поисковых системах, в нем перечисляются ключевые слова, встречаемые на веб-странице (пример 2).

**Пример 2. Использование keywords**

<!DOCTYPE HTML>

**<html>**

**<head>**

**<title>**HTML**</title>**

**<meta** name="keywords" content="HTML, META, метатег, тег, поисковая система"**>**

**<meta** http-equiv="content-type" content="text/html; charset=utf-8"**>**

**</head>**

**<body>**

**<p>**...**</p>**

**</body>**

**</html>**

Ключевые слова можно перечислять через пробел или запятую. Поисковые системы сами приведут запись к виду, который они используют.

# *<!-- Комментарий -->*

Комментарии представляют собой текст, который не виден на веб-странице. Они используется для разного рода пояснений, напоминаний, описаний для вебмастеров, что позволяет структурировать документ. Комментарии незаменимы при отладке кода, позволяют быстро сориентироваться в разметке веб-страницы и найти нужный блок. Часто комментарии применяются для отладки кода HTML. К примеру, можно временно закомментировать конкретный блок кода, чтобы он не исполнялся, а в случае необходимости легко его восстановить. Начинаются они тегом ***<!--***и заканчиваются тегом ***-->.*** Все, что находится между этими тегами, отображаться на веб-странице не будет.

# *Тег <p>*

Тег ***<p>*** определяет абзац (параграф) текста. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

***<p>Страница на HTML5</p>***

Тег <p> является блочным элементом, поэтому текст всегда начинается с новой строки, абзацы идущие друг за другом разделяются между собой отбивкой (так называется пустое пространство между ними). Это хорошо видно на рис. 4.1.

***</body>***

Следует добавить закрывающий тег </body>, чтобы показать, что тело документа завершено.

***</html>***

Последним элементом в коде всегда идет закрывающий тег </html>.

Скопируйте код и посмотрите что должно получиться.

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Пример страницы**</title>**

**</head>**

**<body>**

**<p>**Страница на HTML5**</p>**

**</body>**

**</html>**