

Для создания веб-страниц часто используется язык гипертекстовой разметки HTML. Конечно, каждый сайт индивидуален, но существуют общие правила построения HTML-страниц. Им обязательно нужно следовать — только в этом случае ваш HTML-код будет верно распознаваться и отображаться браузером.

1.1. Общие понятия HTML

Существуют общие правила записи HTML-документов и принципы, используемые при создании сайтов. Их мы и рассмотрим в этом разделе.

Элемент

Основой языка HTML является *элемент*. Он несет в себе определенную информацию, может описывать документ в целом или способ форматирования текста. Элементы можно назвать основой построения сайта. Все, что вы захотите создать на вашей странице, будет сделано с помощью элементов.

Название элемента помещается в угловые скобки, например <P>. Полученное выражение называется *тегом*. В данном случае это открывающий тег. Иногда необходимо задать парный закрывающий тег, который записывается так: </P>. В основном парные теги используются при форматировании текста, они задают начало и конец блока форматирования. Иногда закрывающий тег вообще не требуется, а иногда его можно пропустить, однако для корректной обработки документа рекомендуется всегда использовать закрывающий тег.

Кратко функции данных тегов можно описать так: открывающий тег включает форматирование, а закрывающий выключает. При этом основным отличием в записи тегов, кроме постановки символа / в закрывающем теге, является отсутствие атрибутов у последнего.

Примером необходимости использования закрывающего тега является работа с элементом P, который обозначает абзац:

```
<P>Текст абзаца</P>
```

Однако и в данном случае закрывающий тег является необязательным, но желательным. Элемент IMG, который добавляет картинку на сайт, наоборот, не требует наличия закрывающего тега. По назначению элемента зачастую можно догадаться, требуется ли ему закрывающий тег.

Элементы применяются для того, чтобы сказать браузеру, какой блок вы хотите видеть в определенном месте страницы, а также какую информацию этот блок

должен содержать. Кроме того, браузеру нужно сообщить, как отображать эту информацию. Для этого используют атрибуты элементов.

Атрибут

С помощью *атрибутов* можно указывать различные способы отображения информации, добавляемой с помощью одинаковых элементов, а в некоторых случаях применение элемента без атрибутов не дает результатов. Например, в одном абзаце нужно выровнять текст по левому краю, а в другом — по правому. Чтобы задать выравнивание абзаца, используем атрибут `align` элемента `P`:

```
<P align="left">Выравнивание по левому краю</P>
```

```
<P align="right">Выравнивание по правому краю</P>
```

Значения атрибутов задаются после знака равенства в кавычках и могут быть разными. Некоторым атрибутам присущ набор фиксированных значений, для других количество значений не ограничено.

Элементы и их атрибуты являются основой языка HTML, но для правильного отображения страницы в браузерах еще важно верно создать структуру документа. Для этого существуют строгие правила. Есть элементы, без которых HTML-документ не может обойтись, потому что именно они определяют его структуру.

1.2. Структура HTML-документа

Для создания структуры документа и хранения служебной информации в нем предусмотрено много элементов, которые охватывают все необходимые пункты построения документа.

Из листинга 1.1 видно, что HTML-документ содержит следующие компоненты:

- строку объявления типа документа;
- декларативный заголовок;
- тело документа.

Листинг 1.1. Описание документа в элементе DOCTYPE

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<HTML>  
<HEAD>  
<TITLE>Это листинг структуры документа HTML</TITLE>
```

...Элементы заголовка...

```
</HEAD>
```

```
<BODY>
```

...Тело документа...

```
</BODY>
```

```
</HTML>
```

Объявление типа документа

В начале каждого HTML-документа следует помещать строку объявления такого рода:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Эта строка дает браузеру общую информацию об HTML-документе. Несмотря на то что вам вряд ли придется заполнять строку иначе, рассмотрим ее по частям и выясним, какую информацию о документе несут данные из элемента DOCTYPE.

- ❑ HTML — показывает, что для создания документа применяется язык HTML.
- ❑ PUBLIC — указывает на то, что при создании документа использована общепринятая версия HTML.
- ❑ "-//W3C//DTD HTML 4.01 Transitional//EN" — задает публичное имя спецификации языка, применяемого для разметки документа. Если браузер по этому имени сможет распознать, где находятся правила обработки данного документа, он воспользуется ими, иначе сможет загрузить их по ссылке в следующем атрибуте. В данном случае это язык HTML версии 4.01, новейшей на момент написания книги.
- ❑ "http://www.w3.org/TR/html4/loose.dtd" — URL-адрес документа, содержащего *наборы определений типа документа*, подготовленного в соответствии с языком HTML 4.01.

Что такое набор правил определения типа документа (Document Type Definition, DTD), мы обсудим в конце книги, когда приступим к освоению языка XHTML. Сейчас же просто запомните, что это сведения, которые необходимы браузеру или другой программе, предназначенной для работы с данным документом HTML, для его правильной обработки. Для документов HTML 4.01 организация W3C определила три набора таких правил DTD.

- ❑ Набор *строгих правил DTD*, которые требуют, чтобы данный HTML-документ точно соответствовал всем требованиям спецификации HTML 4.01. Документы с этим набором правил содержат такое объявление:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

- ❑ Набор *переходных (transitional) правил DTD*, которые допускают использование устаревших, не поддерживаемых в версии HTML 4.01 элементов и атрибутов. Документы с этим набором правил содержат такое объявление:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

- ❑ Набор *правил DTD для документов HTML 4.01 с фреймами*. Что такое фреймы, вы уже знаете: если веб-страница выглядит в окне браузера как набор нескольких окон со своими полосами прокрутки, значит, это и есть фреймовый HTML-документ. Документы такого типа должны содержать следующее объявление:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

В этом объявлении содержится общая служебная информация о данном документе HTML 4.01. Если вы пропустите его при создании HTML-документа, браузер, скорее всего, сам сможет догадаться, как ему отобразить соответствующую страницу, но организация W3C настоятельно рекомендует включать объявления DOCTYPE во все разрабатываемые документы.

Может возникнуть вопрос: а зачем существуют строгие и переходные правила DTD и какие из них следует использовать? Ответ связан с историей развития Интернета в целом и языка HTML в частности. Отказ от устаревших средств языка и использование строгих DTD могут привести к тому, что ваш документ HTML 4.01 не будет корректно воспроизводиться отдельными устаревшими программами, поэтому в настоящее время безопаснее использовать переходный набор DTD.

После того как вы ввели общую информацию о странице, нужно разобраться с ее структурой.

Элемент HTML

Корневым элементом документа HTML 4.01 является `<HTML>`. Это значит, что все остальные элементы содержатся внутри тегов `<HTML>` и `</HTML>`. Тем не менее в документах HTML 4.01 этот элемент не является обязательным, хотя W3C рекомендует включать его. В XHTML-документах наличие HTML обязательно.

В элементе HTML могут применяться следующие атрибуты.

- ❑ `lang` — определяет язык документа.
- ❑ `dir` — задает направление чтения на языке документа (RTL — справа налево, LTR — слева направо).

- ❑ `version` — определяет версию стандарта HTML, использованного при составлении документа. Это устаревший атрибут, и его применение не рекомендовано.
- ❑ `title` — определяет всплывающую подсказку для страницы.

В листинге 1.2 представлен пример использования элемента HTML вместе с атрибутами, указывающими на использование русского языка и направление чтения слева направо.

Листинг 1.2. Элемент HTML с атрибутами

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html lang="ru" dir="LTR" title="Моя страничка">  
  <!--Содержимое документа-->  
</html>
```

На рис. 1.1 представлен результат использования атрибутов элемента HTML.

Следует отметить, что атрибут `dir` в данном случае не обеспечен достаточной поддержкой известных браузеров даже последних версий. Например, IE 6 и Opera 9 просто выравнивают текст по правому краю, а некоторые устаревшие браузеры этот атрибут просто игнорируют. Так что, если вы намерены использовать текст с альтернативным направлением чтения (иврит, арабский язык), этот атрибут нужно применять с осторожностью.

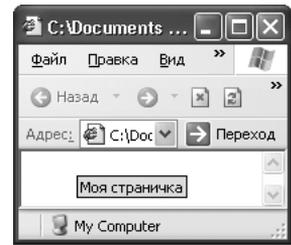


Рис. 1.1. Значение атрибута `title` отображено в виде экранной подсказки

Элемент HEAD

После того как для документа создана основа, внутри нее нужно создать заглавную область. Информация, вводимая в элемент HEAD, не отображается в окне браузера, а помогает ему в обработке страницы.

В заголовке должны присутствовать как открывающий, так и закрывающий теги `<HEAD>` и `</HEAD>`, между которыми располагаются другие элементы, несущие служебную информацию о странице. Элементы, находящиеся внутри элемента HEAD, играют очень важную роль: данные, содержащиеся в них, помогают браузеру в обработке страницы, а поисковым системам — в индексации документа.

Для элемента HEAD определены те же атрибуты, что и для HTML, а также атрибут `profil`. Он позволяет задать адрес файла с настройками, которые устанавливаются элементами `<META>` внутри заголовка. С его помощью можно будет избежать мно-

гократной записи одних и тех же элементов <МЕТА>. Однако этот атрибут пока не введен в действие и рассчитан на будущее развитие языка HTML.

Элементы, которые можно использовать внутри элемента HEAD, представлены в табл. 1.1.

Таблица 1.1. Элементы, используемые внутри элемента HEAD

Элемент	Описание
TITLE	Задаёт заголовок окна, отображающего документ, требует наличия закрывающего тега. В заголовке может быть только один такой элемент, и его наличие обязательно
META	Определяет различную служебную информацию
SCRIPT	Позволяет добавлять сценарии (см. гл. 12)
LINK	Задаёт ссылку на таблицы стилей (см. гл. 7–9)
STYLE	Позволяет добавить стили для страницы (см. гл. 7–9)
BASE	Задаёт базовый адрес документа
OBJECT	Определяет методы обработки нестандартного содержимого (см. гл. 4)

В таблице элементы описаны кратко, дальше мы рассмотрим некоторые из них подробнее. Как можно увидеть из краткого описания в табл. 1.1, они содержат данные, единые для всей страницы.

Начнем с элемента, определяющего заголовок страницы.

Элемент TITLE

Он задает название страницы, которое будет отображаться в строке заголовка окна браузера. Согласно спецификации HTML 4.01 в содержимом элемента HEAD обязательно наличие элемента TITLE, причем в единственном числе.

Элемент требует наличия закрывающего тега </TITLE>. Текст, содержащийся между открывающим и закрывающим тегами, и будет отображаться в строке заголовка окна браузера.

Помимо основной функции — рассказать посетителю, о чем страница, — элемент выполняет несколько косвенных задач. Некоторые поисковые системы используют текст, содержащийся в этом элементе, для поиска и выводят его в качестве заголовка результата поиска, поэтому корректно и качественно составленное заглавие может привлечь посетителей на сайт.

По тексту заголовка пользователь получает дополнительную информацию: что это за сайт и как называется текущая страница. Не стоит думать, что достаточно в документе указать логотип сайта и проигнорировать заголовок. Посетитель может

сворачивать окна, и тогда заголовки будут отображаться на кнопках Панели задач — по ним можно будет легко сориентироваться, с каким сайтом работать.

Большинство браузеров поддерживают возможность сохранения веб-страницы на компьютер. В этом случае имя сохраненного файла совпадает с названием заголовка документа. Если в тексте заголовка содержатся символы, недопустимые в имени файла (`\ / : * ? " < > |`), то они будут проигнорированы или заменены другими разрешенными символами.

При сохранении в разделе браузера Избранное в качестве названия ссылки будет использоваться текст, записанный в элементе `TITLE`. В этом случае адрес текущей страницы с ее заголовком помещается в список ссылок. Поскольку этот список, как правило, хранится в виде отдельных файлов, к их именам также применяется вышеописанное правило.

В листинге 1.3 показан пример использования элемента `TITLE`.

Листинг 1.3. Использование элемента `TITLE`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Пример создания заголовка страницы</title>
    <!--Другое содержимое заголовка-->
  </head>
  <!--Содержимое документа-->
</html>
```

Название страницы нужно придумывать, логически исходя из ее содержимого.

На рис. 1.2 представлен результат работы листинга 1.3. Видно, что текст заголовка отображается в строке заголовка окна.

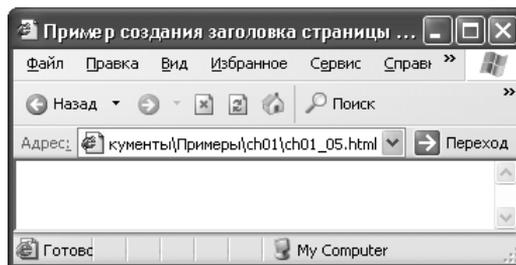


Рис. 1.2. Вид заголовка страницы