

Глава 1

HTML и CSS

Структура языка HTML

Структура HTML-документа

Ввод и оформление текста

Ссылки

Форматирование текста

Работа с таблицами

Изображения

Формы

Общие принципы CSS

Форматирование текста средствами CSS

Оформление документа CSS

Резюме

В этой главе излагается, что такое язык HTML и как с его помощью можно создавать простые веб-страницы. Вы также узнаете об огромных возможностях по управлению содержимым HTML-документов, которые предоставляет CSS.

Структура языка HTML

Для создания страниц в Интернете служит язык гипертекстовой разметки (HTML — HyperText Markup Language). Каждый сайт, конечно, индивидуален, однако существуют общие правила построения HTML-страниц, следовать которым нужно обязательно.

Существуют общие правила записи HTML-документов и общие понятия, используемые при создании сайтов. Вначале рассмотрим понятия.

Тег

Основой языка HTML является тег. Этот элемент несет в себе определенную информацию, может описывать документ в общем или способ форматирования текста. Все содержимое веб-страниц задается и редактируется с помощью тегов.

Тег помещается в угловые скобки `<ТЕГ>`. Иногда необходимо задать парный закрывающий тег, который пишется так: `</ТЕГ>`. В некоторых случаях закрывающий тег вообще не требуется, а иногда его можно пропустить. Однако для корректной обработки документа рекомендуется всегда использовать закрывающий тег. Информация, заключенная между открывающимся и закрывающимся тегом, называется его *контейнером*.

Кратко функции тегов можно описать так: открывающий тег включает форматирование, а закрывающий — выключает его.

Примером использования закрывающегося тега является тег `<P>`, который обозначает абзац: `<P>Текст абзаца</P>`, при этом закрывающий тег не является обязательным, хотя желателен. Закрывающего тега не требует тег ``, который добавляет картинку на сайт.

Тег нужен для того, чтобы сообщить браузеру, какой элемент должен быть на месте тега или какого типа информация заключена внутри тега. Кроме того, теги показывают, как следует отображать эту информацию.

Атрибут

Атрибуты сообщают браузеру, каким образом должен отображаться тот или иной элемент страницы. Атрибуты позволяют сделать более разнообразным внешний вид информации, добавляемой с помощью одинаковых тегов. Следует отметить, что в некоторых случаях использование тега без атрибутов не дает результатов.

Например, в одном абзаце нужно выровнять текст по левому краю, а в другом — по правому. Чтобы задать выравнивание абзаца, используется атрибут ALIGN тега <P>.

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

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

Значение атрибута задается после знака равенства, как правило, в кавычках. Значения атрибутов различаются в зависимости от тега.

Теги и их атрибуты являются основой языка HTML, но для правильного отображения страницы в браузерах важно также по строгим правилам правильно создать структуру документа. Существуют теги, без которых HTML-документ не может обойтись, потому что именно они определяют его структуру.

Структура HTML-документа

Для создания структуры документа и хранения служебной информации о нем предусмотрено много тегов. Рассмотрим их по порядку.

Тег <!DOCTYPE>

Этот тег записывается в начале HTML-файла. Он описывает создаваемый документ и предоставляет основную информацию для браузера.

В листинге 1.1 приводится пример заполнения информацией тега <!DOCTYPE>. Для большинства сайтов текст внутри этого тега будет таким.

Листинг 1.1. Описание документа в теге <DOCTYPE>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Эта строчка дает браузеру общую информацию об HTML-документе. Вряд ли вам придется заполнять эту строку иначе, однако рассмотрим ее по частям и выясним, какую информацию о документе несут данные из тега `<!DOCTYPE>`.

- ❑ `html` — показывает, что для создания документа используется язык HTML.
- ❑ `PUBLIC` — указывает на то, что при создании документа использовалась общепринятая версия HTML.
- ❑ `"-//W3C//DTD XHTML 1.0 Transitional//EN"` — отражает, какая версия языка используется, а также какие элементы и атрибуты можно применять.
- ❑ `"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"` — адрес URL-документа, в котором описываются элементы и атрибуты, используемые в соответствии с версией языка. В этом теге содержится общая служебная информация о странице. Он не влияет на то, как страница будет выглядеть на экране, более того, без него страница не сможет нормально загрузиться. Использование этого тега подразумевается стандартом HTML.

После того как мы ввели общую информацию о странице, надо разобраться с ее структурой.

Тег `<HTML>`

Главным структурным тегом страницы является тег `<HTML>`, все содержимое страницы записывается внутри его контейнера. Все, что находится за пределами тегов `<HTML>` и `</HTML>`, не воспринимается браузером как код HTML и никак им не обрабатывается.

В листинге 1.2 представлен пример использования тега `<HTML>`.

Листинг 1.2. Применение тега `<html>`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<!-- Содержимое документа -->
</html>
```

Тег `<HTML>` обязательно должен использоваться с закрывающим тегом, который указывает на окончание документа.

В теге `<HTML>` могут применяться следующие атрибуты:

- ❑ `xmlns` — задает пространство имен для xml-документа, поскольку наш документ является разновидностью xml-документа;

- ❑ `lang` — задает язык документа;
- ❑ `xml:lang` — тоже задает язык документа;
- ❑ `title` — задает всплывающую подсказку для страницы.

В листинге 1.3 приведен пример использования тега `<HTML>` с атрибутами.

Листинг 1.3. Тег `<HTML>` с атрибутами

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" title="Моя страничка" lang="ru"
xml:lang="ru">
<!-- Содержимое документа -->
</html>
```

На рис. 1.1 представлен результат использования атрибутов тега `<HTML>`.

На рис. 1.1 вы видите всплывающую подсказку, которая появляется при наведении указателя мыши на страницу, открытую в браузере. Эта подсказка кратко описывает сайт. Остальные атрибуты тега не влияют на отображение страницы.

После того как вы создали блок для документа, внутри него можно сформировать заглавную часть документа.

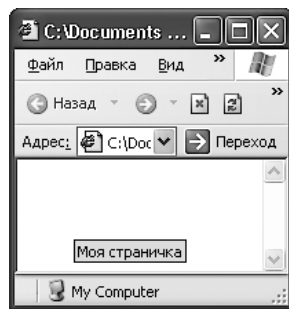


Рис. 1.1. Использование атрибутов тега `<HTML>`

Тег `<HEAD>`

Тег `<HEAD>` используется для служебных целей, введенная в нем информация не отображается в окне браузера, однако он содержит множество данных, которые указывают браузеру, как следует обрабатывать страницы.

Тег `<HEAD>` требует закрывающий тег `</HEAD>`. Между ними располагаются другие теги, которые несут служебную информацию о странице. Теги, находящиеся внутри блока тегов `<HEAD>`, играют очень важную роль: данные, которые в них содержатся, помогают браузеру в обработке страницы, а поисковым системам — в индексации документа.

Теги, которые можно использовать внутри блока тегов `<HEAD>`, представлены в табл. 1.1.

Теги в табл. 1.1 описываются кратко. Некоторые из них мы рассмотрим далее более подробно.

Таблица 1.1. Теги, используемые внутри тегов <head>

Тег	Описание
<title>	Задаёт заголовок окна, требует закрывающего тега
<meta>	Задаёт различную служебную информацию
<script>	Позволяет добавлять сценарии
<link>	Задаёт ссылку на таблицы стилей
<style>	Позволяет добавить стили для страницы
<base>	Задаёт базовый адрес документа
<basefont>	Задаёт для страницы шрифт по умолчанию
<bgsound>	Фоновый звук на странице

Тег <HEAD> не имеет никаких атрибутов, он выполняет исключительно структурирующие функции, показывая обработчику часть документа, в которой хранится служебная информация.

В листинге 1.4 представлен пример записи тега <HEAD>.

Листинг 1.4. Тег <HEAD>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<!--Содержимое заголовка-->
</head>
<!--Содержимое документа-->
</html>
```

Рассмотрим подробнее теги, которые располагаются внутри блока тегов <HEAD>. Как можно увидеть из их краткого описания в табл. 1.1, они содержат информацию и данные, единые для всей страницы.

Начнем с тега, определяющего заголовок страницы.

Тег <TITLE>

Тег <TITLE> задаёт название страницы, которое будет отображаться в строке с названием приложения окна браузера. Этот тег требует закрывающий тег </TITLE>. Текст, содержащийся между этими тегами, отображается в качестве заголовка страницы.

Помимо основной функции (рассказать посетителю, какую информацию содержит страница), этот тег решает несколько косвенных задач. Например, при сохранении страницы в разделе «Избранное» в качестве названия ссылки будет использоваться текст, записанный в теге `<TITLE>`. Некоторые поисковые системы используют текст, содержащийся в этом теге, для поиска и выводят его в качестве заголовка результата поиска. Следовательно, корректное заглавие может привлечь внимание посетителей к сайту.

ПРИМЕЧАНИЕ



По тексту заголовка пользователь получает дополнительную информацию: что это за сайт, на котором он находится, и как называется текущая страница. Не следует думать, что достаточно в документе указать логотип сайта и проигнорировать заголовок: посетитель может свернуть окно, и тогда заголовок будет отображаться на кнопке панели задач. По нему 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 проигнорируются или заменяются дозволенными символами.

При сохранении в разделе браузера «Избранное» адрес текущей страницы с ее заголовком помещается в список ссылок. Как правило, этот список хранится в виде отдельных файлов, потому к их именам также относится вышеописанное правило.

В листинге 1.5 показан пример использования тега `<TITLE>`.

Листинг 1.5. Использование тега `<TITLE>`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Пример создания заголовка страницы</title>
</head>
<!--Содержимое документа-->
</html>
```

На рис. 1.2 представлен результат выполнения листинга 1.5: текст заголовка отображается в заголовке окна.

Мы определились с заголовком новой страницы, теперь посмотрим, какая служебная информация может содержаться внутри тега `<HEAD>`.

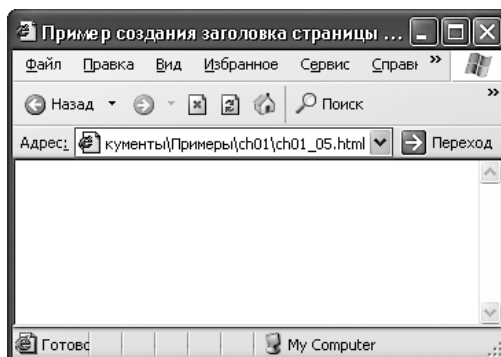


Рис. 1.2. Вид заголовка страницы

Тег <META>

Тег <META> используется для хранения дополнительной информации о странице. Эту информацию используют браузеры для обработки страницы, а поисковые системы — для ее индексации.

В блоке <HEAD> можно использовать несколько тегов <META>, потому что в зависимости от используемых атрибутов они могут нести разную информацию.

Рассмотрим атрибуты тега <META>.

Атрибут CONTENT используется вместе с атрибутами HTTP-EQUIV и NAME. Он задает значение свойства, которое указывается с их помощью.

Атрибут HTTP-EQUIV устанавливает служебные свойства документа, например его кодировку, дает браузеру информацию о том, как нужно обрабатывать документ.

В табл. 1.2 представлены возможные значения атрибута HTTP-EQUIV.

Таблица 1.2. Возможные значения атрибута HTTP-EQUIV

Значение атрибута	Описание
Content-Type	Определяет тип содержимого документа. Этот параметр желательно указывать всегда
Expires	Задаёт время действия документа. После даты, указанной для этого свойства, документ считается устаревшим
Pragma	Определяет тип кэширования вашего документа, то есть можно запретить браузеру сохранять страницу в кэш

Продолжение ➤

Таблица 1.2 (продолжение)

Значение атрибута	Описание
Refresh	Позволяет задать параметры автоматической загрузки документа в то же окно браузера, что и текущий документ
Content-Language	Задаёт язык содержимого, является аналогом атрибута lang тега html
Content-Script-Type	Типы сценариев, используемых на сайте
Content-Style-Type	Задаёт типы таблиц стилей

Рассмотрим подробнее применение описанных выше атрибутов.

В листинге 1.6 представлен пример того, как можно задать свойства обработки страницы с помощью атрибута HTTP-EQUIV.

Листинг 1.6. Применение тега HTTP-EQUIV

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" title="Моя страничка" lang="ru"
xml:lang="ru">
<head>
<title>Использование тега META</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251"
/>
<meta http-equiv="refresh" content="10; URL=http://www.test.ru" />
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires" content="Sun, Oct 2010 10:09:00 GMT+3" />
</head>
</html>
```

Как видно из этого примера, значение атрибута HTTP-EQUIV указывает на переменную, значение которой указано с помощью атрибута CONTENT.

Значение атрибута Content-Type, использованное в примере, будет одинаковым для всех сайтов в кириллической кодировке. Рекомендуется всегда указывать его, иначе браузер будет отображать текст на вашей странице некорректно.

Применение параметра Refresh полезно в случае, когда страницу перенесли в другое место или когда у страницы много адресов. В таком случае вместо создания множества сайтов по разным адресам можно организовать переброску посетителя на основной сайт. Например, в листинге 1.6 задается загрузка сайта www.test.ru через 10 с после загрузки страницы.