

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 4.1 показан код простого документа, содержащего основные теги.

Пример 4.1. Исходный код веб-страницы

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

<html>

  <head>

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

    <title>Пример веб-страницы</title>

  </head>

  <body>

    <h1>Заголовок</h1>

    <!-- Комментарий -->

    <p>Первый абзац.</p>

    <p>Второй абзац.</p>

  </body>

</html>
```

Скопируйте содержимое данного примера и сохраните его в папке c:\www\ под именем example41.html. После этого запустите браузер и откройте файл через пункт меню **Файл > Открыть файл (Ctrl+O)**. В диалоговом окне выбора документа укажите файл example41.html. В браузере откроется веб-страница, показанная на рис. 4.1.

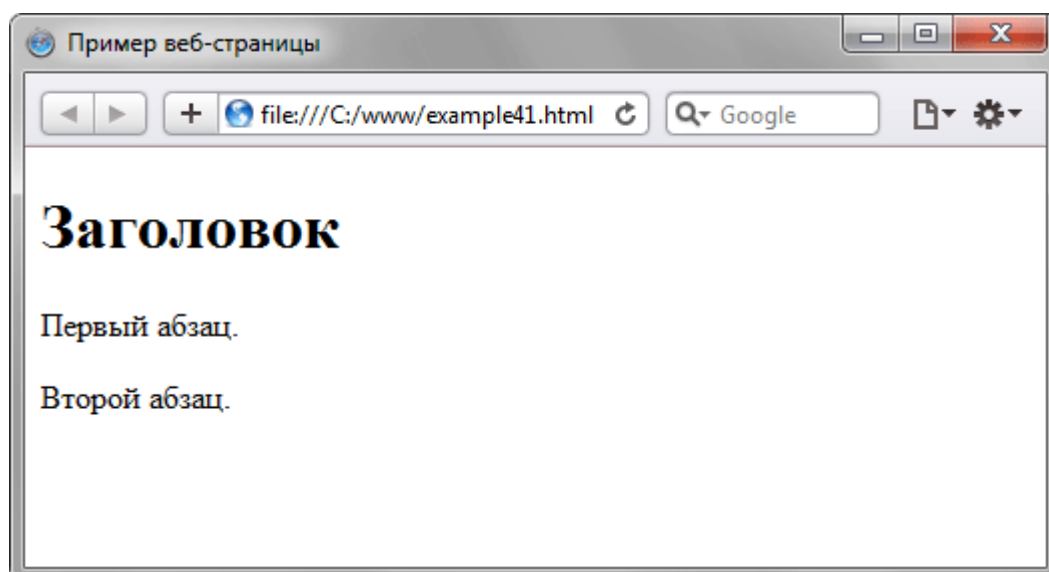


Рис. 4.1. Результат выполнения примера

Далее разберем отдельные строки нашего кода.

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

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, кроме того, имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису. 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 HTML, на которую ориентированы. В табл. 4.1. приведены основные типы документов с их описанием.

Табл. 4.1. Допустимые DTD	
DOCTYPE	Описание
HTML 4.01	
<!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.
HTML 5	
<!DOCTYPE html>	В этой версии HTML только один доктайп.
XHTML 1.0	
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">	Строгий синтаксис XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	Переходный синтаксис XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">	Документ написан на XHTML и содержит фреймы.
XHTML 1.1	

<pre><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"></pre>	Разработчики XHTML 1.1 предполагают, что он постепенно вытеснит HTML. Как видите, никакого деления на виды это определение не имеет, поскольку синтаксис один и подчиняется четким правилам.
--	---

Разница между строгим и переходным описанием документа состоит в различном подходе к написанию кода документа. Строгий HTML требует жесткого соблюдения спецификации HTML и не прощает ошибок. Переходный HTML более «спокойно» относится к некоторым огрехам кода, поэтому этот тип в определенных случаях использовать предпочтительнее.

Например, в строгом HTML и XHTML непременно требуется наличие тега `<title>`, а в переходном HTML его можно опустить и не указывать. При этом помним, что браузер в любом случае покажет документ, независимо от того, соответствует он синтаксису или нет. Подобная проверка осуществляется при помощи валидатора и предназначена в первую очередь для разработчиков, чтобы отслеживать ошибки в документе.

В дальнейшем будем применять преимущественно строгий `<!DOCTYPE>`, кроме случаев, когда это оговаривается особо. Это позволит нам избегать типичных ошибок и приучит к написанию синтаксически правильного кода.

Часто можно встретить код HTML вообще без использования `<!DOCTYPE>`, веб-страница в подобном случае все равно будет показана. Тем не менее, может получиться, что один и тот же документ отображается в браузере по-разному при использовании `<!DOCTYPE>` и без него. Кроме того, браузеры могут по-своему показывать такие документы, в итоге страница «рассыплется», т.е. будет отображаться совсем не так, как это требуется разработчику. Чтобы не произошло подобных ситуаций, всегда добавляйте `<!DOCTYPE>` в начало документа.

```
<html>
```

Тег `<html>` определяет начало HTML-файла, внутри него хранится заголовок (`<head>`) и тело документа (`<body>`).

```
<head>
```

Заголовок документа, как еще называют блок `<head>`, может содержать текст и теги, но содержимое этого раздела не показывается напрямую на странице, за исключением контейнера `<title>`.

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

Тег `<meta>` является универсальным и добавляет целый класс возможностей, в частности, с помощью метатегов, как обобщенно называют этот тег, можно изменять кодировку страницы, добавлять ключевые слова, описание документа и многое другое. Чтобы браузер понимал, что имеет дело с кодировкой UTF-8 (Unicode transformation format, формат преобразования Юникод) и добавляется данная строка.

```
<title>Пример веб-страницы</title>
```

Тег `<title>` определяет заголовок веб-страницы, это один из важных элементов предназначенный для решения множества задач. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера (рис. 4.2).

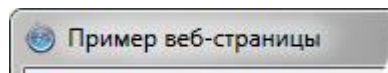


Рис. 4.2. Вид заголовка в браузере

Тег `<title>` является обязательным и должен непременно присутствовать в коде документа.

```
</head>
```

Обязательно следует добавлять закрывающий тег `</head>`, чтобы показать, что блок заголовка документа завершен.

```
<body>
```

Тело документа `<body>` предназначено для размещения тегов и содержательной части веб-страницы.

```
<h1>Заголовок</h1>
```

HTML предлагает шесть текстовых 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.

```
<!-- Комментарий -->
```

Некоторый текст можно спрятать от показа в браузере, сделав его комментарием. Хотя такой текст пользователь не увидит, он все равно будет передаваться в документе, так что, посмотрев исходный код, можно обнаружить скрытые заметки.

Комментарии нужны для внесения в код своих записей, не влияющих на вид страницы. Начинаются они тегом `<!--` и заканчиваются тегом `-->`. Все, что находится между этими тегами, отображаться на веб-странице не будет.

```
<p>Первый абзац.</p>
```

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

```
<p>Второй абзац.</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>`.