

Глава 2. Общая структура HTML-документа

2.1. Общие правила

2.1.1. Элементы

Элемент (тег, tag) - способ разметки текста.

В HTML-документе элемент - это метка, которая прикрепляется к участку простого текста и служит для сообщения о нем дополнительной информации, например, что участок текста надо вывести жирным шрифтом.

Когда браузер получает символы простого текста, он их просто отображает на экране, например, если текст содержит символы "Hello world!", то и на экране появляются именно они. Совершенно другие события должны происходить, когда браузер встречает элемент. Символы, составляющие элемент, браузер не должен выводить на экран, а должен анализировать. Потом результаты этого анализа он должен использовать для отображения другой информации, к которой прикреплен элемент.

Имена элементов можно записывать как прописными, так и строчными буквами.

Чтобы это произошло, некоторым символам должна быть дана роль специальных, управляющих символов. Главными из таких символов являются открывающая и закрывающая угловые скобки:

"<" и ">".

Синтаксис
<code><элемент></code> фрагмент страницы

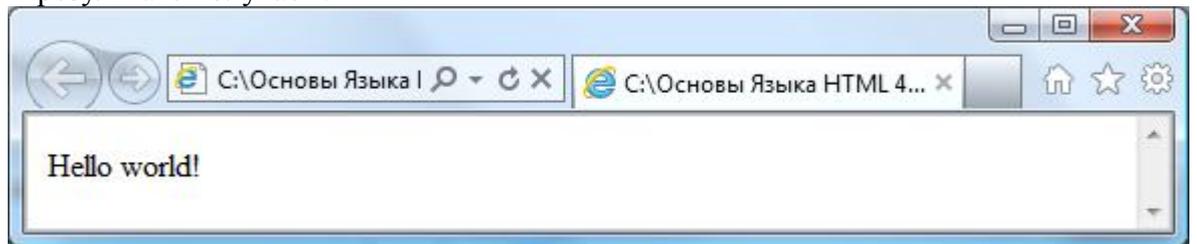
Создадим простейший пример.

В текстовом редакторе достаточно набрать текст: Hello world!

Пример:

Содержимое файла sample2.1.1-1.html:
Hello world!

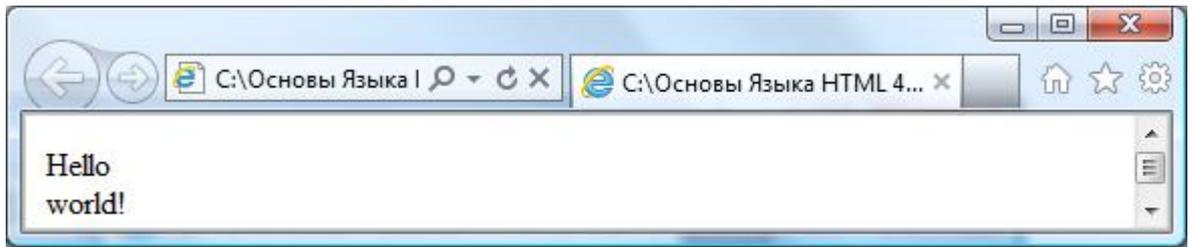
В результате получаем:



Ещё один пример, теперь введём управляющий элемент `
`.

Содержимое файла sample2.1.1-2.html:
Hello
world!

В результате слово "world" отображается на новой строке:



2.1.2. Элементы-контейнеры

Некоторые элементы состоят из пары - открывающего элемента и закрывающего элемента. Закрывающий элемент дополнительно содержит после открывающей угловой скобки символ "/". Например:

`текст страницы`

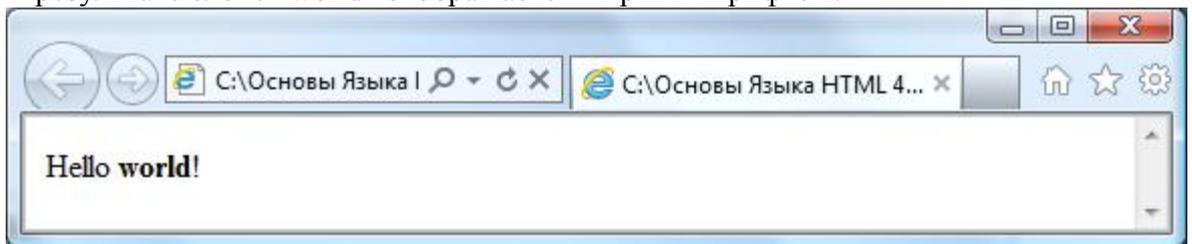
Текст, расположенный между открыванием и закрыванием называется содержимым элемента. Элементы, у которых есть содержимое, называются **элементами-контейнерами**.

Синтаксис
<code><элемент>содержимое элемента</элемент></code>
Общий случай:
<code><элемент атрибут1="значение1" атрибут2="значение2" ... ></code> <code>содержимое элемента</элемент></code>

Пример с использованием элемента-контейнера.

Содержимое файла sample2.1.2-1.html:
Hello <code></code> world <code></code> !

В результате слово "world" отображается жирным шрифтом:



В примере все, что идет после символов `` перестанет отображаться жирным. В примере восклицательный знак отображается уже обычным шрифтом.

Элементы могут вкладываться друг в друга иерархически, но без пересечений.

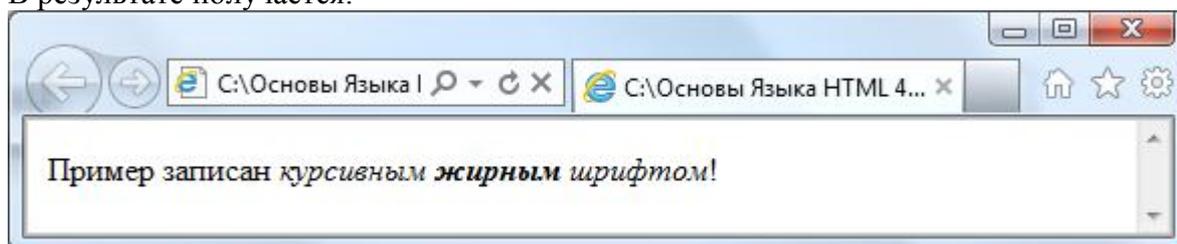
Синтаксис
<code><элемент1>содержимое элемента 1...</code> <code><элемент2>содержимое элемента 2</элемент2></code> <code>содержимое элемента 1</элемент1></code>

Например, чтобы пометить текст курсивом, а внутри него сделать часть жирным шрифтом:

Содержимое файла sample2.1.2-2.html:

Пример записан *жирным* шрифтом

В результате получается:



В данном примере элемент **** вложен в элемент *<i>*.

2.1.3. Мета-элементы

Элемент **meta** - необязательный. Располагается в теле заголовка страницы. Этот элемент содержит описание страницы, ключевые слова, информацию об авторе и прочую служебную информацию. Это визитная карточка страницы для различных служб Интернета. Существуют два способа применения мета-элементов.

Первый способ позволяет определить некоторую переменную путем указания ее имени и значения.

Синтаксис
<code><meta name="имя" content="значение"></code>

Второй способ позволяет определить эквиваленты команд протокола передачи гипертекстов.

Синтаксис
<code><meta http-equiv="тип" content="значение"></code>

Мета-элементов существует очень много и далеко не все они нужны на конкретной странице.

Наиболее часто используемые значения атрибута **name**:

Имя	Пояснение
<code>Author</code>	Определяет имя автора страницы
<code>Copyright</code>	Указывает наличие авторских прав на документ
<code>Title</code>	Заглавие страницы
<code>Description</code>	Описание страницы. Если страница не содержит этот атрибут, то поисковая система в качестве аннотации возьмет начальный фрагмент текста страницы, что зачастую нежелательно, т.к. в верхней части страницы может располагаться, например, реклама
<code>Keywords</code>	Список ключевых слов страницы через запятую. Часто используется поисковыми системами. Ключевые слова должны быть простыми и описывать содержимое страницы
<code>Document-state</code>	<code>content="Dynamic"</code> . Указывает, что страница динамичная, т.е. ее содержание периодически меняется. <code>content="Static"</code> .

	Указывает, что страница статичная, т.е. ее содержание не меняется
Generator	Содержит имя HTML-редактора, в котором создана страница
Robots	content="All" . Указывает поисковому роботу занести страницу в каталог поисковой системы. content="Noindex" . Запрет поисковому роботу занести страницу в каталог поисковой системы.

Таблица 1. Значения атрибута name в мета-элементах.

Наиболее часто используемые значения атрибута **http-equiv**:

Тип	Пояснение
Reply-to	Содержит адрес электронной почты автора
Content-Type	Указывает на тип документа и его кодировку. Например, "text/html; charset=windows-1251"
Refresh	Указывает браузеру через сколько секунд после завершения загрузки текущего документа пользователь будет перенаправлен на другую страницу, адрес которой записан в элементе. Если указано только время, то будет обновляться только содержимое текущей страницы. Например, "[время в секундах];URL=[адрес страницы]"

Таблица 2. Значения атрибута http-equiv в мета-элементах.

Пример мета-элементов сайта <http://www.rsreu.ru>:

```
<meta name="description" content="Официальный сайт Рязанского государственного радиотехнического университета (РГРТУ)" />
<meta name="keywords" content="радиотехнический университет, РГРТУ, Рязань, высшее образование" />
<meta name="robots" content="index, follow" />
```

Пример мета-элементов сайта <http://www.gismeteo.ru/>:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta name="description" content="Подробный прогноз погоды по 1300+ городам России и по 13000+ городам Мира на ближайшие 72 часа на GISMETEO.RU. Долгосрочные прогнозы (на две недели). Погодные карты России и Мира.">
<meta name="keywords" content="прогноз, погода, карты, осадки, ветер, влажность, давление">
```

2.1.4. Атрибуты

Элементы могут иметь свойства, называемые атрибутами, которые могут иметь в свою очередь значения. Пары атрибут/значение появляются перед конечным символом ">" начального элемента. Допустимо любое количество пар значений атрибута, разделённых пробелами, которое разрешено правилами языка. Они могут появляться в любом порядке.

Например:

```
<p align="center">Параграф выводится с выравниванием по центру.</p>
```

2.2. Создание HTML-документа

Документы в формате HTML являются обычными текстовыми файлами с расширением "htm" или "html", содержащими специальные элементы разметки. HTML-документ можно создавать с помощью как специальных HTML-редакторов, так и с помощью любого текстового редактора.

Для изучения основ HTML достаточно воспользоваться, например, редактором Notepad (Блокнот), входящий в стандартные средства операционной системы Windows.

2.3. Структура HTML-документа

В целом HTML-документ состоит из начала, заголовка и тела.

Три части документа:

- начало - строка, содержащая информацию о версии HTML и элемент `html`;
- заголовок - раздел, ограниченный элементом `head`;
- тело, содержащее собственно сам документ. Тело может выполняться в элементах `body` или `frameset`.

2.4. Объявление типа документа

Для того, чтобы текстовый файл превратился в HTML-файл, поменять его расширение с "txt" на "html" недостаточно.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки объявления типа документа (ОТД) `<!DOCTYPE>`, например: `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">`

Эта строка поможет браузеру определить, как правильно интерпретировать данный документ. В примере строки объявления указывается, что далее HTML-страница соответствует международной спецификации версии 4.01.

Корректный документ HTML объявляет, какую версию HTML использует данный документ.

HTML 4.01 предлагает три варианта ОТД, поэтому необходимо включить в свой документ одно из нижеследующих объявлений.

HTML 4.01 Strict - строгое ОТД включает все элементы и атрибуты, которые не являются не рекомендуемыми или не входят в документ типа "набор фреймов". Для документов, использующих это ОТД, следует такое объявление типа:

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

HTML 4.01 Transitional - переходное ОТД включает всё из строгого ОТД плюс не рекомендуемые элементы и атрибуты (большинство из которых касается визуального представления). Для документов, использующих это ОТД, следует такое объявление типа:

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

HTML 4.01 Frameset - набор фреймов ОТД включает всё из переходного ОТД плюс сами фреймы. Для документов, использующих это ОТД, следует такое объявление типа:

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

2.5. Начало HTML-документа

После объявления версии и типа документа необходимо обозначить его начало и конец. Это делается с помощью элемента `html`.

Синтаксис
<code><html></code> <i>текст страницы</i> <code></html></code>

2.6. Заголовок HTML-документа

Элемент `head` содержит информацию о текущем документе, такую как название, ключевые слова, которые могут оказаться полезными при использовании машины поиска, и другие данные, не являющиеся содержимым документа.

Заголовок определяется с помощью элемента `head`, который вложен в элемент `html`.

Синтаксис
<code><head></code> <i>тело заголовка</i> <code></head></code>

Одним из элементов тела заголовка служит специальный элемент `title`. Помещенный внутри него текст выводится в строке заголовка окна браузера.

Синтаксис
<code><title></code> <i>название страницы</i> <code></title></code>

Необходимо использовать элемент `title` для идентификации содержимого документа. Поскольку пользователи часто обращаются к документам вне контекста, необходимо предоставлять осмысленное название HTML-страницам. Таким образом, вместо такого заголовка, как "Введение", который не даёт достаточно информации о документе, следует записать, например, так: "Введение в HTML 4.01".

2.7. Тело HTML-документа

Именно в теле находится все то, что отображается на странице: текст, картинки, таблицы. Тело документа размещается внутри элемента `body` и вложено в элемент `html`.

Синтаксис элемента без атрибутов
<code><body></code> <i>тело страницы</i> <code></body></code>

Атрибуты данного элемента будут рассмотрены позднее.

2.8. Комментарий в HTML-документах

Комментарий в HTML заключается в элементы `<!--` и `-->`, который не отображается браузером.

Синтаксис

```
<!-- текст комментария -->
```

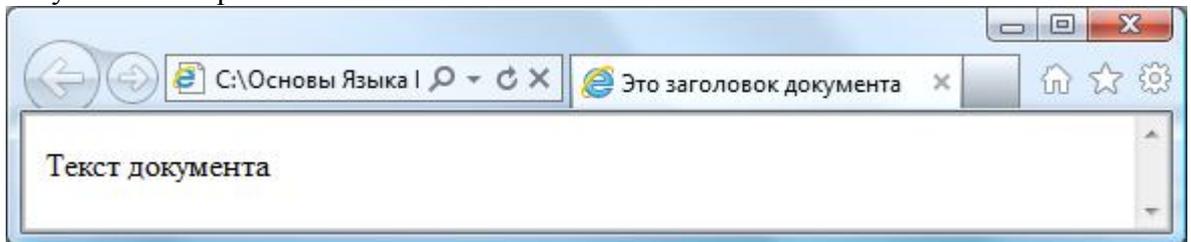
2.9. Примеры

Пример короткого HTML-документа:

Содержимое файла sample2.9-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//RU">
<html>
  <head>
    <title>Это заголовок документа</title>
  </head>
  <body>
    Текст документа
  </body>
</html>
```

Результат на экране:

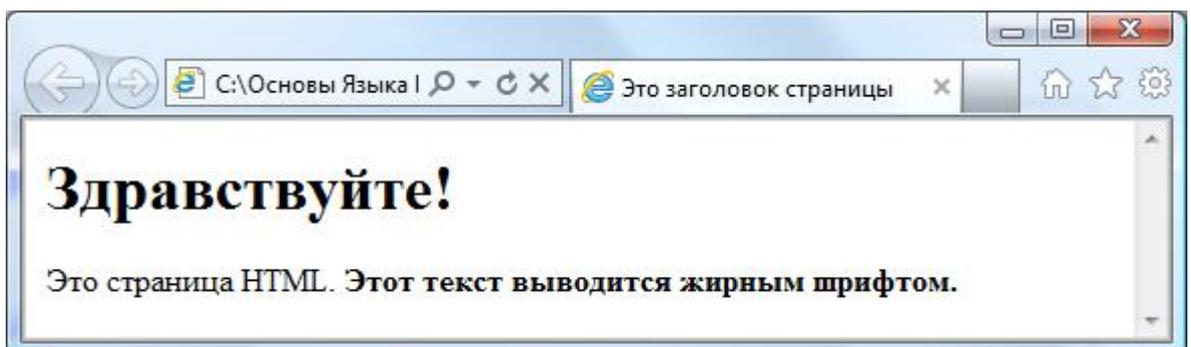


Пример:

Содержимое файла sample2.9-2.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//RU">
<html>
  <head>
    <title>Это заголовок страницы</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <h1>Здравствуйете!</h1>
    <!-- Это комментарий -->
    <p>Это страница HTML. <b>Этот текст выводится жирным шрифтом.</b></p>
  </body>
</html>
```

Результат на экране:



Разберем пример:

- Мета-элемент указывает на то, что документ содержит HTML-текст в кодировке windows-1251 (кириллица).
- Весь текст внутри элемента **body** отображается в окне браузера.
- Текст «Здравствуйте!» в элементе **h1** отображается стилем заголовка первого уровня (обычно жирным шрифтом большего размера).
- Комментарий не отображается браузером, хотя присутствует в исходном коде страницы.
- Элемент **p** означает, что начинается новый параграф, а **/p** означает конец параграфа.