

# Структурирование текста

## Абзацы

Если оформить текст в виде большого монолитного "куска", его вряд ли кто-то будет читать. Такой "кусочек" текста выглядит как высокий черный забор, за которым не видно ни единой мысли автора, забор без единой дверцы, без единой щелочки.

Именно поэтому текст всегда разбивают на абзацы. Небольшие, включающие по несколько связанных по смыслу предложений, они доносят авторский текст постепенно, по частям, от простого к сложному. В общем, превращают непроницаемый "забор" в читабельный текст.

HTML для создания абзаца предоставляет парный тег `<P>`. Содержимое этого тега становится текстом абзаца:

```
<P>Я – совсем короткий абзац.</P>
```

```
<P>А я – уже более длинный абзац. Возможно, Web-обозреватель разобьет  
меня на две строки.</P>
```

Абзац HTML отделяется небольшим отступом от предыдущего и последующего элементов страницы. Если абзац полностью помещается по ширине в родительский элемент Web-страницы, он будет выведен в одну строку; в противном случае Web-обозреватель разобьет его текст на несколько более коротких строк.

Абзац — это независимый элемент Web-страницы, который отображается отдельно от других элементов. Такие элементы Web-страницы называются *блочными*, или *блоками*.

Правила отображения текста абзаца Web-обозревателем:

- ❑ два и более следующих друг за другом пробела считаются за один пробел;
- ❑ перевод строки считается за пробел;
- ❑ пробелы и переводы строки между тегами, создающие блочные элементы, никак не отображаются на Web-странице. (Благодаря этому мы можем форматировать HTML-код для более удобного чтения, в том числе ставить отступы для обозначения вложенности тегов.)

Эти же правила справедливы для других блочных элементов, которые мы изучим

Настало время попрактиковаться. Давайте создадим главную Web-страницу нашего первого Web-сайта — справочника по HTML и CSS. Откроем Блокнот и наберем в нем HTML-код, приведенный в листинге 2.1.

### Листинг 2.1

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
    <TITLE>Справочник по HTML и CSS</TITLE>
  </HEAD>
  <BODY>
    <P>Справочник по HTML и CSS</P>
    <P>Приветствуем на нашем Web-сайте всех, кто занимается Web-дизайном!
    Здесь вы сможете найти информацию обо всех интернет-технологиях,
    применяемых при создании Web-страниц. А именно, о языках HTML и
    CSS.</P>
    <P>Русская Википедия определяет термин HTML так:</P>
    <P>HTML (от англ. HyperText Markup Language – язык разметки
    гипертекста) – стандартный язык разметки документов во Всемирной
    паутине.</P>
    <P>Пожалуй, ни убавить ни прибавить...</P>
    <P>HTML позволяет формировать на Web-страницах следующие
    элементы:</P>
    <P>абзацы;</P>
    <P>заголовки;</P>
    <P>цитаты;</P>
    <P>списки;</P>
    <P>таблицы;</P>
    <P>графические изображения;</P>
    <P>аудио- и видеоролики.</P>
    <P>Основные принципы HTML</P>
    <P>. . .</P>
    <P>Теги HTML</P>
```

```
<P>!DOCTYPE, BODY, EM, HEAD, HTML, META, P, STRONG, TITLE</P>  
</BODY>  
</HTML>
```

Пока это только заготовка для главной Web-страницы. Позднее мы будем дополнять и править ее.

Создадим папку, куда будем "складывать" файлы, составляющие наш Web-сайт. И сохраним в этой папке набранный HTML-код, дав файлу имя index.htm.

Сразу же откроем созданную Web-страницу в Web-обозревателе — так мы сразу сможем определить, нет ли ошибок. Если ошибки все-таки есть, исправим их.

Пока что наша Web-страничка содержит одни абзацы. Первое, что мы должны в нее добавить, — это...

## Заголовки

Помимо абзацев, большой текст для удобства чтения и поиска в нем нужного фрагмента обычно делят на более крупные части: параграфы, главы, разделы. HTML не предоставляет средств для такого структурирования текста. Но он позволяет создать заголовки, которые делят текст на части, по крайней мере, визуально. Как в обычной "бумажной" книге.

Прежде всего, уясним, что в HTML есть понятие *уровня заголовка*, указывающее, насколько крупную часть текста открывает данный заголовок. Всего таких уровней шесть, и обозначаются они числами от 1 до 6.

- Заголовок первого уровня (1) открывает самую крупную часть текста. Как правило, это заголовок всей Web-страницы. Web-обозреватель выводит заголовок первого уровня самым большим шрифтом.
- Заголовок второго уровня (2) открывает более мелкую часть текста. Обычно это большой раздел. Web-обозреватель выводит заголовок второго уровня меньшим шрифтом, чем заголовок первого уровня.
- Заголовок третьего уровня (3) открывает еще более мелкую часть текста; обычно главу в разделе. Web-обозреватель выводит такой заголовок еще меньшим шрифтом.
- Заголовки четвертого, пятого и шестого уровней (4–6) открывают отдельные параграфы, крупные, более мелкие и самые мелкие соответственно. Web-обозреватель выводит заголовки четвертого и пятого уровня еще меньшим шрифтом, а заголовок шестого уровня — тем же шрифтом, что и обычные абзацы, только полужирным.

На Web-страницах небольшого и среднего размера обычно применяют заголовки первого, второго и третьего уровня. Меньшие уровни используются только на очень больших Web-страницах, содержащих сложно структурированный текст.

Для создания заголовка HTML предоставляет парный тег `<h $n$ >`, где  $n$  — уровень заголовка. Содержимое этого тега станет текстом заголовка (листинг 2.2).

### Листинг 2.2

```
<h1>Я — заголовок Web-страницы, самый главный</h1>
<h2>Я — заголовок раздела</h2>
<h3>Я — заголовок главы</h3>
<h4>Я — заголовок крупного параграфа</h4>
<h5>Я — заголовок параграфа поменьше</h5>
<h6>А я — заголовок маленького параграфа. Ой, какие все вокруг
большие!..</h6>
```

Заголовок также относится к блочным элементам Web-страницы. При его выводе на экран Web-обозреватель следует тем же правилам, что и для абзаца.

Заголовки — это то, чего не хватает нашей Web-страничке `index.htm`. Давайте их добавим (листинг 2.3).

### Листинг 2.3

```
<h1>Справочник по HTML и CSS</h1>
. . .
<h2>Основные принципы HTML</h2>
. . .
<h2>Теги HTML</h2>
```

Мы просто заменили теги `<p>` в соответствующих фрагментах HTML-кода на теги `<h1>` и `<h2>`. Теперь можем открыть Web-страницу в Web-обозревателе и посмотреть на результат.

## Списки

Списки используются для того, чтобы представить читателю перечень каких-либо позиций, пронумерованных или пронумерованных, — пунктов списка. Список с пронумерованными пунктами так и называется — *нумерованным*, а с пронумерованными — *маркированным*. В маркированных списках пункты помечаются особым значком — *маркером*, который ставится левее пункта списка.

Маркированные списки обычно служат для простого перечисления каких-либо позиций, порядок следования которых не важен. Если же следует обратить внимание читателя на то, что позиции должны следовать друг за другом именно в том порядке, в котором они перечислены, следует применить нумерованный список.

Web-обозреватель выводит список с отступом слева. Расстояние между пунктами списка он делает меньшими чем в случае абзацев или заголовков. Также он сам расставляет необходимые маркеры или нумерацию.

Любой список в HTML создается в два этапа. Сначала пишут строки, которые станут пунктами списка, и каждую из этих строк помещают внутрь парного тега <LI>. Затем все эти пункты помещают внутрь парного тега <UL> (если создается маркированный список) или <OL> (при создании нумерованного списка) — эти теги определяют сам список (листинг 2.4).

#### Листинг 2.4

```
<UL>
  <LI>Я — первый пункт маркированного списка.</LI>
  <LI>Я — второй пункт маркированного списка.</LI>
  <LI>Я — третий пункт маркированного списка.</LI>
</UL>
. . .
<OL>
  <LI>Я — первый пункт нумерованного списка.</LI>
  <LI>Я — второй пункт нумерованного списка.</LI>
  <LI>Я — третий пункт нумерованного списка.</LI>
</OL>
```

Списки можно помещать друг в друга, создавая *вложенные списки*. Делается это следующим образом. Сначала во "внешнем" списке (в который должен быть помещен вложенный) отыскивают пункт, после которого должен находиться вложенный список. Затем HTML-код, создающий вложенный список, помещают в разрыв между текстом этого пункта и его закрывающим тегом </LI>. Если же вложенный список должен помещаться в начале "внешнего" списка, его следует вставить между открывающим тегом <LI> первого пункта "внешнего" списка и его текстом. Что, впрочем, логично.

В листинге 2.5 представлен HTML-код, создающий два списка, один из которых вложен внутри другого. Обратите внимание, где размещается HTML-код, создающий вложенный список.

#### Листинг 2.5

```
<UL>
  <LI>Я — первый пункт внешнего списка.</LI>
  <LI>Я — второй пункт внешнего списка.
    <UL>
      <LI>Я — первый пункт вложенного списка.</LI>
      <LI>Я — второй пункт вложенного списка.</LI>
      <LI>Я — третий пункт вложенного списка.</LI>
    </UL>
```

```
</LI>
<LI>Я — третий пункт внешнего списка.</LI>
</UL>
```

HTML позволяет вкладывать нумерованный список внутрь маркированного и наоборот. Глубина вложения списков не ограничена.

Еще HTML позволяет создать так называемый *список определений*, представляющий собой перечень терминов и их разъяснений. Такой список создают с помощью парного тега <DL>. Внутри него помещают пары "термин — разъяснение", причем термины заключают в парный тег <DT>, а разъяснения — в парный тег <DD> (листинг 2.6).

#### Листинг 2.6

```
<DL>
  <DT>Содержимое</DT>
  <DD>Информация, отображаемая на Web-странице</DD>
  <DT>Представление</DT>
  <DD>Набор правил, определяющих формат отображения содержимого</DD>
  <DT>Поведение</DT>
  <DD>Набор правил, определяющих реакцию Web-страницы или ее элементов на
  действия посетителя</DD>
</DL>
```

Осталось сказать, что списки и их пункты относятся к блочным элементам Web-страницы, и при их выводе на экран Web-обозреватель руководствуется теми же правилами, что и при выводе абзацев и заголовков.

На нашей Web-странице есть несколько абзацев, перечисляющих основные возможности HTML. Превратим их в маркированный список (листинг 2.7).

#### Листинг 2.7

```
<UL>
  <LI>абзацы;</LI>
  <LI>заголовки;</LI>
  <LI>цитаты;</LI>
  <LI>списки;</LI>
  <LI>таблицы;</LI>
  <LI>графические изображения;</LI>
  <LI>аудио- и видеоролики.</LI>
</UL>
```

Теперь наша Web-страничка стала выглядеть симпатичнее.

## Цитаты

В тексте Web-страницы, которую мы создаем, присутствует большая цитата из Русской Википедии. Давайте ее как-то выделим.

HTML уже приготовил для нас выход из положения — парный тег `<BLOCKQUOTE>`, внутри которого размещается HTML-код, создающий цитату (листинг 2.8). Web-обозреватель выводит цитату с отступом слева.

### Листинг 2.8

```
<BLOCKQUOTE>
  <P>Я — начало большой цитаты.</P>
  <P>Я — продолжение большой цитаты.</P>
</BLOCKQUOTE>
```

Как видим, в тег `<BLOCKQUOTE>` можно поместить несколько абзацев. Там также могут быть заголовки и списки (если уж возникнет такая потребность).

Большая цитата HTML также относится к блочным элементам.

Осталось только сделать то, что было задумано, — оформить цитату (листинг 2.9).

### Листинг 2.9

```
<BLOCKQUOTE>
  <P>HTML (от англ. HyperText Markup Language — язык разметки
    гипертекста) — стандартный язык разметки документов во Всемирной
    паутине.</P>
</BLOCKQUOTE>
```

## Текст фиксированного формата

Аппетит приходит во время еды. Мы еще не успели доделать свою первую Web-страницу, а уже хотим сделать еще одну. Давайте же ее сделаем. Дадим аппетиту разгуляться!

Новая Web-страница (листинг 2.10) будет посвящена тегу `<TITLE>`.

### Листинг 2.10

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
    <TITLE>Тег TITLE</TITLE>
  </HEAD>
```

```

<BODY>
  <H1>Тег TITLE</H1>
  <P>Тег TITLE служит для указания названия Web-страницы. Он ставится в
    ее секции заголовка.</P>
  <H6>Пример:</H6>
  <P>!HEAD!
    !TITLE!Я – заголовок Web-страницы!/TITLE!
    !HEAD!</P>
</BODY>
</HTML>

```

Здесь мы поместили краткое описание тега `<TITLE>` и код примера, имеющий такой вид:

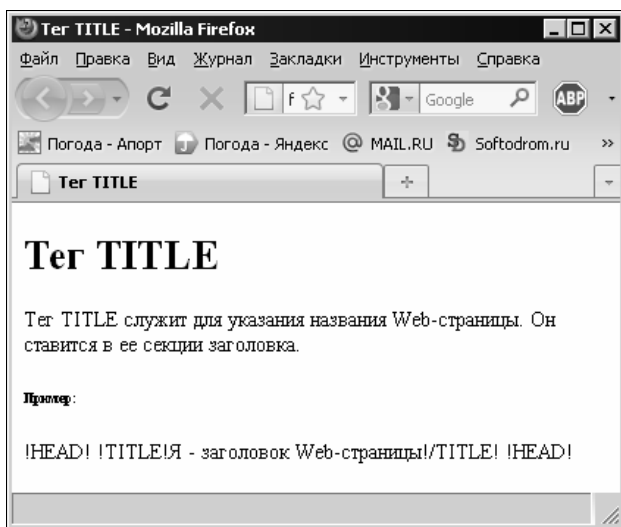
```

!HEAD!
!TITLE!Я – заголовок Web-страницы!/TITLE!
!/HEAD!

```

Вместо символов `<` и `>`, которые, как мы помним из *главы 1*, недопустимы в обычном тексте, мы поставили восклицательные знаки. В *главе 3* мы узнаем, как все-таки вставить в текст недопустимые символы, и заменим их.

Сохраним набранный код в файле с именем `t_title.htm` и откроем его в Web-обозревателе. И что мы там увидим?



**Рис. 2.1.** Web-страница `t_title.htm` в Web-обозревателе. Обратим внимание на код примера

Как видно на рис. 2.1, Web-обозреватель вывел код примера в одну строку и без всяких отступов. Но ведь мы разбили его на три строки и создали отступы с помощью пробелов, чтобы HTML-код лучше читался и сразу была видна вложенность тегов! Что случилось?

Вспомним два правила, которыми руководствуется Web-обозреватель при выводе текста блочного элемента и которые были перечислены в разделе, посвященном



абзацам. Эти правила гласят, что два или более следующих друг за другом пробела или переноса строки преобразуются в один пробел и перенос строки считается за пробел. Так Web-обозреватель и поступил: преобразовал переносы строки в пробелы, а последовательные пробелы — в один пробел. И вывел код примера в виде обычного абзаца, который у него поместился в одну строку.

Web-обозреватель все сделал правильно! Просто мы не указали ему, как следует выводить данный текст.

Специально для случаев, когда текст должен быть выведен именно так, как набран, с сохранением всех пробелов и переносов строк, язык HTML предусматривает парный тег `<PRE>`. Выводимый текст помещают внутри этого тега (листинг 2.11).

#### Листинг 2.11

```
<PRE>Я — текст,  
который будет выведен  
на Web-страницу  
со всеми  
    отступами  
и  
переносами  
строк.</PRE>
```

Такой текст называется текстом *фиксированного формата*.

Правила отображения текста фиксированного формата:

- ❑ для вывода используется моноширинный шрифт (у *моноширинного шрифта* все символы имеют одинаковую ширину, в отличие от *пропорциональных*, у которых ширина символов различна);
- ❑ все пробелы и переносы строк сохраняются при выводе (это мы уже знаем);
- ❑ если строка текста фиксированного формата не помещается в окне Web-обозревателя по ширине, она ни в коем случае не будет переноситься. Из-за этого может возникнуть потребность в горизонтальной прокрутке Web-страницы (что, вообще-то, плохой стиль Web-дизайна...);
- ❑ возможно наличие HTML-тегов для выделения текста и гиперссылок

Текст фиксированного формата также является блочным элементом.

Исправим HTML-код Web-страницы `t_title.htm` так, чтобы пример выводился в виде текста фиксированного формата (листинг 2.12).

#### Листинг 2.12

```
<!DOCTYPE html>  
<HTML>
```

. . .

```
<H6>Пример:</H6>
<PRE>!HEAD!
!TITLE!Я – заголовок Web-страницы!/TITLE!
!HEAD!</PRE>
</BODY>
</HTML>
```

Откроем исправленную Web-страницу в Web-обозревателе и убедимся, что он выводится правильно (рис. 2.2).

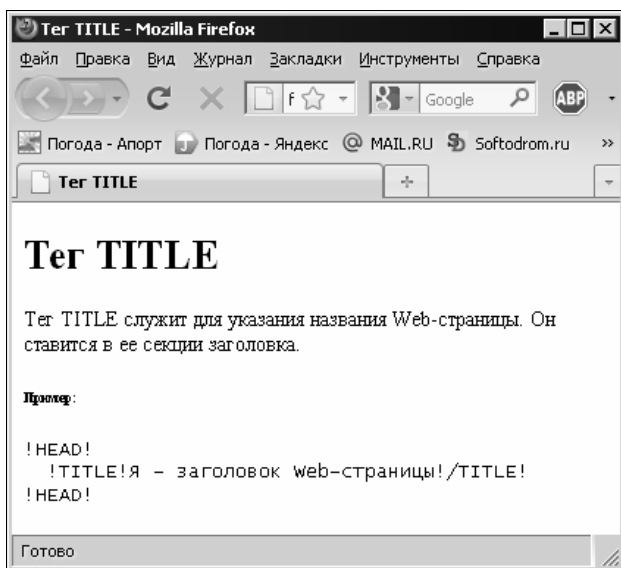


Рис. 2.2. Исправленная Web-страница t\_title.htm в Web-обозревателе. Код примера оформлен как текст фиксированного формата

Как правило, текст фиксированного формата используется для вывода исходных текстов программ и быстрой публикации в Сети документов, набранных обычным текстом. В качестве примера можно вспомнить библиотеку Мошкова (<http://www.lib.ru>), в которой все книги опубликованы как раз в виде текста фиксированного формата.

## Горизонтальные линии

Что бы нам еще такое сделать с Web-страницами... Давайте дополнительно выделим на главной Web-странице index.htm цитату из Википедии, описывающую HTML.

Для любителей все выделять HTML припас подарок — горизонтальную линию, создаваемую с помощью одинарного тега `<HR>`:

```
<P>Я буду отделен от следующего абзаца горизонтальной линией.</P>
<HR>
<P>Я отделен от предыдущего абзаца горизонтальной линией.</P>
```

Горизонтальная линия HTML растягивается по горизонтали на всю ширину Web-страницы, имеет один-два пиксела в толщину и выпуклый или вдавленный вид (конкретные параметры зависят от Web-обозревателя). Она применяется для отделения одной части содержимого Web-страницы от другой и просто "для красоты". Однако нужно помнить, что слишком большое число горизонтальных линий — дурной тон Web-дизайна.

Больше о горизонтальных линиях рассказывать нечего. Так что внесем в HTML-код страницы index.htm необходимые исправления (листинг 2.13).

### Листинг 2.13

```
<P>Приветствуем на нашем Web-сайте всех, кто занимается Web-дизайном!  
Здесь вы сможете найти информацию обо всех интернет-технологиях,  
применяемых при создании Web-страниц. А именно, о языках HTML и  
CSS.</P>  
<HR>  
<P>Русская Википедия определяет термин HTML так:</P>  
. . .  
<P>Пожалуй, ни убавить ни прибавить...</P>  
<HR>  
<P>HTML позволяет формировать на Web-страницах следующие элементы:</P>
```

Результат показан на рис. 2.3. Симпатично вышло, не так ли?

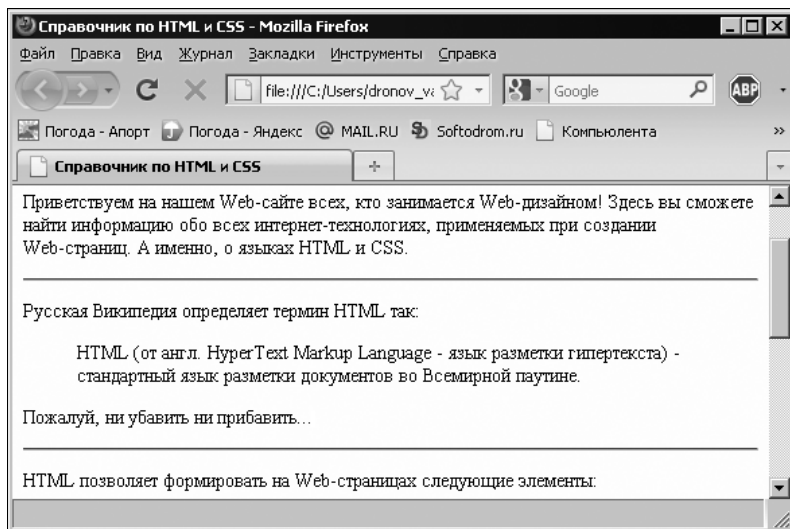


Рис. 2.3. Web-страница index.htm с горизонтальными линиями

## Адреса

Часто на Web-страницах указывают контактные данные их создателей (почтовые и электронные адреса, телефоны, факсы и пр.). Для этого HTML предусматривает

особый тег <ADDRESS>. Он ведет себя так же, как тег абзаца <P>, но его содержимое выводится курсивом:

```
<ADDRESS>Я – адрес создателя данной Web-страницы: почтовый, электронный,  
телефоны и факсы.</ADDRESS>
```

## Комментарии

Напоследок рассмотрим одну очень важную возможность HTML, которая, хоть и не касается напрямую Web-дизайна, но сильно поможет забывчивым Web-дизайнерам.

*Комментарий* — это фрагмент HTML-кода, который не выводится на Web-страницу и вообще не обрабатывается Web-обозревателем. Он служит для того, чтобы Web-дизайнер смог оставить текстовые заметки для себя или своих коллег.

Текст комментария помещают между открывающим тегом <!-- и закрывающим тегом --> и обязательно отделяют от этих тегов пробелами. Как видим, теги комментария не укладываются в основное правило HTML: закрывающий тег должен иметь то же имя, что и открывающий. Открывающий и закрывающий теги комментария — разные!

Пример:

```
<!-- Я – комментарий. Меня не видно на Web-странице. -->
```

Мы можем создать в HTML-коде наших Web-страниц комментарии, указывающие, что нам следует доработать. Хотя бы просто для практики:

```
<!-- Выделить важные фрагменты текста и доделать код примеров. -->
```

```
<!-- Создать Web-страницы, посвященные остальным тегам HTML. -->
```