

## Глава 3. Форматирование текста

Язык HTML имеет широкие возможности по форматированию текста. Правильно структурированный текст дает возможность пользователям быстро и комфортно читать информацию с экрана монитора.

При наборе текста для HTML-документа следует избегать переносов слов по слогам и выравнивания текста с помощью пробелов или табуляций. Все форматирование в HTML следует осуществлять через элементы.

### 3.1. Пробельные символы

Наборы символов документа содержат большое количество разных пробельных символов. Многие из них являются типографскими элементами, используемыми в некоторых приложениях для создания особых визуальных эффектов. В HTML как пробельные символы определены только:

ASCII space - пробел (&#x0020;)

ASCII tab - табуляция (&#x0009;)

ASCII form feed - прогон страницы (&#x000C;)

Zero-width space -нулевой пробел (&#x200B;)

Переносы строк (перевод строки, запрет перевода строки) также являются пробельными символами.

В языках с латиницей и в русском языке, межсловное пространство обычно представлено как пробел ASCII (&#x0020;).

Обычно браузеры сжимают полученные на вводе последовательности пробелов при создании межсловного пространства на выходе, т.е. при употреблении нескольких пробелов между словами, они визуально сжимаются в автоматически один.

#### 3.1.1. Перевод строки

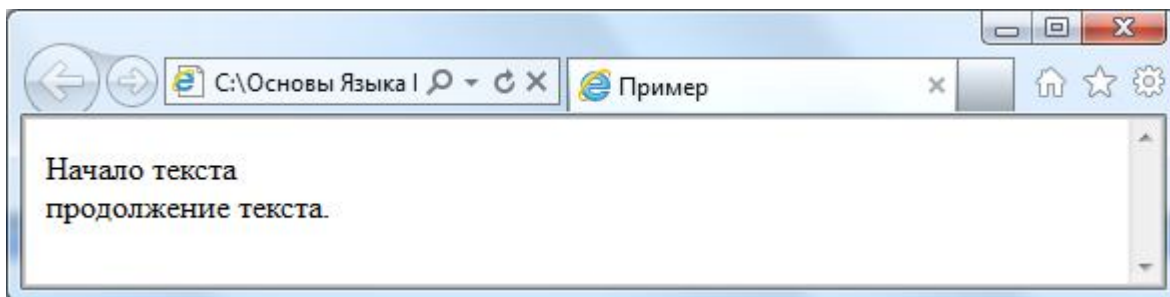
Браузер автоматически переводит строку текста по достижении правой границы окна. Однако, иногда требуется принудительно осуществить перенос части текста на новую строку. Для этого служит специальный элемент.

Перевод строки определён как символ возврата каретки (&#x000D;), символ прогона строки (&#x000A;) или как пара символов возврат каретки/прогон строки.

Синтаксис
<i>Начало текста</i> <b>&lt;br&gt;</b> <i>продолжение текста</i>

```
Содержимое файла sample3.1.1-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    Начало текста<br>продолжение текста.
  </body>
</html>
```

Результат на экране:



### 3.1.2. Запрет перевода строки

Для запрета автоматического перевода строки браузером служит элемент **no**br.

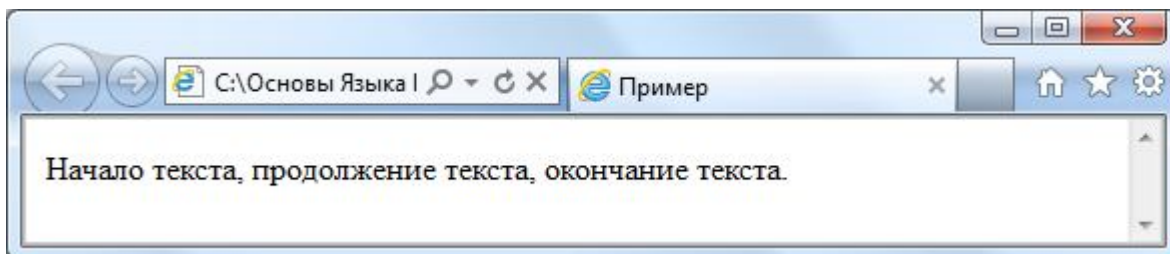
Синтаксис

*Начало текста***<no**br**>***продолжение текста***</no**br**>***окончание текста*

Содержимое файла sample3.1.2-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    Начало текста, <no
```

Результат на экране:



### 3.2. Горизонтальная линия

Для отделения одной части текста от другой горизонтальной линией необходимо использовать элемент **hr**.

Синтаксис

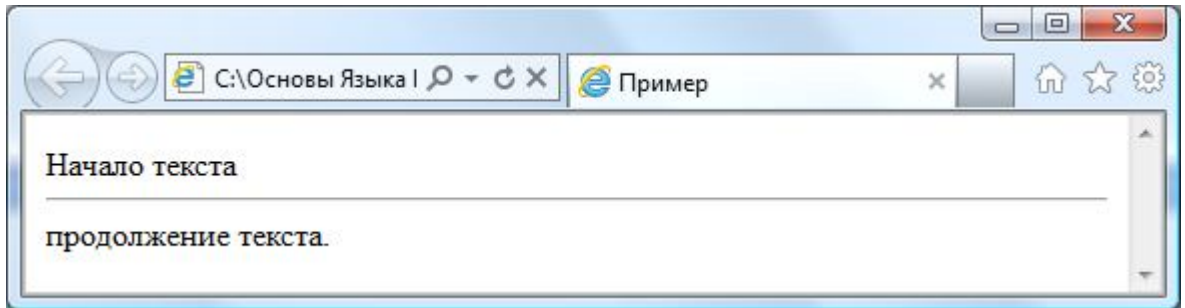
*Начало текста***<hr****>***окончание текста*

Содержимое файла sample3.2-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
```

```
<body>
Начало текста<hr>продолжение текста.
</body>
</html>
```

Результат на экране:



### 3.3. Предварительное форматирование текста

В отдельных случаях на странице необходимо сохранить форматирование текста, созданное текстовым редактором в неизменном виде. Для реализации такой возможности существует элемент `pre`. Браузер не будет дополнительно обрабатывать эту часть текста. Особенностью данного элемента является то, что текст выводится моноширинным шрифтом (шрифтом фиксированного размера).

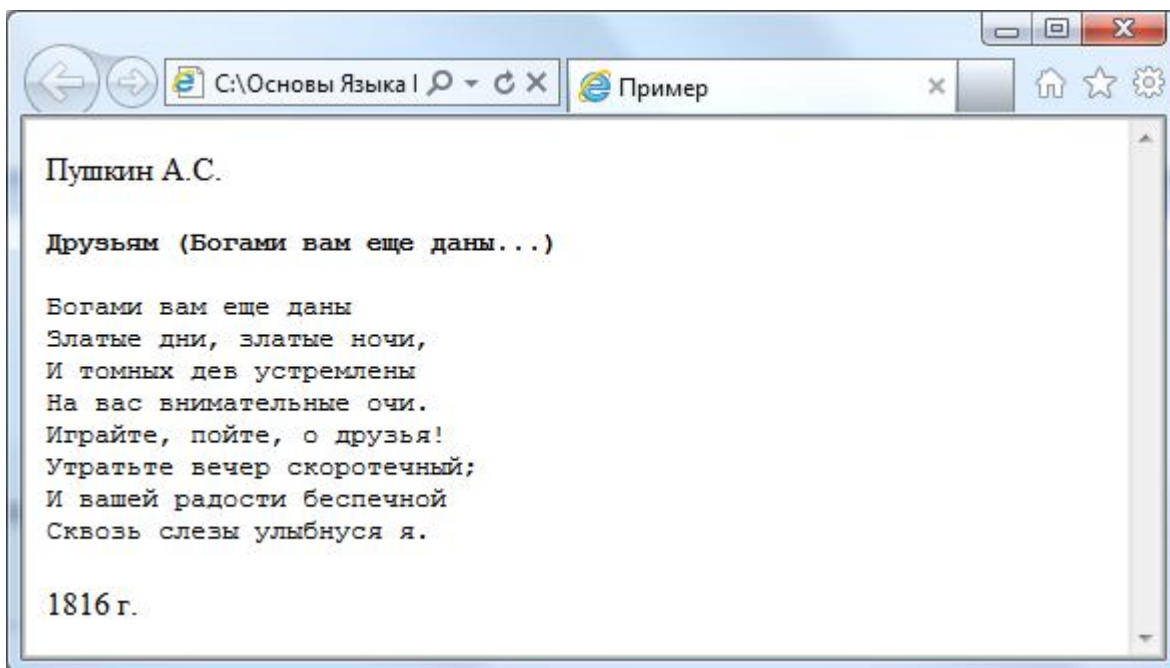
Синтаксис

```
<pre>
текст
</pre>
```

Содержимое файла sample3.3-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    Пушкин А.С.
    <pre>
    <b>Друзьям (Богами вам еще даны...)</b><br>
    Богами вам еще даны
    Златые дни, златые ночи,
    И томных дев устремлены
    На вас внимательные очи.
    Играйте, пойте, о друзья!
    Утратьте вечер скоротечный;
    И вашей радости беспечной
    Сквозь слезы улыбнуся я.
    </pre>
    1816 г.
  </body>
</html>
```

Результат на экране:



### 3.4. Выделение блока

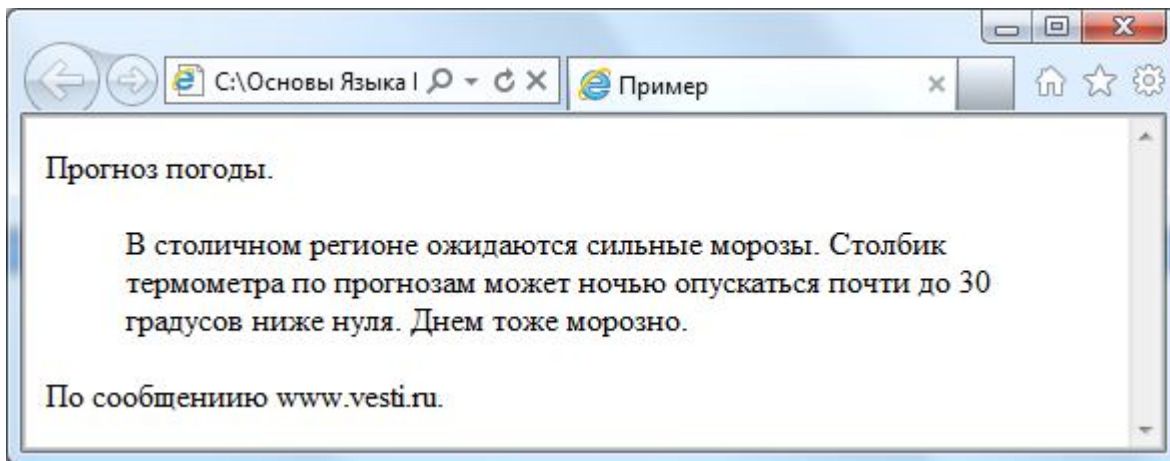
Часть текста на странице можно выделить отдельным блоком, все строки которого будут отображаться с некоторым отступом от левого края. Размер отступа зависит от браузера.

Синтаксис
<pre>&lt;blockquote cite="адрес"&gt; текст &lt;/blockquote&gt;</pre>

Атрибут **cite** обозначает источник документа или сообщения. Этот атрибут предназначается для сообщения информации об источнике, из которого заимствована информация. Значение атрибута не отображается браузером.

```
Содержимое файла sample3.4-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    Прогноз погоды.
    <blockquote cite="http://www.vesti.ru/doc.html?id=697522&cid=7">
    В столичном регионе ожидаются сильные морозы. Столбик термометра по прогнозам может ночью опускаться почти до 30 градусов ниже нуля. Днем тоже морозно.
    </blockquote>
    По сообщению www.vesti.ru.
  </body>
</html>
```

Результат на экране:



### 3.5. Заголовки

Заголовок кратко описывает содержимое фрагмента страницы, которому он предшествует. Заголовки отображаются жирным шрифтом.

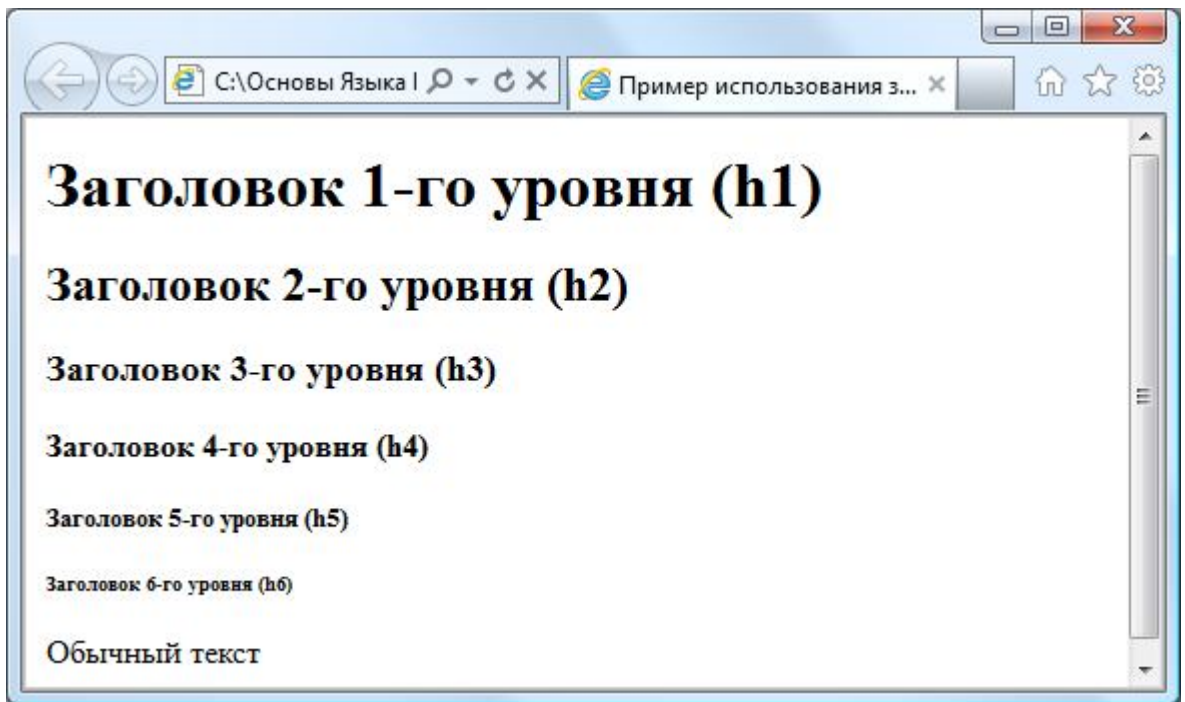
Заголовки имеют 6 уровней с 1-го по 6-й и различаются только размером шрифта.

Синтаксис
<code>&lt;h<math>n</math>&gt;текст заголовка&lt;/h<math>n</math>&gt;</code> , где $n$ - номер уровня с 1 по 6.

Содержимое файла sample3.5-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Пример использования заголовков</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <h1>Заголовок 1-го уровня (h1)</h1>
    <h2>Заголовок 2-го уровня (h2)</h2>
    <h3>Заголовок 3-го уровня (h3)</h3>
    <h4>Заголовок 4-го уровня (h4)</h4>
    <h5>Заголовок 5-го уровня (h5)</h5>
    <h6>Заголовок 6-го уровня (h6)</h6>
    Обычный текст
  </body>
</html>
```

Результат на экране:



### 3.6. Параграф

Основной структурой текста в HTML является параграф (абзац). Каждый параграф начинается с новой строки. Для выделения абзаца служит элемент `p`. В элементе можно указать способ выравнивания текста.

Синтаксис
<code>&lt;p align="выравнивание"&gt;текст параграфа&lt;/p&gt;</code>

выравнивание	Пояснение
<code>left</code>	выравнивание текста по левому краю
<code>center</code>	выравнивание текста по центру
<code>right</code>	выравнивание текста по правому краю
<code>justify</code>	выравнивание текста по ширине

Таблица 3. Выравнивание текста

Параграф не может быть вложенным, т.е. нельзя использовать элемент для создания подпараграфа.

Особенности использования атрибута выравнивания заметны при изменении размера окна браузера.

Пример:

```

Содержимое файла sample3.6-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Форматирование текста. Параграф.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p>Это простой параграф без Атрибутов. Может занимать несколько строк текста при отображении в браузере. По умолчанию выравнивание происходит по левому краю.</p>
  </body>
</html>

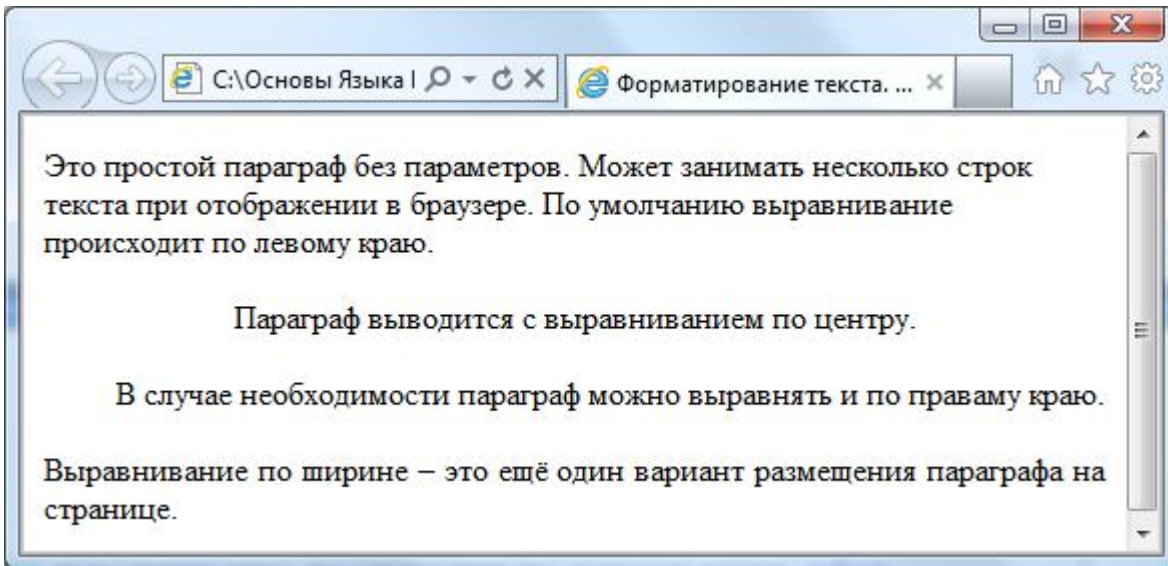
```

```

<p align="center">Параграф выводится с выравниванием по центру.</p>
<p align="right">В случае необходимости параграф можно выравнивать и по правому
краю.</p>
<p align="justify">Выравнивание по ширине – это ещё один вариант размещения параграфа
на странице.</p>
</body>
</html>

```

Результат на экране:



### 3.7. Увеличенный шрифт

Текст внутри элемента будет отображаться больше на 1 пункт, чем основной шрифт.

Синтаксис
<code>&lt;big&gt;текст&lt;/big&gt;</code>

### 3.8. Уменьшенный шрифт

Текст внутри элемента будет отображаться меньше на 1 пункт, чем основной шрифт.

Синтаксис
<code>&lt;small&gt;текст&lt;/small&gt;</code>

### 3.9. Полужирный шрифт

Текст внутри элемента будет отображаться полужирным шрифтом.

Синтаксис
<code>&lt;strong&gt;текст&lt;/strong&gt;</code>

или устаревший вариант:

Синтаксис
<code>&lt;b&gt;текст&lt;/b&gt;</code>

### 3.10. Наклонный шрифт (курсив)

Текст внутри элемента будет отображаться курсивом.

Синтаксис
<code>&lt;em&gt;текст&lt;/em&gt;</code>

или устаревший вариант:

Синтаксис
<code>&lt;i&gt;текст&lt;/i&gt;</code>

### 3.11. Подчеркнутый шрифт

Текст внутри элемента будет подчеркнут.

Синтаксис
<code>&lt;u&gt;текст&lt;/u&gt;</code>

### 3.12. Перечеркнутый шрифт

Текст внутри элемента будет перечеркнут.

Синтаксис
<code>&lt;s&gt;текст&lt;/s&gt;</code>

### 3.13. Верхний индекс (надиндекс)

Текст внутри элемента будет отображаться верхним индексом.

Синтаксис
<code>&lt;sup&gt;текст&lt;/sup&gt;</code>

### 3.14. Нижний индекс (подиндекс)

Текст внутри элемента будет отображаться нижним индексом.

Синтаксис
<code>&lt;sub&gt;текст&lt;/sub&gt;</code>

### 3.15. Компьютерный код

Обозначает фрагмент компьютерного кода.

Синтаксис
<code>&lt;code&gt;текст&lt;/code&gt;</code>

### 3.16. Образец

Образец вывода программ, скриптов и т.п.

Синтаксис
<code>&lt;samp&gt;текст&lt;/samp&gt;</code>

### 3.17. Аргумент переменной или программы.

Аргумент переменной или программы.

Синтаксис
-----------



```
<var>текст</var>
```

### 3.18. Пользовательский текст

Обозначает текст, введённый пользователем.

Синтаксис

```
<kbd>текст</kbd>
```

### 3.19. Определение термина

Определение термина.

Синтаксис

```
<dfn>текст</dfn>
```

### 3.20. Цитата

Содержит цитату или ссылку на другой ресурс.

Синтаксис

```
<cite>текст</cite>
```

### 3.21. Сокращение, аббревиатура

Сокращение, аббревиатура (напр., WWW, HTTP и т.д.).

Синтаксис

```
<abbr title="описание">текст</abbr>
```

### 3.22. Акроним

Акроним (например, БАМ, бейсик, лазер, НИИ и т.п.).

Синтаксис

```
<acronym title="описание">текст</acronym>
```

В западных языках акронимы широко используются: "GmbH", "NATO" и "F.B.I.", также как и аббревиатуры: "M.", "Inc.", "et al.", "etc."

Содержимое элементов `abbr` и `acronym` определяет само сокращённое выражение так, как оно нормально выглядит в тексте. Атрибут `title` этих элементов может использоваться для предоставления полной формы этих выражений.

Пример:

Содержимое файла sample3.22-1.html:

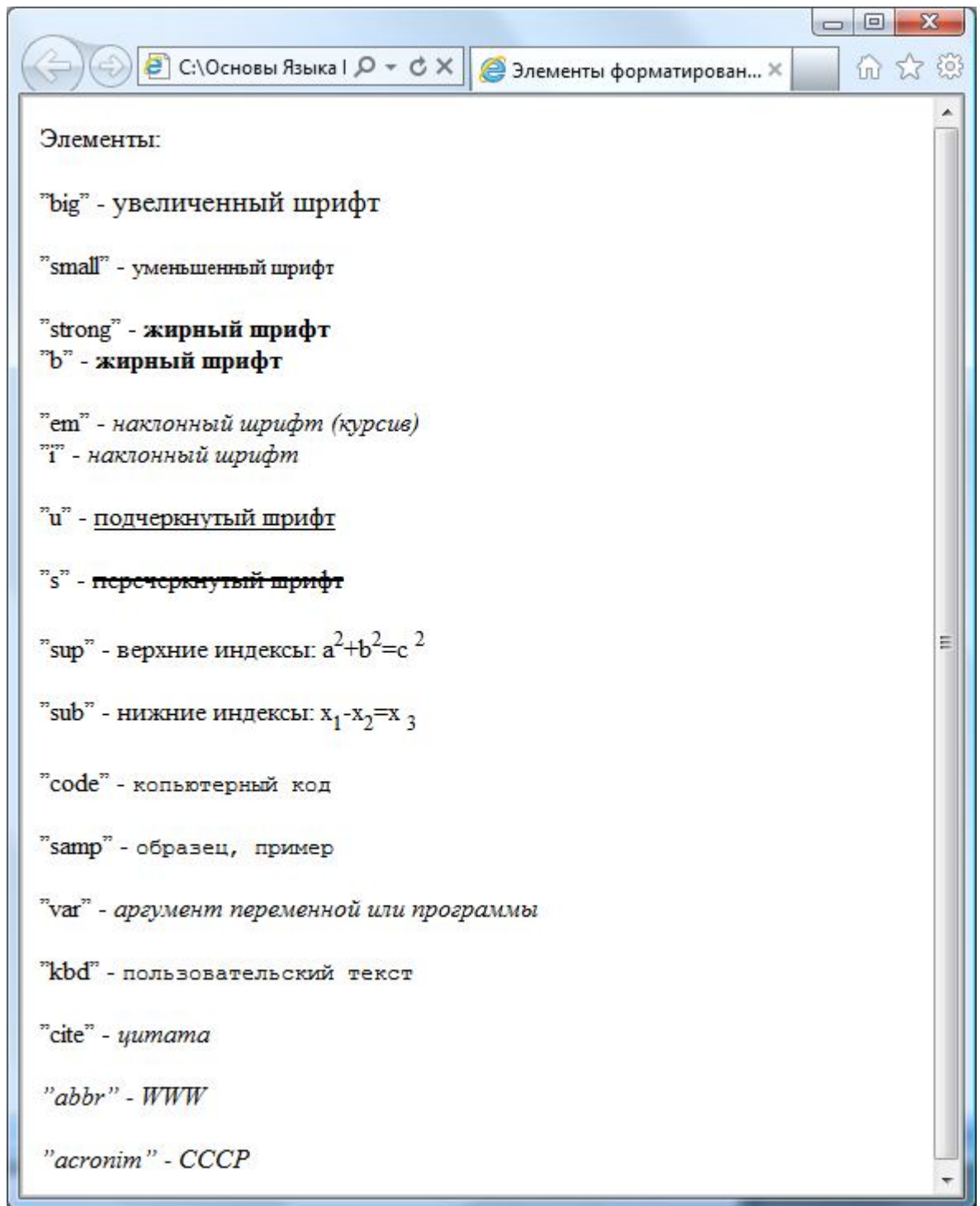
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Элементы форматирования.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p>Элементы:</p>
    <p>"big" - <big>увеличенный шрифт</big></p>
```

```

<p>"small" - <small>уменьшенный шрифт</small></p>
<p>"strong" - <strong>жирный шрифт</strong><br>"b" - <b>жирный шрифт</b></p>
<p>"em" - <em>наклонный шрифт (курсив)</em><br>"i" - <i>наклонный шрифт</i></p>
<p>"u" - <u>подчеркнутый шрифт</u></p>
<p>"s" - <s>перечеркнутый шрифт</s></p>
<p>"sup" - верхние индексы: a<sup>2</sup>+b<sup>2</sup>=c <sup>2</sup></p>
<p>"sub" - нижние индексы: x<sub>1</sub>-x<sub>2</sub>=x <sub>3</sub></p>
<p>"code" - <code>компьютерный код</code></p>
<p>"samp" - <samp>образец, пример</samp></p>
<p>"var" - <var>аргумент переменной или программы</var></p>
<p>"kbd" - <kbd>пользовательский текст</kbd></p>
<p>"cite" - <dfn>цитата</cite></p>
<p>"abbr" - <abbr>WWW</abbr></p>
<p>"acronym" - <acronym>СССР</acronym></p>
</body>
</html>

```

Результат на экране:



### 3.23. Тип, размер и цвет шрифта

Размер шрифта на странице устанавливается браузером по умолчанию. Элемент **font** позволяет изменить размер, цвет и тип шрифта.

Синтаксис

```
<font face="шрифт" size="размер" color="цвет">текст</font>
```

Атрибут **face** может не поддерживаться некоторыми браузерами. Следующие виды шрифтов, типичные на ПК на основе Windows:

шрифт	Пояснение
-------	-----------

Times New Roman	шрифт Times New Roman
Arial Narrow	шрифт Arial Narrow
Arial	шрифт Arial
Courier New	шрифт Courier New
Tahoma	шрифт Tahoma
Comic Sans MS	шрифт Comic Sans MS
Verdana	шрифт Verdana

Таблица 4. Стандартные шрифты

Можно указать несколько шрифтов через запятую - в этом случае для форматирования текста будет использован первый подходящий шрифт из списка, например: "Comic Sans MS, Tahoma". Таким образом, если у пользователя на компьютере нет шрифта Comic Sans MS, браузер подставит следующий в этом списке - Tahoma.

Использовать необходимо шрифты, которые установлены на компьютере пользователя, в противном случае браузер будет использовать шрифт, определенный по умолчанию (обычно Times New Roman). К стандартным шрифтам можно отнести шрифты, поставляемые с MS Windows, MS Plus, MS Office.

Атрибут **size** определяет размер шрифта, начиная с цифры 1 по цифру 7 либо целое со знаком, например, "+1" или "-2". Число со знаком добавляется к текущему базовому размеру шрифта.

Атрибут **color** определяет цвет шрифта. По умолчанию черный - "#000000" (black). Цвет может быть представлен как символьной нотацией, например, red, blue и т.д, так и представлен в шестнадцатичном виде - "#FF0000" (красный), "#0000FF" (синий), при этом обозначение должно быть взято в кавычки.

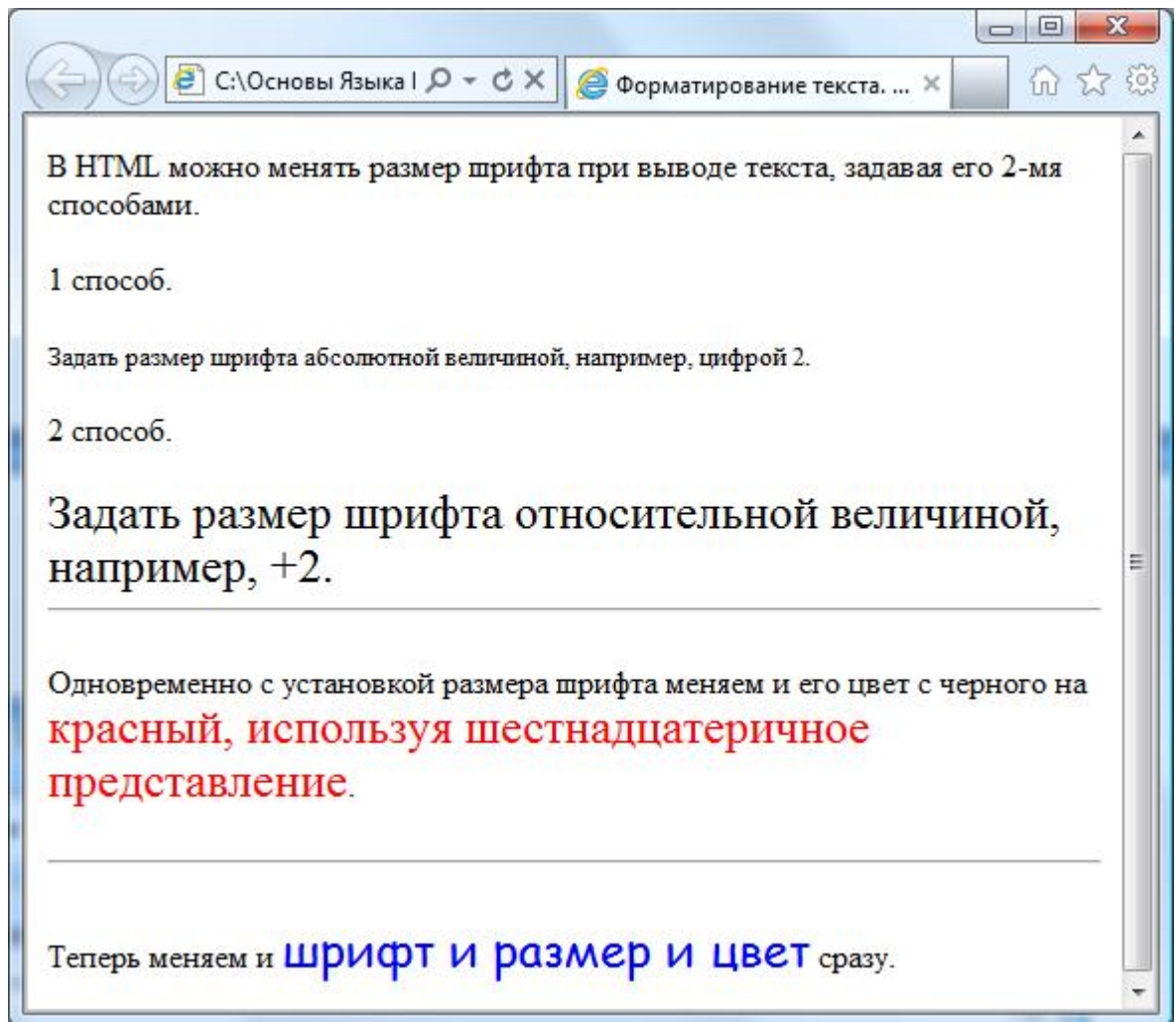
Пример:

```

Содержимое файла sample3.23-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title> Форматирование текста. Параграф. Размер и цвет шрифта.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p>В HTML можно менять размер шрифта при выводе текста, задавая его 2-мя способами.</p>
    <p>1 способ.</p>
    <font size="2">Задать размер шрифта абсолютной величиной, например, цифрой 2.</font>
    <p>2 способ.</p>
    <font size="+2">Задать размер шрифта относительной величиной, например, +2.</font>
    <hr>
    <p>Одновременно с установкой размера шрифта меняем и его цвет с черного на <font size="+2" color="#FF0000">красный, используя шестнадцатичное представление.</font>.</p>
    <hr>
    <p>Теперь меняем и <font face="Comic Sans MS" size="+2" color="#0000FF">шрифт и размер и цвет.</font> сразу.</p>
  </body>
</html>

```

Результат на экране:



### 3.24. Списки

В языке HTML поддерживаются упорядоченные списки, неупорядоченные списки и списки определений. Отличаются эти разновидности списков лишь способом оформления. Перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера или символы латинского алфавита.

В HTML 4 во всех типах списков не рекомендуется использовать атрибуты.

#### 3.24.1. Маркированные (Неупорядоченные) списки

Маркированный список начинается с элемента `ul`. Каждый новый пункт списка начинается с элемента `li`.

Синтаксис
<pre>&lt;ul compact type=тип_маркеров&gt;   &lt;li&gt;пункт 1   &lt;li&gt;пункт 2   ...   &lt;li&gt;пункт N &lt;/ul&gt;</pre>

Атрибут `compact` делает отображение списка более компактным.

Атрибут `type` определяет тип маркеров для списка:

тип маркеров	Пояснение
circle	○ - окружность
disc	● - диск (круг)
square	▪ - квадрат

Таблица 5. Типы маркеров списка

В качестве элемента списка могут служить параграфы, переносы строк, изображения, ссылки, другие списки, и т.д.

Не рекомендуется использовать атрибут **type**, если планируется использовать вложенные списки. Следует воспользоваться возможностями браузеров заданных по умолчанию.

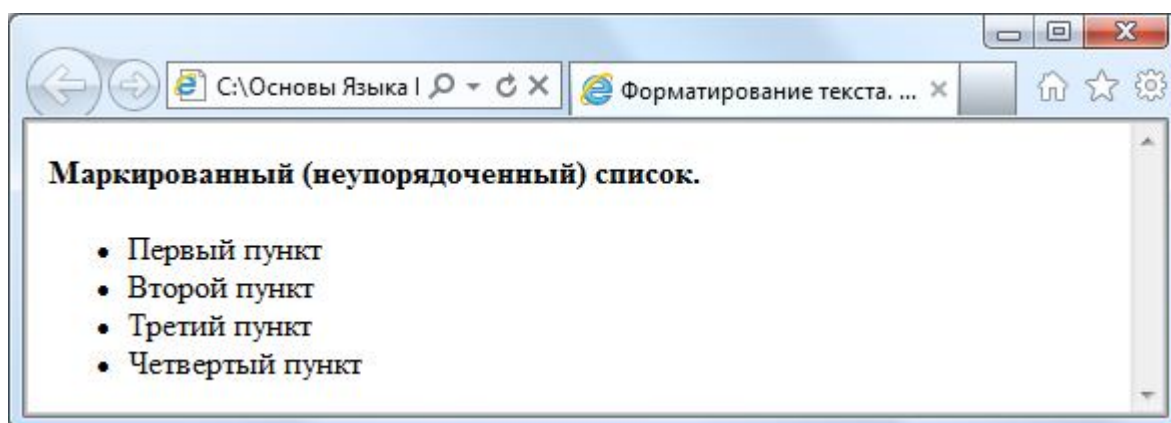
Пример:

```

Содержимое файла sample3.24.1-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Форматирование текста. Списки.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p><b>Маркированный (неупорядоченный) список.</b></p>
    <ul type=disc>
      <li>Первый пункт
    <li>Второй пункт
    <li>Третий пункт
    <li>Четвертый пункт
    </ul>
  </body>
</html>

```

Результат на экране:



### 3.24.2. Нумерованные (упорядоченные) списки

Элементы списка маркируются с помощью чисел или букв. Начинается с элемента **o1**. Каждый элемент списка начинается с элемента **li**.

Синтаксис

```

<ol compact type="стиль_нумерации" start="число">
  <li>пункт 1
  <li>пункт 2
  ...
  <li>пункт N
</ol>

```

Атрибут **compact** делает отображение списка более компактным.

Атрибут **start** определяет первое число или символ, с которого начинается нумерация пунктов.

Атрибут **type** определяет стиль нумерации:

стиль_нумерации	Пояснение
"A"	заглавные буквы A, B, C ...
"a"	строчные буквы a, b, c ...
"I"	большие римские числа I, II, III ...
"i"	маленькие римские числа i, ii, iii ...
"1"	арабские числа 1, 2, 3 ...

Таблица 6. Стили нумерации

В качестве элемента списка могут служить параграфы, переносы строк, изображения, ссылки, другие списки, и т.д.

Не рекомендуется использовать атрибут **type** и атрибут **start**, если планируется использовать вложенные списки. Следует воспользоваться возможностями браузеров заданных по умолчанию.

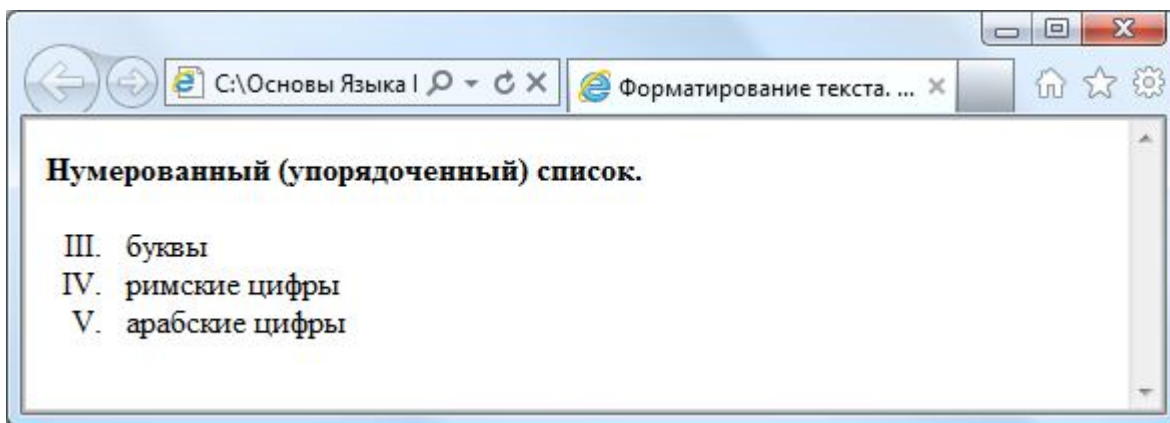
Пример:

```

Содержимое файла sample3.24.2-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Форматирование текста. Списки.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p><b>Нумерованный (упорядоченный) список.</b></p>
    <ol type="I" start="3">
      <li>буквы
      <li>римские цифры
      <li>арабские цифры
    </ol>
  </body>
</html>

```

Результат на экране:



### 3.24.3. Списки определений

Список определений начинается с элемента **dl**. Перед каждым термином присутствует элемент **dt**. После термина указывается элемент **dd**, после которого следует определение термина.

Синтаксис
<pre> &lt;dl compact&gt;   &lt;dt&gt;термин 1     &lt;dd&gt;определение термина 1     &lt;dd&gt;другое определение термина 1   &lt;dt&gt;термин 2     &lt;dd&gt;определение термина 2     &lt;dd&gt;другое определение термина 2   ...   &lt;dt&gt;термин N     &lt;dd&gt;определение термина N     &lt;dd&gt;другое определение термина N &lt;/dl&gt; </pre>

Атрибут **compact** делает отображение списка более компактным.

Пример:

```

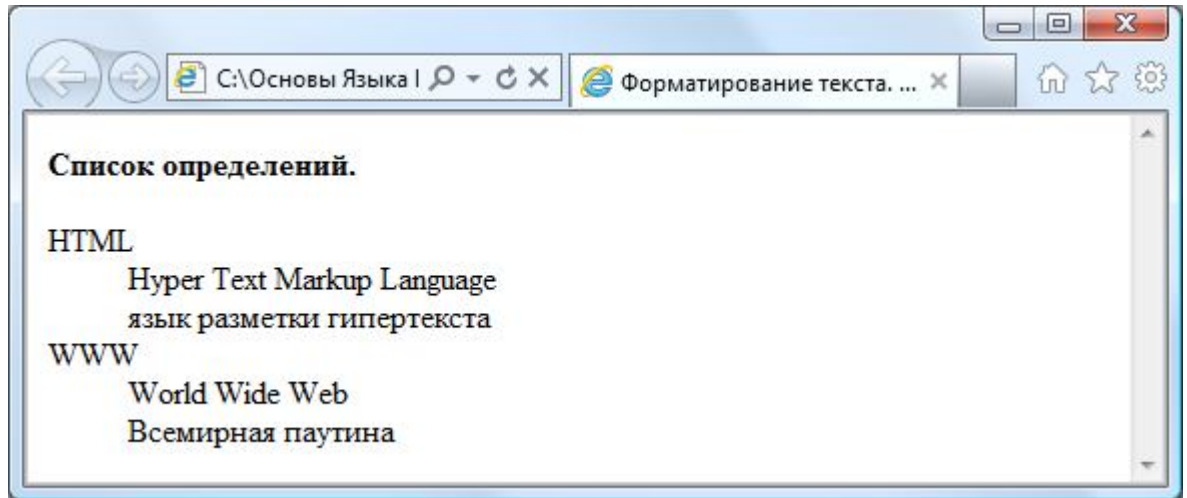
Содержимое файла sample3.24.3-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Форматирование текста. Списки.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p><b>Список определений.</b></p>
    <dl compact>
      <dt>HTML
        <dd>Hyper Text Markup Language
        <dd>язык разметки гипертекста
      <dt>WWW
        <dd>World Wide Web
        <dd>Всемирная паутина
    </dl>

```



```
</body>  
</html>
```

Результат на экране:



### 3.24.4. Вложенные списки

Элемент любого списка может содержать в себе целый список любого вида. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками все же не следует.

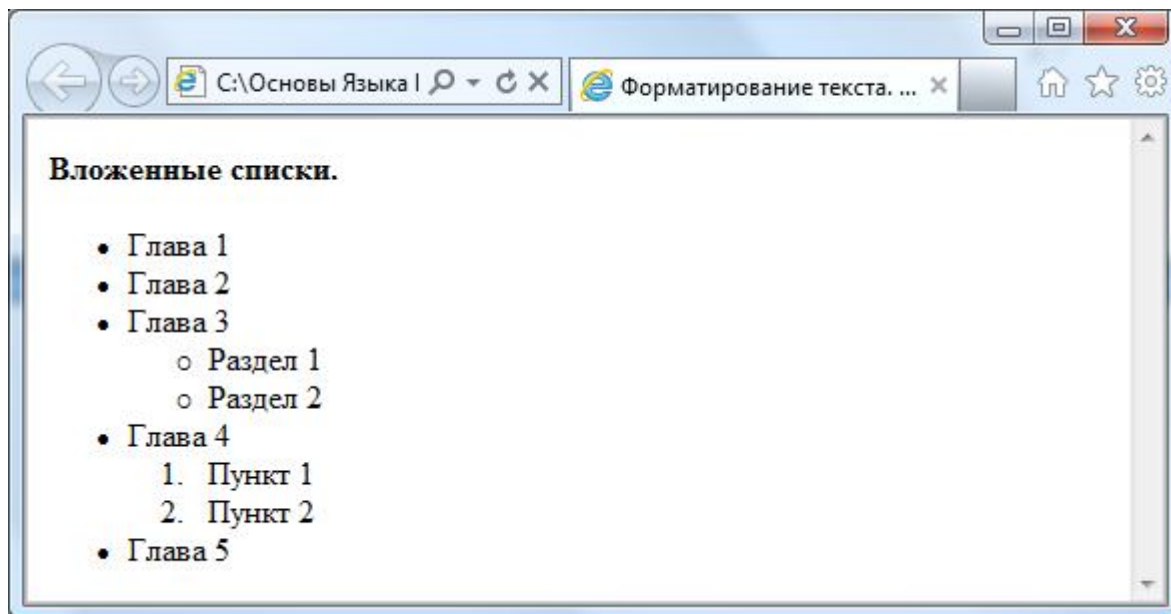
Вложенные списки очень удобны при подготовке разного рода планов и оглавлений.

Пример:

Содержимое файла sample3.24.4-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
  <head>  
    <title>Форматирование текста. Списки.</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">  
  </head>  
<body>  
<p><b>Вложенные списки.</b></p>  
<ul>  
  <li>Глава 1  
  <li>Глава 2  
  <li>Глава 3  
  <ul>  
    <li>Раздел 1  
    <li>Раздел 2  
  </ul>  
  <li>Глава 4  
  <ol>  
    <li>Пункт 1  
    <li>Пункт 2  
  </ol>  
  <li>Глава 5  
</ul>  
</body>  
</html>
```

Результат на экране:



### 3.24.5. Графические элементы при оформлении списков

В качестве маркеров списка можно использовать графические изображения для создания привлекательных, красиво оформленных HTML-документов. Такая возможность не предоставляется непосредственно языком HTML, а реализуется искусственно.

Как реализовать такую возможность?

Элемент списка выполняет единственную задачу - указывает браузеру, что вся информация, располагаемая после данного элемента должна отображаться с отступом. Элементы `li`, указывающие на отдельные элементы списка, обеспечивают вывод стандартных маркеров элементов списка.

Если же нам требуется построить список с графическими маркерами, то можно вообще обойтись без элементов `li`. Достаточно будет перед каждым элементом списка вставить желаемое графическое изображение (работа с графическими изображениями подробно будет рассмотрена в одной из глав). Далее необходимо отделение элементов списка друг от друга. Для этого можно использовать элементы абзаца `p` или перевода строки `br`.

Следует заметить, что подобное не рекомендуется делать согласно стандартам HTML. Однако, позволяет продемонстрировать широкие возможности языка при создании HTML-документов.

Пример:

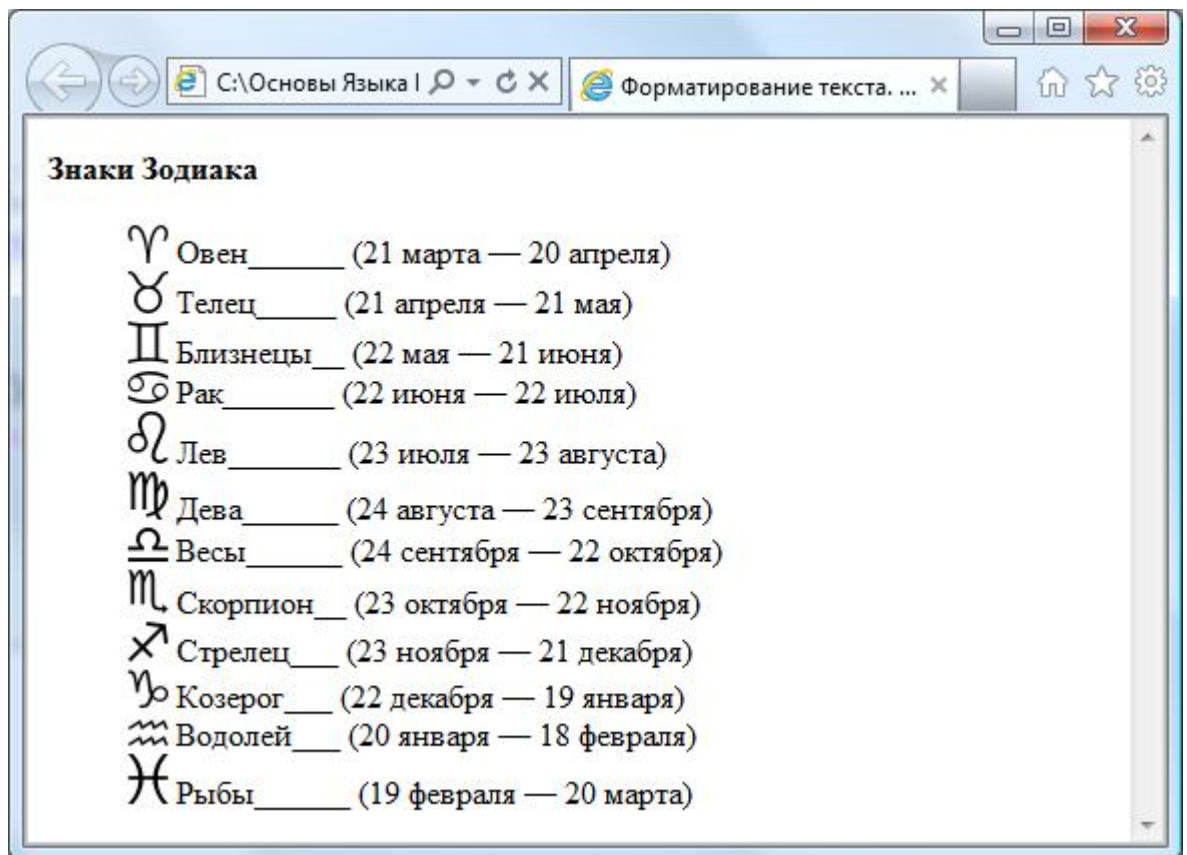
```
Содержимое файла sample3.24.5-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Форматирование текста. Списки.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p><b>Знаки Зодиака</b></p>
    <ul>
      <li> Овен _____ (21 марта — 20 апреля)<br>
      <li> Телец _____ (21 апреля — 21 мая)<br>
```

```

 Близнецы__ (22 мая — 21 июня)<br>
 Рак_____ (22 июня — 22 июля)<br>
 Лев_____ (23 июля — 23 августа)<br>
 Дева_____ (24 августа — 23 сентября)<br>
 Весы_____ (24 сентября — 22 октября)<br>
 Скорпион__ (23 октября — 22 ноября)<br>
 Стрелец__ (23 ноября — 21 декабря)<br>
 Козерог__ (22 декабря — 19 января)<br>
 Водолей__ (20 января — 18 февраля)<br>
 Рыбы_____ (19 февраля — 20 марта)
</ul>
</body>
</html>

```

Результат на экране:



### 3.25. Символьные объекты

Escape-последовательности или, точнее, символьные объекты, представляют собой метод представления специальных символов. Символьная последовательность начинается с символа "&".

Некоторые символы, такие как "<", используются в HTML в специальном значении, поэтому должен существовать способ их выражения, как символьных данных внутри самого документа. Принято использовать следующие нотации (соответствия):

Символ	Пояснение
<	&lt;
>	&gt;
&	&amp;

"	&quot;
©	&copy;
®	&reg;
_	&nbsp; (непрерывный пробел)

Таблица 7. Примеры символьных объектов

Точка с запятой является частью escape-последовательности, но в принципе она может быть опущена. Однако одни браузеры обработают такой код HTML, а другие могут и не понять.

В escape-последовательности важен вариант написания символов, то есть различаются заглавные и строчные буквы. Например, амперсанд (&) не может быть представлен как &AMP (escape-последовательность – не определена).

Необходимость в вышеупомянутых escape-последовательностях проистекает из синтаксиса HTML. На самом деле escape-последовательности существуют для всех символов в стандартном наборе символов ISO 8859-1 Latin 1.

Существуют последовательности вида &#n;, где n - позиционный код символа в десятичном исчислении (в диапазоне от 0 до 255)

Некоторые браузеры поддерживают и другие escape-последовательности, например, &trade;, &shy;, &nbsp;. Использование таких нотаций не рекомендуется. (Нотация &trade; представляет символ, который вообще не принадлежит к ISO 8859-1 Latin 1. Вместо этого можно использовать проверенную нотацию `<sup><small>TM</small></sup>`.

Если нужно вставить в текст один или несколько неизменяемых пробелов, для этого используется сочетание символов &nbsp;.

Пример:

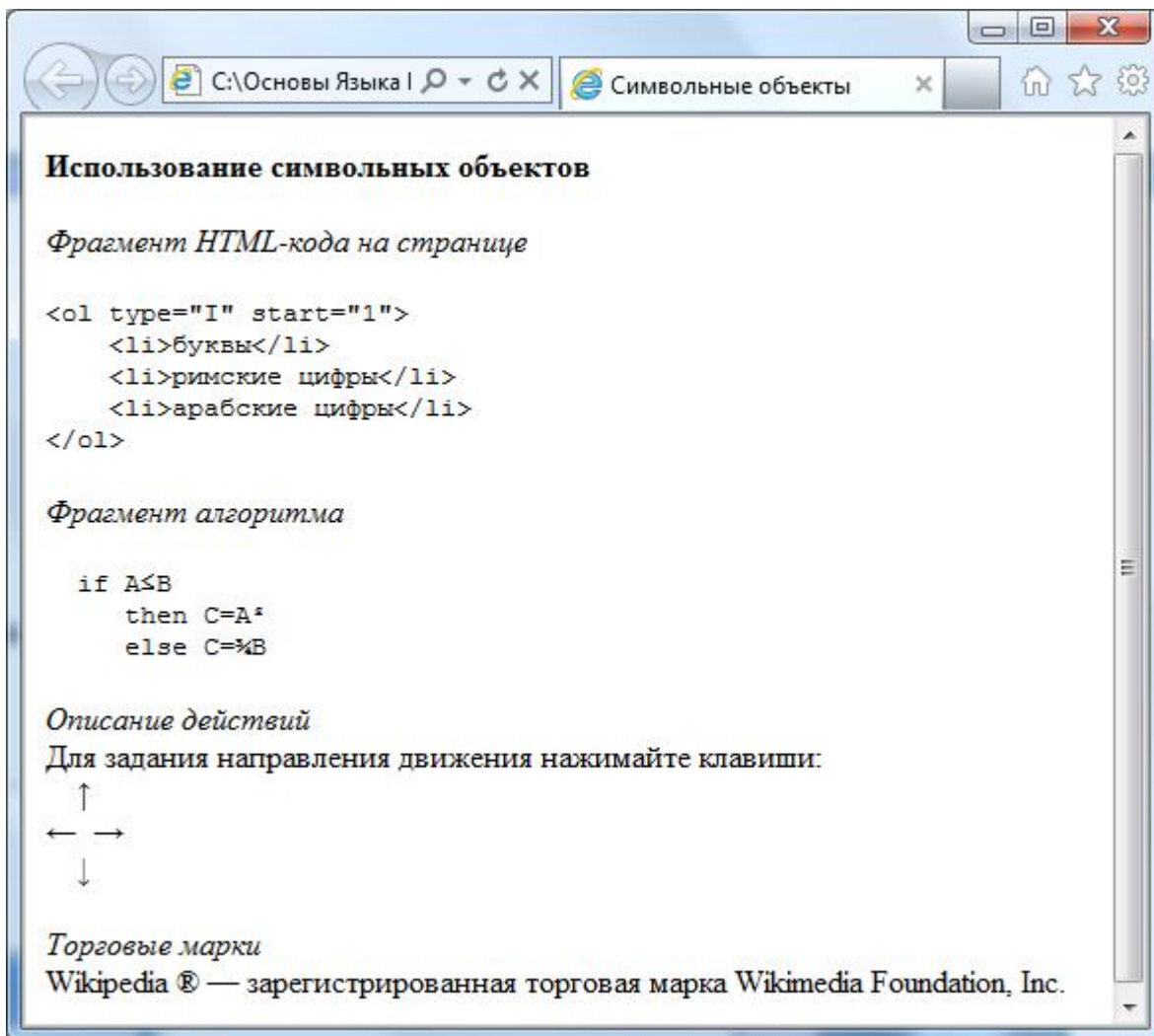
```

Содержимое файла sample3.25-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Символьные объекты</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <p><b>Использование символьных объектов</b></p>
    <p><i>Фрагмент HTML-кода на странице</i></p>
    <pre>
    &lt;ol type="I"&gt;
      &lt;li&gt;буквы&lt;/li&gt;
      &lt;li&gt;римские цифры&lt;/li&gt;
      &lt;li&gt;арабские цифры&lt;/li&gt;
    &lt;/ol&gt;
    </pre>
    <p><i>Фрагмент алгоритма</i></p>
    <pre>
    if A&le;B
      then C=A&sup2;
      else C=&frac34;B
    </pre>
    <p><i>Описание действий</i><br>
    Для задания направления движения нажимайте клавиши:<br>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&uarr;<br>
  </body>
</html>

```

```
&larr;&nbsp;&nbsp;&rarr;<br>
&nbsp;&nbsp;&nbsp;&darr;
</p>
<p><i>Торговые марки</i><br>
Wikipedia &reg; — зарегистрированная торговая марка Wikimedia Foundation, Inc.
</p>
</body>
</html>
```

Результат на экране:



### 3.26. Дефисы

В HTML имеются дефисы двух типов: обычный и "мягкий". Обычный дефис должен интерпретироваться браузерами как обычный символ. Мягкий дефис сообщает браузеру о том, где может появляться обрыв строки.

В HTML обычный дефис представлен символом "-" (&#45; или &#x2D;). Мягкий дефис представляется ссылкой-мнемоникой &shy; (&#173; или &#xAD;).