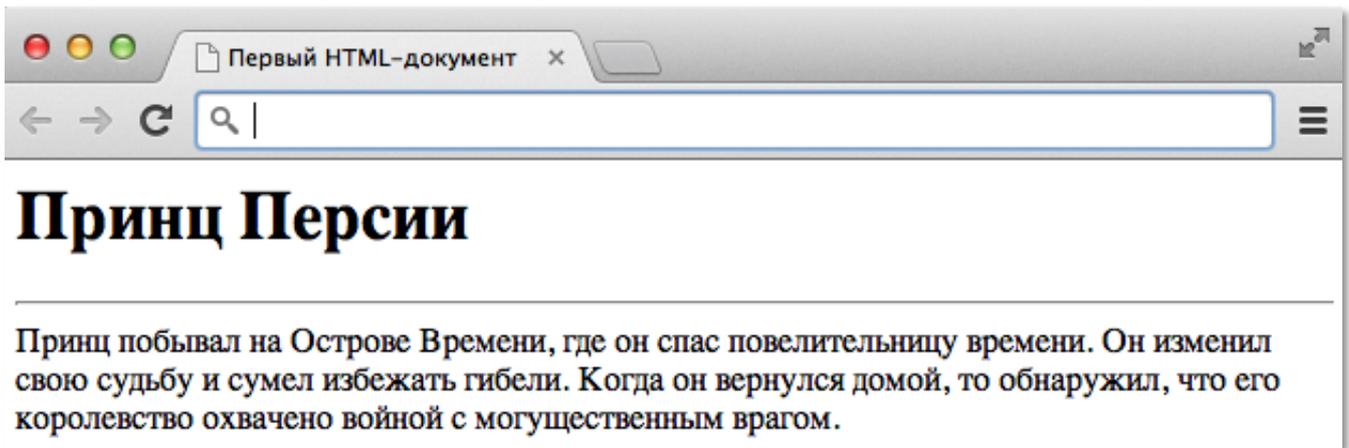


Оглавление

Первый HTML-документ.....	2
Заголовок программы.....	3
Тело программы.....	3
Как работает браузер.....	4
Сохранение файла.....	5
Заголовки в тексте документа.....	5
Горизонтальная линия (разделитель).....	6
Принудительный разрыв строки.....	6
Атрибуты тегов.....	7
Горизонтальная линия (разделитель).....	8
Цвет фона и цвет шрифта.....	9
Изменение размера шрифта.....	11
Начертание шрифта.....	11
Текстовые цитаты.....	12
Авторское форматирование.....	12
Абзац.....	13
Выравнивание абзацев и фрагментов текста.....	13
Маркированный список.....	14
Нумерованный список.....	15
Вложенные списки.....	16
Графика.....	17
Рисунок в тексте.....	17
Имена файлов и ссылки на них.....	18
Переход внутри одного документа.....	20
Переход к другому документу.....	21
Переход к метке другого документа.....	21
Рисунок как ссылка.....	21
Таблицы.....	22
Атрибуты тега <TABLE>.....	22
Атрибуты тега <TR>.....	24
Атрибуты тегов <TD> и <TH>.....	24
Колонки.....	25
Использование таблиц при дизайне страниц.....	26
Бегущая строка.....	27
Специальные символы.....	28

Первый HTML-документ

Результат выполнения HTML-программы браузером Google Chrome:



Текст HTML- программы, которая показывает эту страничку:

<code><HTML></code>	начало HTML-документа
<code><HEAD></code>	служебная информация о документе, в том числе название окна web-страницы
<code><META http-equiv="Content-Type" content="text/html; charset=windows-1251"></code>	
<code><TITLE> Первый HTML-документ </TITLE></code>	
<code></HEAD></code>	
<code><BODY></code>	начало тела документа
<code><H1>Принц Персии</H1></code>	заголовок документа
<code><HR></code>	горизонтальная линия
Принц побывал на Острове Времени, где он спас повелительницу времени. Он изменил свою судьбу и сумел избежать гибели. Когда он вернулся домой, то обнаружил, что его королевство охвачено войной с могущественным врагом.	текст, отображаемый на web-странице
<code></BODY></code>	конец тела документа
<code></HTML></code>	конец HTML-документа

Команды языка (теги) заключаются в угловые скобки. Как правило, теги парные. Первый тег открывает описание команды, определяет ее начало, второй, отличающийся от первого наличием косой черты перед ключевым словом (именем тега), закрывает его.

Тег `<HTML>` должен открывать программу, а тег `</HTML>` – закрывать ее. Между этими двумя основными тегами располагается заголовок программы и ее тело:

```
<HTML>
  заголовок программы (служебная информация для браузера)
  тело программы
</HTML>
```

Заголовок программы

В блоке `<HEAD>...</HEAD>` располагаются общие правила отображения HTML-документа, а также содержится вспомогательная информация о документе ¹.

Команда (тег)

```
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

задает браузеру информацию о кодировке, в которой написан текст HTML-документа. В данном случае задана стандартная кодировка Windows.

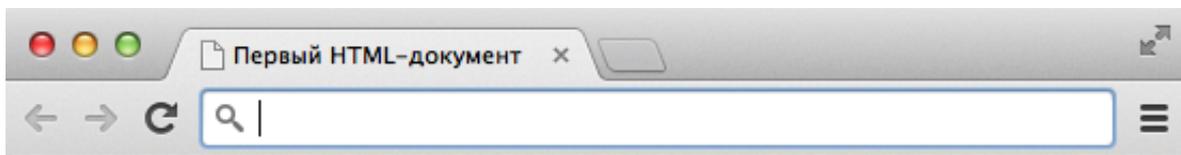


Тег `<META>` не имеет парного (закрывающего) тега

Между парой тегов `<TITLE>` и `</TITLE>` располагается название (имя) HTML-документа. Это имя браузер использует в заголовке окна, в котором показывает документ.

```
<TITLE> Первый HTML-документ </TITLE>
```

Заголовок окна (название ярлыка) браузера Google Chrome выглядит следующим образом:



Тело программы

Между тегами `<BODY>` и `</BODY>` располагаются команды, следуя которым браузер выводит информацию в окно документа.

В первой программе были использованы следующие команды (тэги):

```
<H1>...</H1>    – заголовок  
<HR>           – горизонтальная линия
```

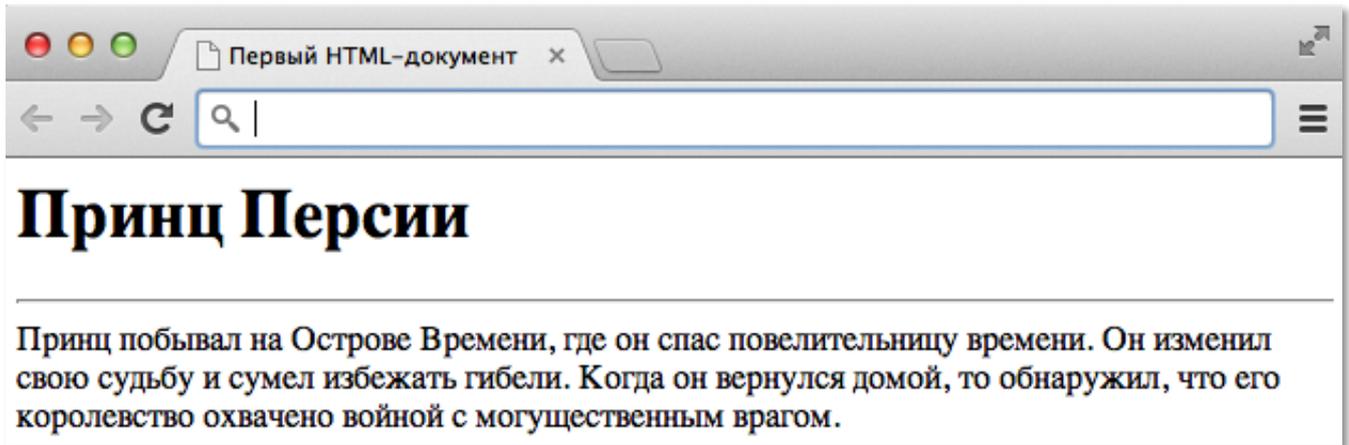
¹ Содержимое тега `<HEAD>...</HEAD>` не отображается напрямую на веб-странице, за исключением тега `<TITLE>...</TITLE>` устанавливающего заголовок окна веб-страницы.

Как работает браузер

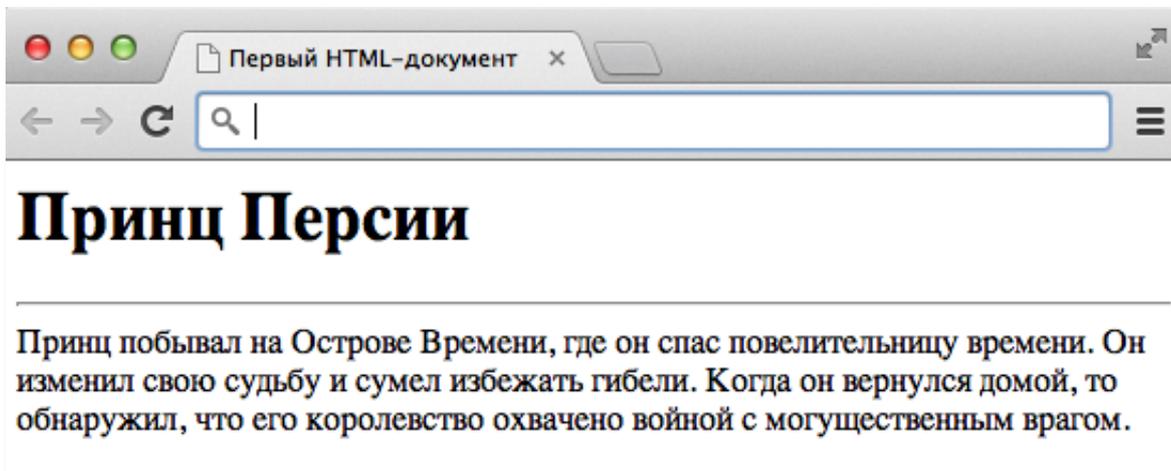
Браузер, выполняя программу (интерпретируя HTML-текст), формирует на экране документ, который видит пользователь. Элементы отображаются на экране последовательно, в соответствии с порядком следования тегов в HTML-программе.

Программируя на HTML, не следует ориентироваться на определенные размеры окна документа. Браузер, выполняя программу, использует то окно, которое предоставил ему пользователь. Одна и та же HTML-программа может отображаться на экран немного по-разному на разных компьютерах и даже на одном компьютере при разных его настройках.

Окно "Первого HTML-документа" может, выглядеть так:



Или так:



Текст выводится таким образом, чтобы поместиться в окне.



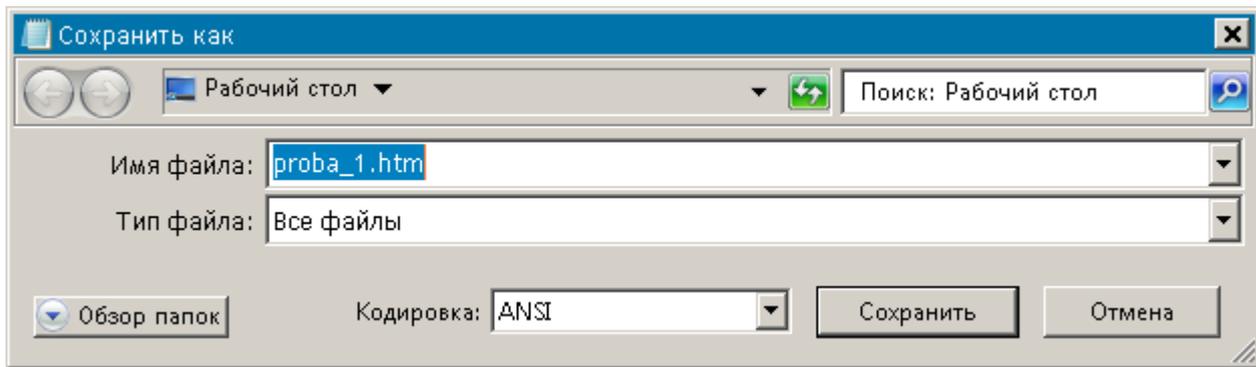
Указанные ограничения являются положительной стороной HTML, делая его независимым от операционной системы (Windows, Linux, MacOS) и параметров конкретного компьютера. Это означает, что HTML-документ способны увидеть все пользователи, на компьютерах которых установлен браузер.

Это свойство HTML-программ играет важную роль в глобальной компьютерной сети Интернет, где представлены различные типы компьютеров и операционных систем.

Сохранение файла

Для сохранения файла под именем *proba_1*, написанного в программе Блокнот, необходимо выполнить команды: **Файл** → **Сохранить как...**

и выбрать параметры сохранения в соответствии с рисунком:



Заголовки в тексте документа

Заголовки являются важной частью документа. Они создают иерархическую структуру документа, разделяя информацию на отдельные логические части и существенно улучшают ее восприятие.

Заголовки бывают разные по значению или, как говорят, по уровню.

Учебник, например, имеет название – это заголовок первого уровня.

Текст в учебнике разбит на главы; названия глав – это заголовки второго уровня.

Главы содержат параграфы с заголовками третьего уровня.

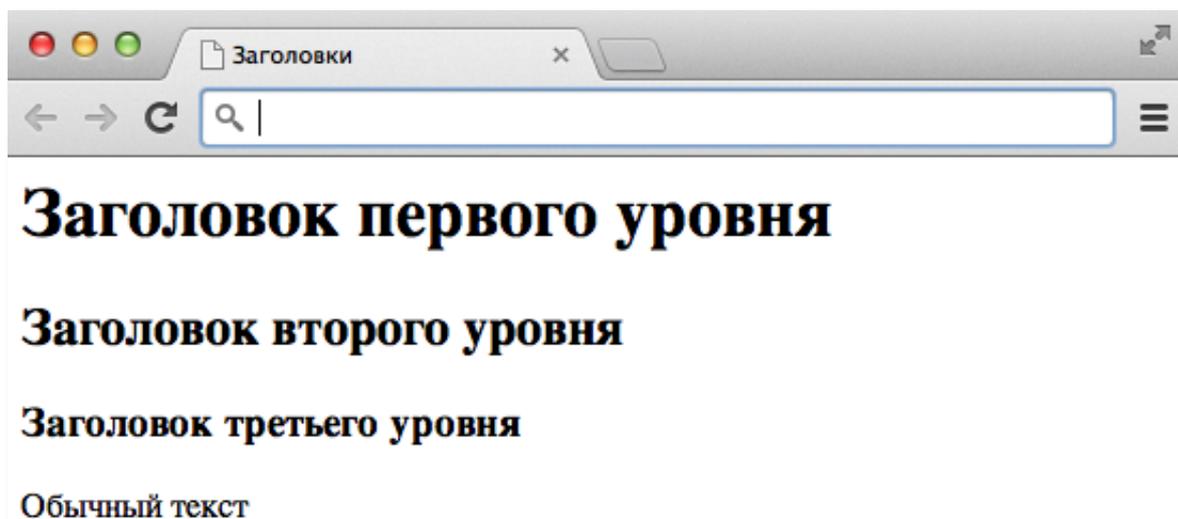
Параграфы – пункты, обозначенные заголовками четвертого уровня.

В HTML-документе поддерживаются заголовки шести уровней при помощи конструкции:

`<Hn>текст заголовка</Hn>`

В качестве *n* следует использовать одну из цифр: 1...6, обозначающую уровень заголовка.

Заголовки выделяются браузером при показе HTML-документа размером шрифта и жирностью:



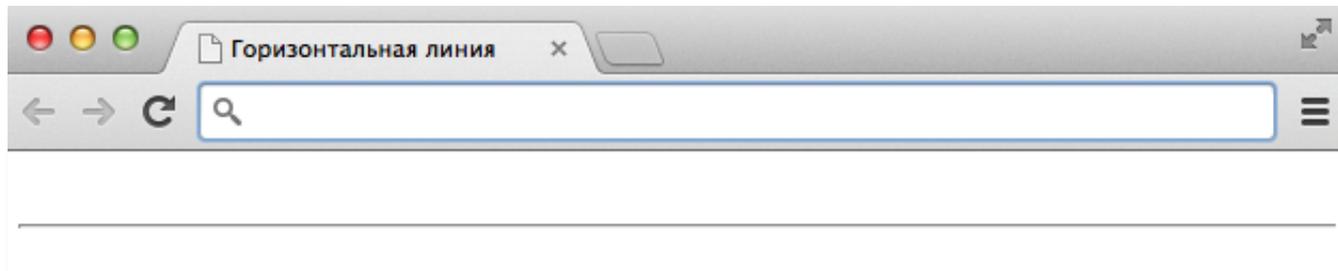
Использовать заголовки следует в соответствии с их логическим уровнем в структуре документа. Не следует использовать тег `<Hn>...</Hn>` для выделения обычного текста.

Горизонтальная линия (разделитель)

Линия – простейший графический элемент, который умеет строить браузер.

Горизонтальные линии удобно использовать для дополнительного выделения (или разделения) элементов документа на экране.

Горизонтальная линия задается тегом `<HR>`. Этот тег не имеет парного (закрывающего) тега. Браузер, выполняя эту команду, выведет на экран горизонтальную линию:



Принудительный разрыв строки

В программе "Первого HTML-документа" текст написан следующим образом:

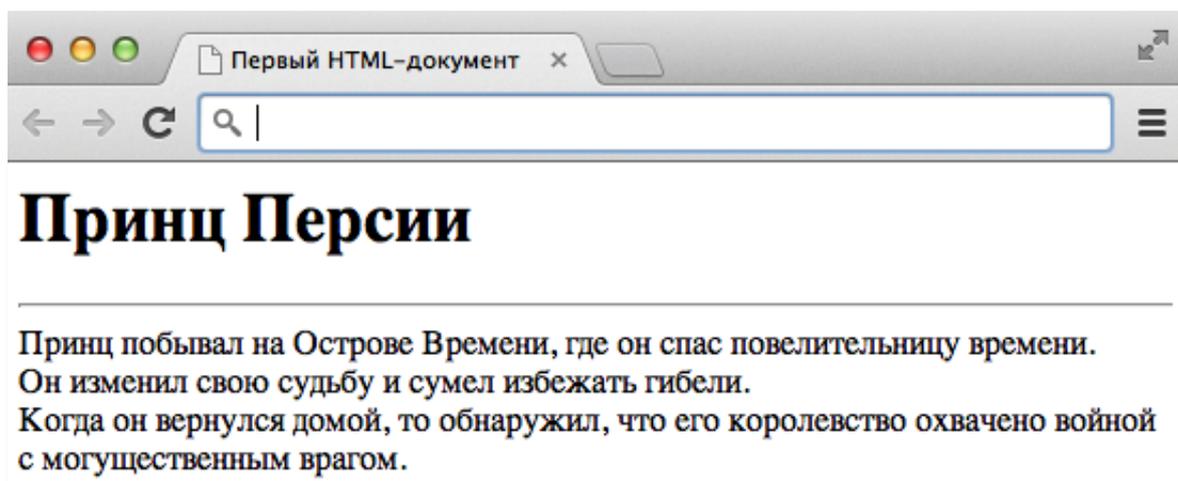
Принц побывал на Острове Времени, где он спас повелительницу времени. Он изменил свою судьбу и сумел избежать гибели. Когда он вернулся домой, то обнаружил, что его королевство охвачено войной с могущественным врагом.

Браузер выводит этот текст как единый абзац.

HTML имеет средство для задания принудительного перехода на новую строку внутри абзаца. Тэг `
` заставляет браузер продолжить вывод абзаца с новой строки.

Изменим "Первый HTML-документ" следующим образом:

Принц побывал на Острове Времени, где он спас повелительницу времени. `
`
Он изменил свою судьбу и сумел избежать гибели. `
`
Когда он вернулся домой, то обнаружил, что его королевство охвачено войной с могущественным врагом.



Полный текст HTML-программы, которая показывает эту страницу:

```
<HTML>
<HEAD>
<TITLE>Первый HTML-документ</TITLE>
</HEAD>
<BODY>
<H1> Принц Персии </H1>
<HR>
Принц побывал на Острове Времени, где он спас повелительницу времени.<BR>
Он изменил свою судьбу и сумел избежать гибели.<BR>
Когда он вернулся домой, то обнаружил, что его королевство охвачено войной с могущественным врагом.
</BODY>
</HTML>
```



Тег **
** не имеет парного
(закрывающего) тега

Атрибуты тегов

Открывающие теги могут содержать атрибуты, то есть дополнительную информацию о свойствах отображаемого на экране элемента.

Атрибуты записываются внутри открывающего тега после его имени в виде отдельного ключевого слова или ключевого слова, знака "=" и параметра (значения атрибута).



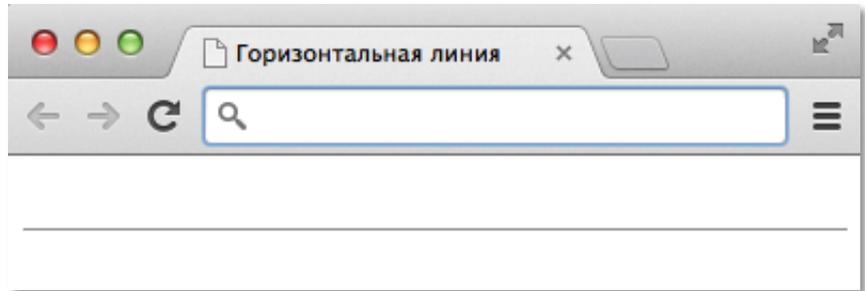
Порядок следования атрибутов в теге
не имеет значения

Атрибут действует от открывающего тега, в котором он задан, до закрывающего или только внутри тега, если тег не имеет парного.

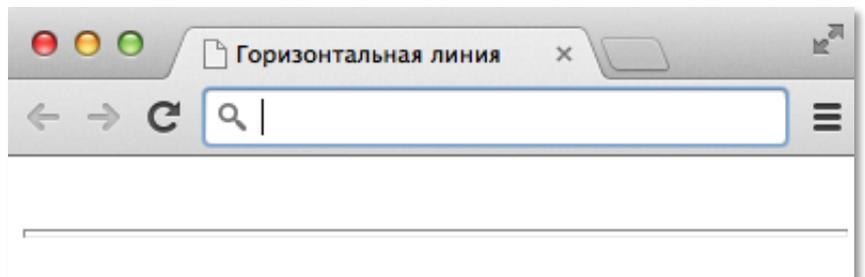
Горизонтальная линия (разделитель)

Тег, рисующий горизонтальную линию, может содержать атрибут `size`, изменяющий толщину линии. Если этот атрибут не задан, то толщина разделителя будет зависеть от браузера. Задавая разные параметры атрибуту `size`, можно получить разные по толщине линии.

`<HR size=1>`

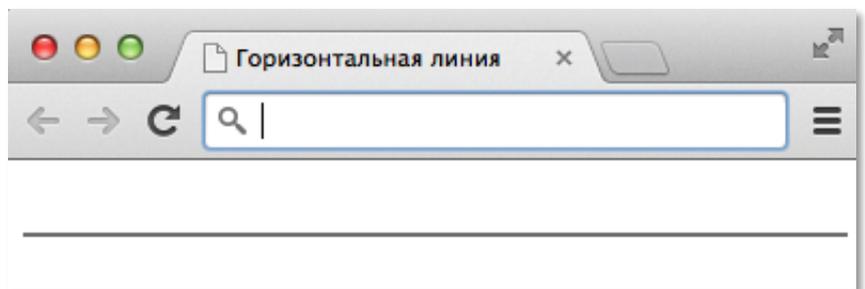


`<HR size=4>`



Чтобы браузер рисовал "плоские" линии, необходимо использовать атрибут `noshade`:

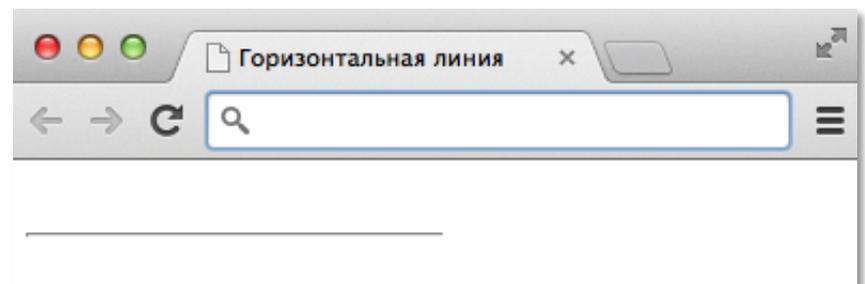
`<HR size=2 noshade>`



Существуют и другие атрибуты для тега, рисующего разделитель.

С их помощью можно выровнять разделитель (по левому краю, по центру, по правому краю), задать ширину разделителя (в процентах от ширины страницы).

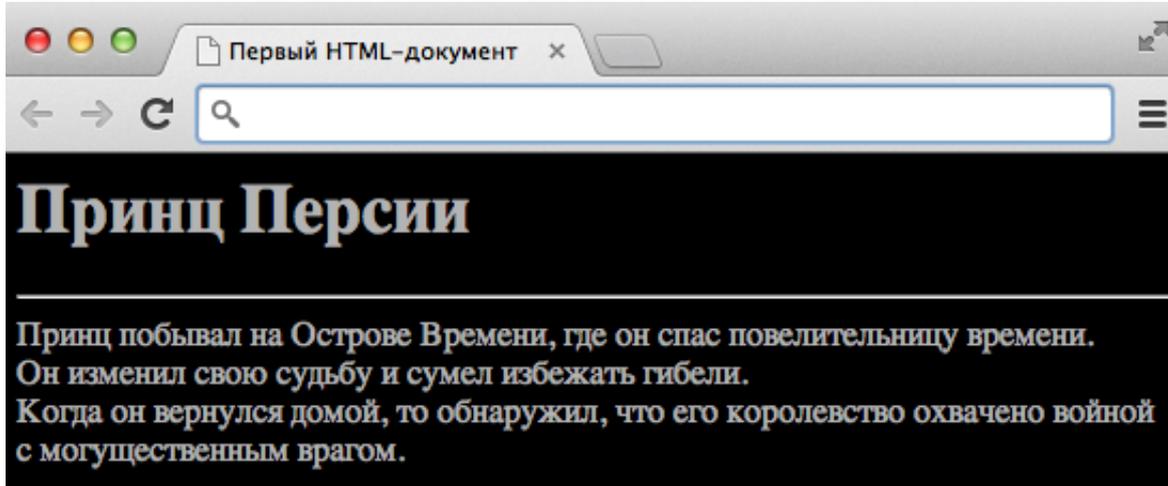
`<HR width=50% align=left>`



Цвет фона и цвет шрифта

В предыдущих примерах не задавались явно ни цвет фона документа, ни цвет шрифта текста. Когда в программе на HTML цвета не заданы, браузер использует установки по умолчанию.

Цвета для фона и шрифта на всем протяжении документа можно задать при помощи атрибутов bgcolor и text в теге <BODY>.



Фрагмент HTML- программы:

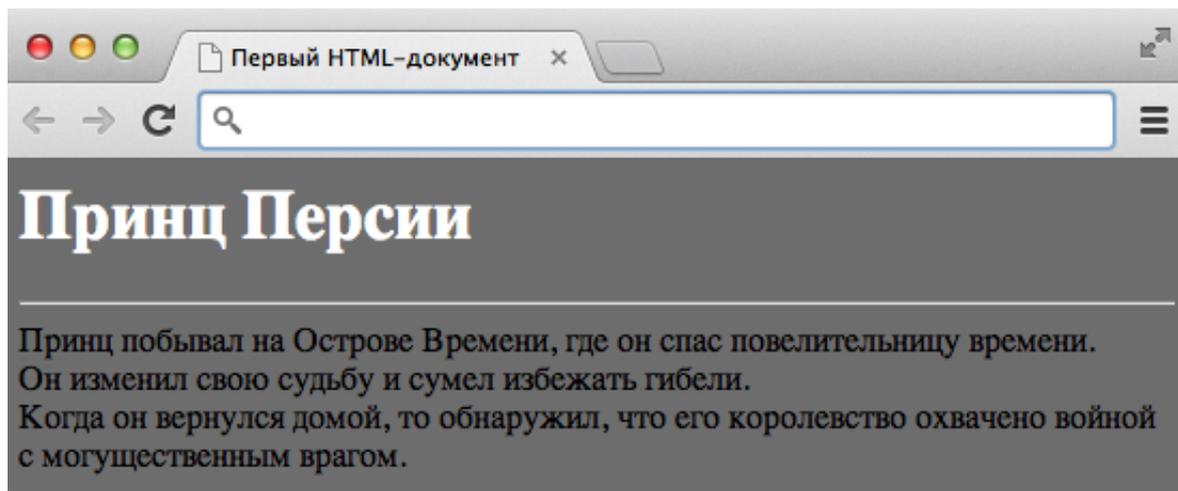
```
<HTML>
  <HEAD>
  ...
</HEAD>
<BODY bgcolor=black text=silver>
  ...
</BODY>
</HTML>
```

Таблица основных цветов

	<i>black (черный)</i>		<i>green (зеленый)</i>
	<i>silver (серебряный)</i>		<i>lime (светло-зеленый)</i>
	<i>gray (серый)</i>		<i>olive (оливковый)</i>
	<i>white (белый)</i>		<i>yellow (желтый)</i>
	<i>maroon (коричневый)</i>		<i>navy (темно-синий)</i>
	<i>red (красный)</i>		<i>blue (синий)</i>
	<i>purple (темно-сиреневый)</i>		<i>teal (петроль)</i>
	<i>fuchsia (сиреневый)</i>		<i>aqua (голубой)</i>

Атрибуты цветов, заданные в теге <BODY>, действуют до парного тега </BODY>, то есть, до конца документа. Цвет фона может задаваться только один раз, а цвет шрифта можно менять внутри документа многократно при помощи команды Установка цвета в этом теге выполняется при помощи атрибута color.

Ниже приведены два варианта одного и того же HTML-документа



Первый вариант.

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <BODY bgcolor=gray text=white >
    ...
    <FONT color=black>
      ...
    </FONT>
  </BODY>
</HTML>
```

заданы цвета для всего документа

задан новый цвет шрифта (черный)

восстановлен цвет шрифта (белый)

Второй вариант.

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <BODY bgcolor=gray text=black >
    <FONT color=white>
      ...
    </FONT>
    ...
  </BODY>
</HTML>
```

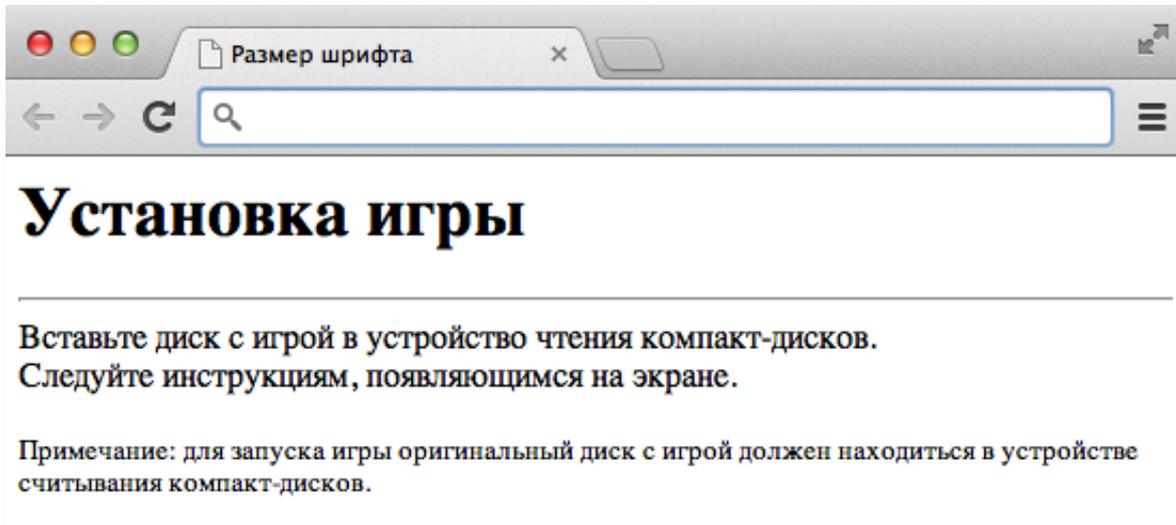
заданы цвета для всего документа

задан новый цвет шрифта (белый)

восстановлен цвет шрифта (черный)

Изменение размера шрифта

Для изменения размеров шрифта существуют теги, сохраняющие единый стиль оформления документа: `<BIG>...</BIG>` – для укрупнения и `<SMALL>...</SMALL>` – для уменьшения. Эти теги могут быть вложенными, что позволяет увеличивать (уменьшать) размер шрифта в несколько раз.



Фрагмент HTML- программы:

```
<HTML>
  <HEAD>
    ...
  </HEAD>
  <BODY>
    ...
    <SMALL>
      Примечание: для запуска игры оригинальный диск
      с игрой должен находиться в устройстве считывания
      компакт-дисков.
    </SMALL>
  </BODY>
</HTML>
```

уменьшен размер шрифта

восстановлен размер шрифта

Начертание шрифта

Для выделения слова или словосочетания курсивом можно использовать тег `<I>...</I>`, а для их выделения полужирным шрифтом – тег `...`

Для отображения шрифта в виде нижнего индекса используется тег `_{...}`.

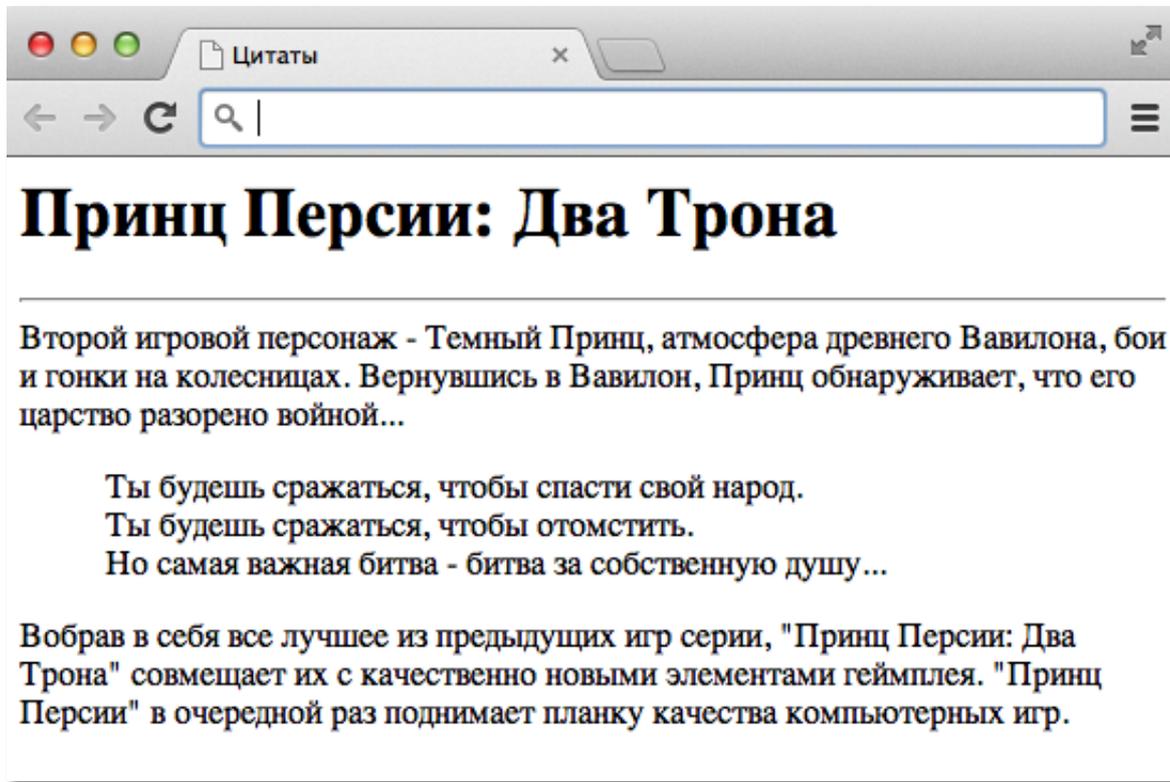
При этом текст располагается ниже базовой линии остальных символов строки.

Для отображения шрифта в виде верхнего индекса используется тег `^{...}`.

При этом текст располагается выше базовой линии остальных символов строки.

Текстовые цитаты

Цитаты выделяют из основного текста при помощи тега `<BLOCKQUOTE>...</BLOCKQUOTE>`. Браузер отображает цитату с большим, чем у обычного текста, левым и правым отступом от края страницы.



Фрагмент HTML- программы:

```
...  
<BLOCKQUOTE>  
Ты будешь сражаться, чтобы спасти свой народ. <BR>  
Ты будешь сражаться, чтобы отомстить. <BR>  
Но самая важная битва - битва за собственную душу...  
</BLOCKQUOTE>  
...
```

Авторское форматирование

Браузеры игнорируют множественные пробелы и символы конца строки в программе при показе документа на экране. Но из этого правила есть исключение.

Тег `<PRE>...</PRE>` заставляет браузер выводить текст на экран так, как он записан в программе – со всеми пробелами и концами строк.

Абзац

Как правило, браузер использует пустую строку для отделения одного абзаца от другого или абзаца от других элементов документа.

Задается абзац тегами `<P>` и `</P>`, между которыми помещается текст².

Фактически работа тега `<P>` сводится к выводу перед текстовым фрагментом абзацного отступа.

По умолчанию браузер выводит на экран текстовую информацию следующим образом:

- абзац выравнивается по левому краю.
- между словами всегда помещается ровно по одному пробелу независимо от того, сколько пробелов есть в HTML-программе.
- перенос текста на новую строку происходит тогда, когда очередное слово не помещается в экранной строке, а не тогда, когда слово написано с новой строки в HTML-программе.

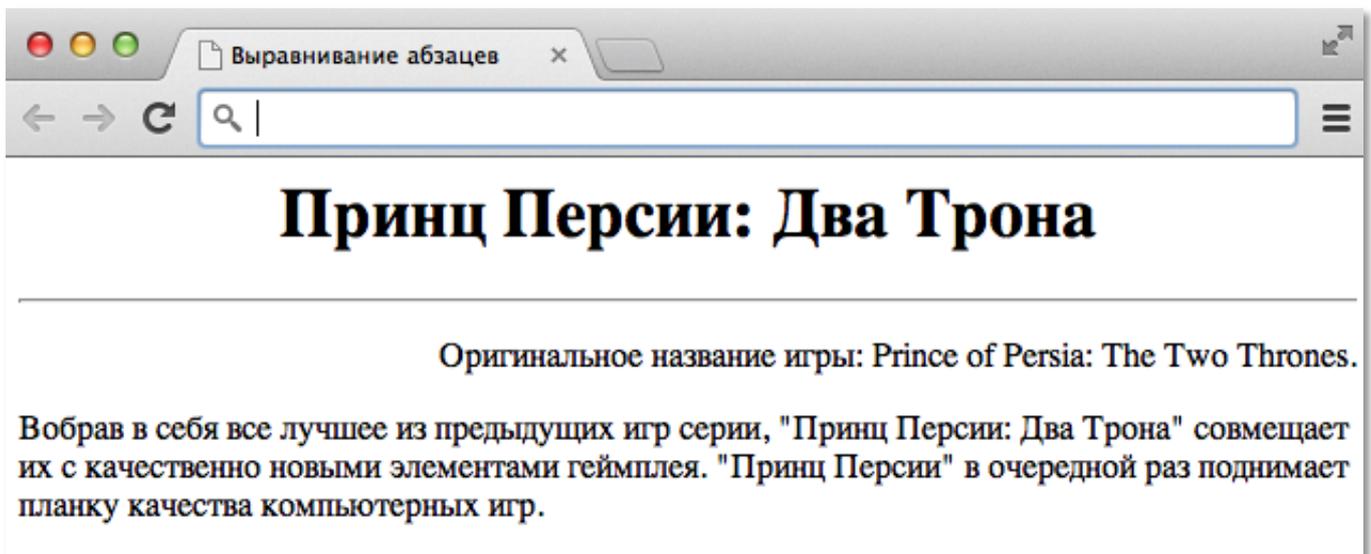
Выравнивание абзацев и фрагментов текста

Тег `<P>`, записанный без атрибутов, выравнивает текст по левому краю страницы.

Можно запрограммировать вывод текста с выравниванием по правому краю или по центру.

`<P align=right>` – выравнивает абзац по правому краю

`<P align=center>` – выравнивает абзац по центру



Фрагмент HTML- программы:

```
<H1 align=center> Принц Персии: Два Трона </H1>
```

```
<HR>
```

```
<P align=right> Оригинальное название игры: Prince of Persia: The Two Thrones.
```

```
<P>
```

```
Вобрав в себя все лучшее из предыдущих игр серии, "Принц Персии: Два Трона" совмещает их с качественно новыми элементами геймплея. "Принц Персии" в очередной раз поднимает планку качества компьютерных игр.
```

² Парный тег `</P>` игнорируется браузером, поэтому он не является обязательным.

Маркированный список

Каждый элемент этого списка браузер предваряет меткой в начале строки. Тег ``, задающий маркированный список, имеет следующую структуру:

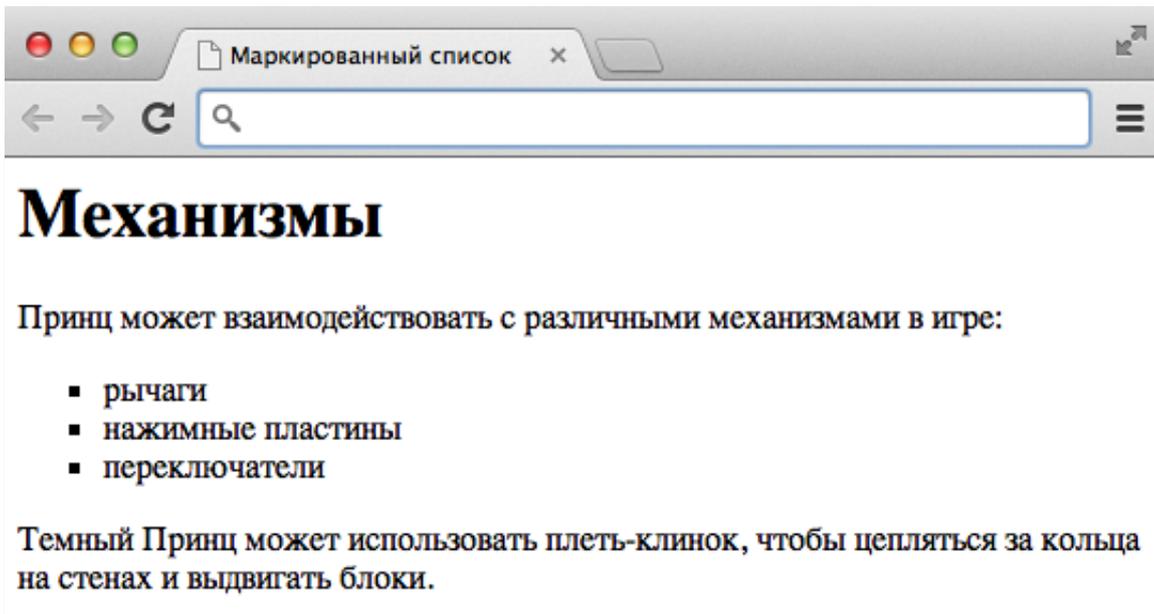
```
<UL>
  <LI>первый элемент</LI>
  <LI>второй элемент</LI>
  ...
  <LI>последний элемент</LI>
</UL>
```



Конечный тег `` не является обязательным

Вид метки, которую браузер помещает перед каждым элементом списка, настраивается при помощи атрибута `type`. Этот атрибут может отсутствовать или принимать одно из значений:

Начальный тег	Вид метки	Название
<code><UL type=disc ></code>	•	диск
<code><UL type=circle></code>	○	окружность
<code><UL type=square></code>	▪	квадрат



Фрагмент HTML- программы:

Принц может взаимодействовать с различными механизмами в игре:

```
<UL type=square>
  <LI>рычаги</LI>
  <LI>нажимные пластины</LI>
  <LI>переключатели
</UL>
```

Темный Принц может использовать плеть-клинок, чтобы цепляться за кольца на стенах и выдвигать блоки.

Нумерованный список

В нумерованных списках каждый элемент снабжён номером, вид и начальное значение которого настраивается специальным атрибутом.

Нумерованный список задается при помощи тега ``.

Тег ``, задающий нумерованный список, имеет следующую структуру:

```
<OL>  
  <LI>первый элемент</LI>  
  <LI>второй элемент</LI>  
  ...  
  <LI>последний элемент</LI>  
</OL>
```



Конечный тег `` не является обязательным

Вид номера определяется значением атрибута `type`.

Этот атрибут может отсутствовать или принимать одно из значений:

Начальный тег	Вид номера
<code></code>	Нумерация выполняется арабскими цифрами (1, 2, 3,...)
<code><OL type=A></code>	Нумерация выполняется прописными буквами (A, B, C,...)
<code><OL type=a></code>	Нумерация выполняется строчными буквами (a, b, c,...)
<code><OL type=I></code>	Нумерация выполняется большими римскими цифрами (I, II, III,...)

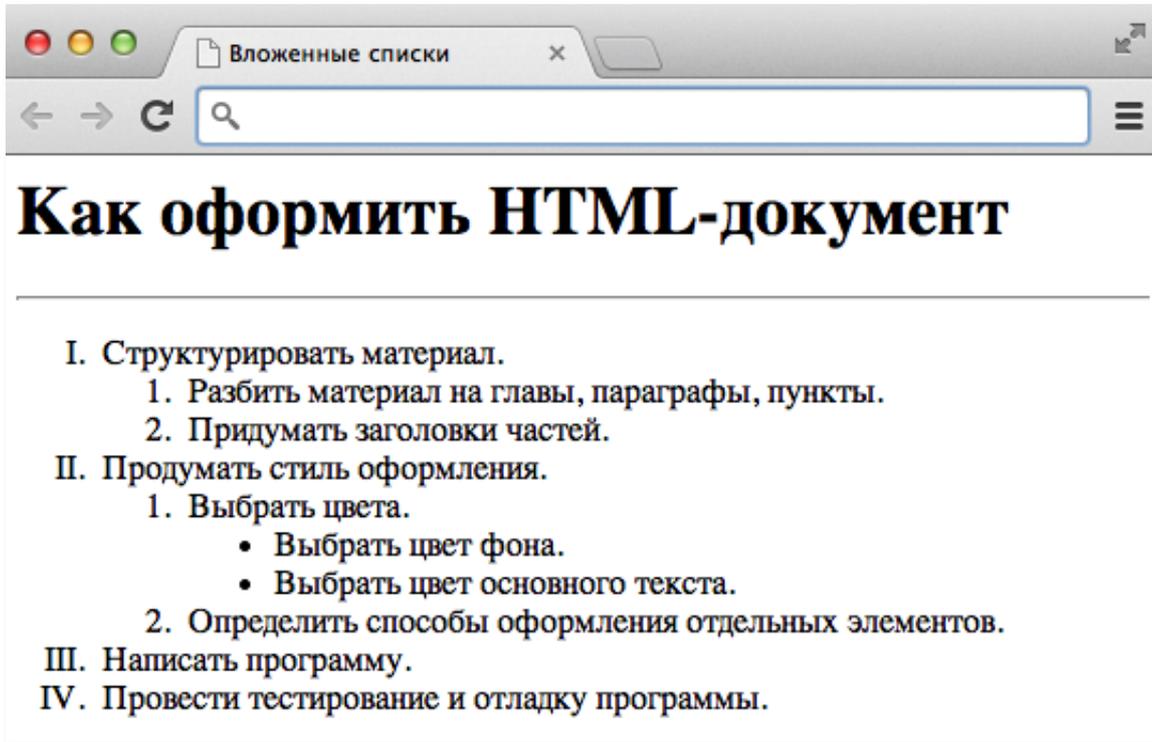


Естественную нумерацию элементов списка (с единицы) можно изменить при помощи атрибута `start=n`. Число `n` задаёт начальное значение нумерации. Если для нумерации используются буквы или римские цифры, то стартовое значение равно `n`-му элементу в выбранной системе нумерации (например, 4 означает D или IV).

Некоторые браузеры делают фиксированный отступ для вывода элементов списка на экран. Нумерация при этом, смещается от отступа к левому краю окна. Когда номер достаточно большой, места не хватает, и номер "налезает" на соответствующий элемент списка. Для исключения этого эффекта можно использовать тег `<BLOCKQUOTE>...</BLOCKQUOTE>`. Этот тег увеличивает левый отступ списка.

Вложенные списки

Уровень вложенности списков не ограничен, что позволяет использовать списки для отображения сложных иерархических структур.



Фрагмент HTML- программы:

```
<OL type=I>  
  <LI> Структурировать материал.  
    <OL>  
      <LI> Разбить материал на главы, параграфы, пункты.  
      <LI> Придумать заголовки частей.  
    </OL>  
  <LI> Продумать стиль оформления.  
    <OL>  
      <LI> Выбрать цвета.  
        <UL type=disc>  
          <LI> Выбрать цвет фона.  
          <LI> Выбрать цвет основного текста.  
        </UL>  
      <LI> Определить способы оформления отдельных элементов.  
    </OL>  
  <LI> Написать программу.  
  <LI> Провести тестирование и отладку программы.  
</OL>
```

Графика

Из всех тегов, рассмотренных выше, только один тег **<HR>** создавал новый элемент (горизонтальную линию), а все остальные теги размечали исходный текст документа.

Тег, предназначенный для вывода на экран картинки из файла, непарный, и он имеет имя ****.

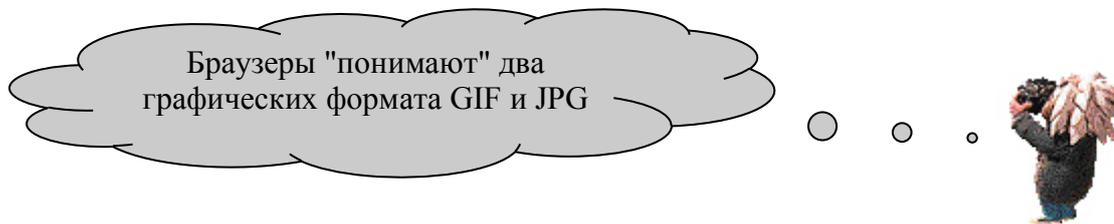


Рисунок в тексте

Для вставки рисунка необходимо использовать тег ****.
У этого тега нет парного закрывающего тега, но есть много атрибутов.

Атрибут *src* = "имя файла"

Главным (и единственным обязательным) атрибутом команды является атрибут *src*, при помощи которого необходимо задать имя файла с картинкой. Например, тег **** заставит браузер отобразить на экране графический файл *risunok.gif* из текущего каталога.

Как правило, графические файлы помещают в отдельный каталог. Команда вывода изображения при этом может иметь, например, такой вид³: ****

Атрибут *alt* = "текст надписи"

Если браузер не находит картинки в указанном месте на диске, он вместо нее рисует на экране маленький прямоугольник и вписывает в него надпись, которая задана атрибутом *alt*:

Если картинка построена браузером, то содержимое *alt*-надписи выводится на экран при зависании курсора мыши над рисунком (если не задан атрибут *title*, иначе выводится значение атрибута *title*). *Alt*-надпись смогут прочитать пользователи, которые работают в текстовых браузерах.

Атрибуты *width=n* и *height=m*

Эти атрибуты задают ширину *n* и высоту *m* (в пикселах) прямоугольника, в который "вписывается" рисунок. Если атрибуты не заданы, картинка рисуется в естественных размерах.

Если указанные размеры не совпадают с размерами рисунка, браузер производит масштабирование рисунка, подгоняя его под указанные величины. Если размеры указываются неверно, то рисунок выводится браузером с искажением.

Атрибут *border=k*

Атрибут позволяет задать рамку рисунка толщиной в *k* пикселей. При отсутствии атрибута рамка не рисуется.

³ Подробнее см ["Имена файлов и ссылки на них"](#)

Атрибут align

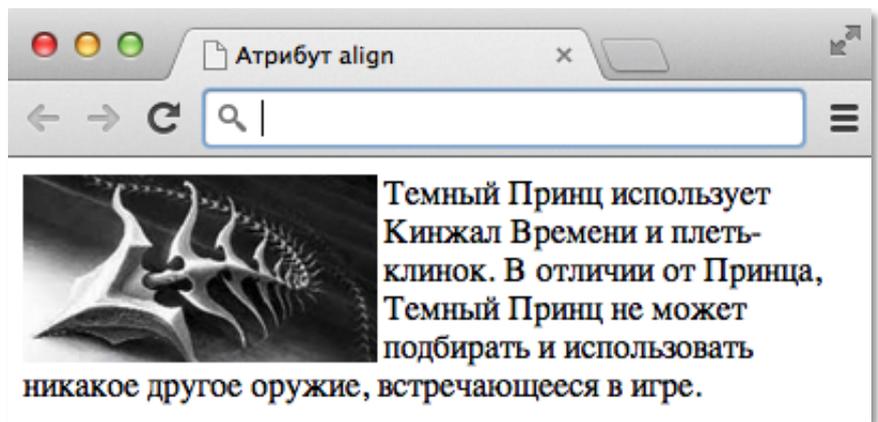
Этот атрибут определяет положение рисунка по отношению к соседним элементам документа.

Можно использовать следующие значения атрибута:

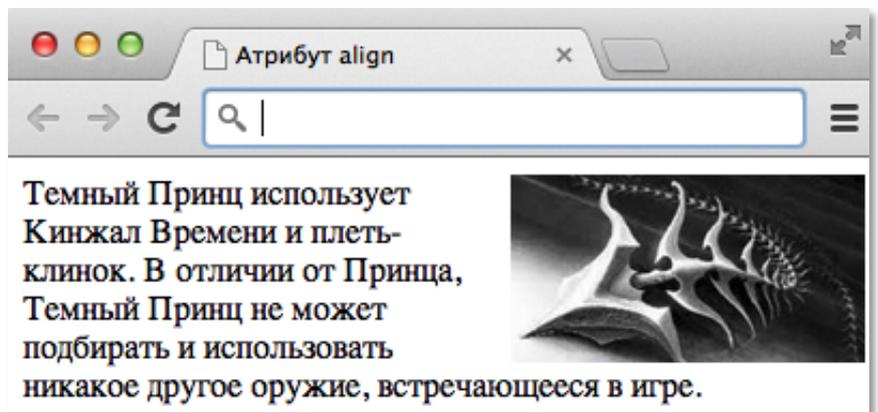
- align=middle – вертикальное выравнивание по центру
- align=top – вертикальное выравнивание по верхнему краю
- align=bottom – вертикальное выравнивание по нижнему краю
- align=left – горизонтальное выравнивание по левому краю
- align=right – горизонтальное выравнивание по правому краю

Примеры использования:

align=left



align=right



Имена файлов и ссылки на них

- Имена файлов HTML-программ имеют расширение htm или html.

Без необходимости обозначение html лучше не использовать, т. к. не все операционные системы "понимают" четыре символа в расширении имени файла.

- Если многофайловый гипертекстовый документ имеет иерархическую структуру, то в обозначениях файлов желательно учитывать иерархию.

Пусть основной файл с оглавлением называется 00.htm. Файл с первой главой – 01.htm, со второй – 02.htm и так далее. Если у глав есть параграфы, имя для каждого файла параграфа следует сформировать из имени главы, добавляя к нему номер параграфа. Это позволит по имени файла однозначно определить его место в общей структуре многофайлового документа.

Браузер ищет файл в том же каталоге, что и файл, содержащий ссылку на него.

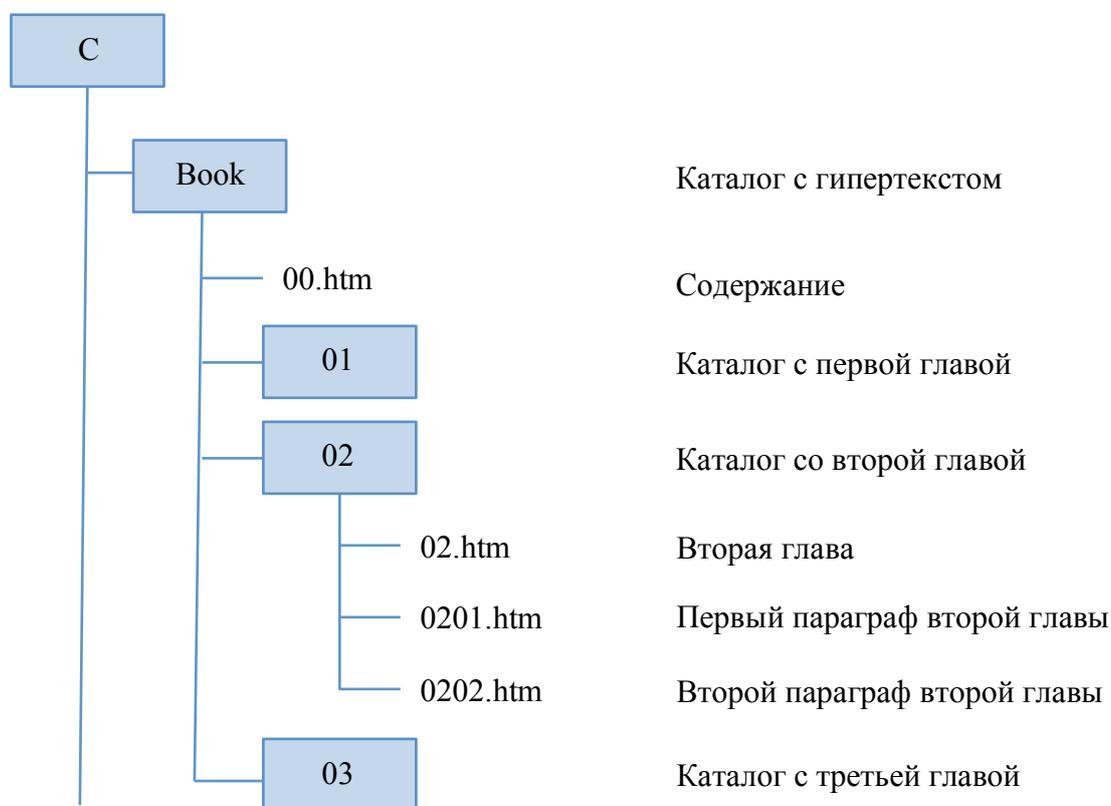
Если необходимо сослаться на файл из другого каталога, то можно:

- указывать в ссылке полный путь к файлу, начиная с имени дисковода;
- указывать в ссылке путь по отношению к файлу со ссылкой.

Второй способ лучше, т. к. позволяет переносить гипертекст в другие каталоги или на другие компьютеры без дополнительных правок в HTML-коде.

Пусть, например, гипертекст расположен в каталоге Book корневого каталога диска C.

В каталоге Book лежит файл 00.htm с оглавлением. Остальные файлы помещены в подкаталоги 01, 02, 03 в соответствии с принадлежностью к той или иной главе документа.



Как можно сослаться из файла 00.htm на файл 02.htm?

- C:/BOOK/02/02.htm (абсолютная адресация)
- ./02/02.htm (относительная адресация)

Абсолютная адресация приводит к непереносимости продукта.

Стоит перенести каталог BOOK, например, на диск D, как ссылки перестают работать.

Относительная адресация предлагает браузеру ориентироваться на текущий каталог.

Можно переместить каталог BOOK и даже переименовать его – ссылки будут работать.

Относительную ссылку из файла 02.htm на файл 00.htm можно записать так: ../00.htm (две точки и косая черта в самом начале записи указывают на родительский каталог). Если необходимо из файла 02.htm сослаться на файл proba.htm, лежащий в каталоге 03, то ссылка может выглядеть так: ../03/proba.htm



При записи пути к файлу лучше использовать косую черту с наклоном вправо (слэш), а не влево, как принято в Windows, даже если браузер находит ссылки, где косая черта наклонена влево (обратный слэш). Наклон черты становится существенным при размещении гипертекста на сервере в Интернет. Серверы, как правило, работают под управлением операционной системы UNIX, а в ней чёрточки, обозначающие вложенность каталогов, наклонены вправо.



Лучше использовать относительную адресацию

Переход внутри одного документа

Для задания гипертекстового перехода внутри документа используют два тега <A>. Первый тег с атрибутом href является началом перехода, второй с атрибутом name – местом, в которое должен перейти браузер.

Фрагмент программы	Комментарий
... текст ...	Задание перехода по метке
... ...	Метка. Сюда браузер переходит по ссылке

Для организации перехода внутри документа нужно:

1. Выбрать имя для метки.
 - Имя должно быть уникальным в HTML-документе. Это означает, что других меток с таким именем в документе быть не должно.
 - Имя может состоять из любых символов. В случае, когда имя содержит пробелы или ключевые символы HTML, кавычки обязательны.
 - Для имени желательно использовать латиницу или цифры.
2. Запрограммировать переход по метке.

В месте перехода нужно написать команду текст

Здесь:

 - # – ключевой символ;
 - метка – выбранное имя для метки;
 - текст – запись, которая будет выглядеть на экране браузера как ссылка.

3. Установить тег с меткой в нужном месте HTML-программы.

Перед фрагментом HTML-программы, на который должен выполняться переход, записать тег - метку ``

При этом:

- имя метки должно быть то же самое, что и в команде, задающей переход;
- на экране монитора тег-метка `` никаких изображений не строит.

Переход к другому документу

Чтобы браузер загрузил в свое окно новый HTML-документ, нужно записать в программе ссылку при помощи тега `<A>` с атрибутом `href=имя_файла`.

Фрагмент программы	Комментарий
... <code>текст</code> ...	При щелчке на ссылке браузер выводит на экран документ по программе, заданной в файле "имя_файла".

Переход к метке другого документа

Выше рассмотрено решение следующих задач:

- передача управления в другую часть документа;
- передача управления в другой документ (HTML-файл).

Эти методы можно совместить, передав управление из одного документа к метке внутри другого.

Первый HTML-файл	Второй HTML-файл
... <code>текст</code> <code></code> ...

Рисунок как ссылка

Можно заставить работать рисунок как гиперссылку.

Для этого необходимо вложить тег `` внутрь тега `<A>`:

``

Курсор при попадании на картинку-ссылку меняет свою форму. Щелчок мыши на рисунке заставляет браузер выполнять переход. Отличие от текстовой гиперссылки только в том, что рисунок-ссылка не подчеркивается, не выделяется цветом, а обрамляется в рамку.

Часто рамка вокруг рисунка нежелательна, даже когда она означает ссылку.

Избавиться от рамки, если рисунок является ссылкой, можно явным образом: `border=0`.

Таблицы

В практике создания сайтов таблицы совершенно незаменимы для создания полей страницы, выравнивания элементов на экране, многоколонной верстки, наложения картинок друг на друга. И, наконец, для отображения на экране табличного материала, но это - в последнюю очередь. Чаще используют таблицы, как ни парадоксально, не для отображения таблиц.

Таблица задается тегами `<TABLE>...</TABLE>`.

Внутри этих тэгов задаются строки тегами `<TR>...</TR>`.

Внутри строк задаются клетки (ячейки) тега `<TD>...</TD>`.



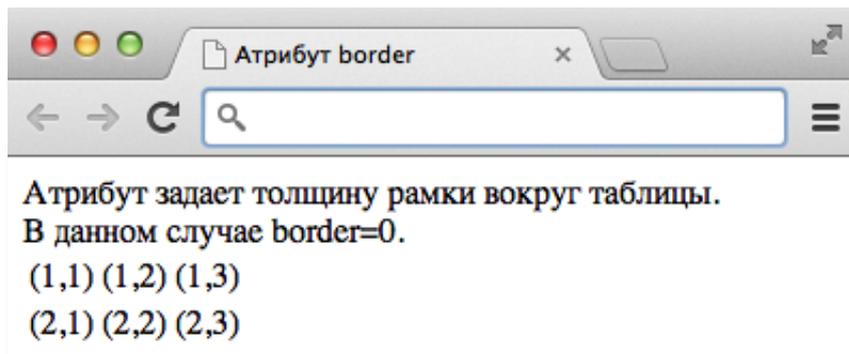
Конечный тег `</TABLE>`
является обязательным

Фрагмент программы	Комментарий	Таблица на экране						
<pre><TABLE border=1> <TR> <TD>(1,1)</TD> <TD>(1,2)</TD> </TR> <TR> <TD>(2,1)</TD> <TD>(2,2)</TD> </TR> <TR> <TD>(3,1)</TD> <TD>(3,2)</TD> </TR> </TABLE></pre>	Начало таблицы	<table border="1"><tbody><tr><td>(1,1)</td><td>(1,2)</td></tr><tr><td>(2,1)</td><td>(2,2)</td></tr><tr><td>(3,1)</td><td>(3,2)</td></tr></tbody></table>	(1,1)	(1,2)	(2,1)	(2,2)	(3,1)	(3,2)
	(1,1)		(1,2)					
	(2,1)		(2,2)					
	(3,1)		(3,2)					
	Начало первой строки							
	Первая ячейка первой строки							
	Вторая ячейка первой строки							
	Конец первой строки							
	Начало второй строки							
	Первая ячейка второй строки							
Вторая ячейка второй строки								
Конец второй строки								
Начало третьей строки								
Первая ячейка третьей строки								
Вторая ячейка третьей строки								
Конец третьей строки								
Конец таблицы								

Атрибуты тега `<TABLE>`

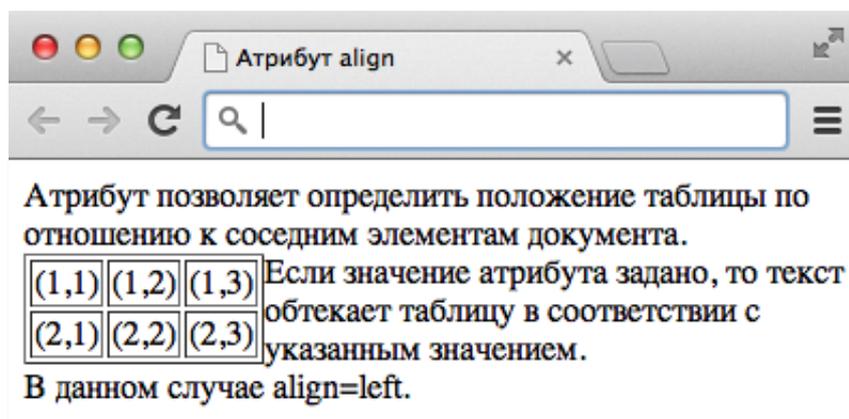
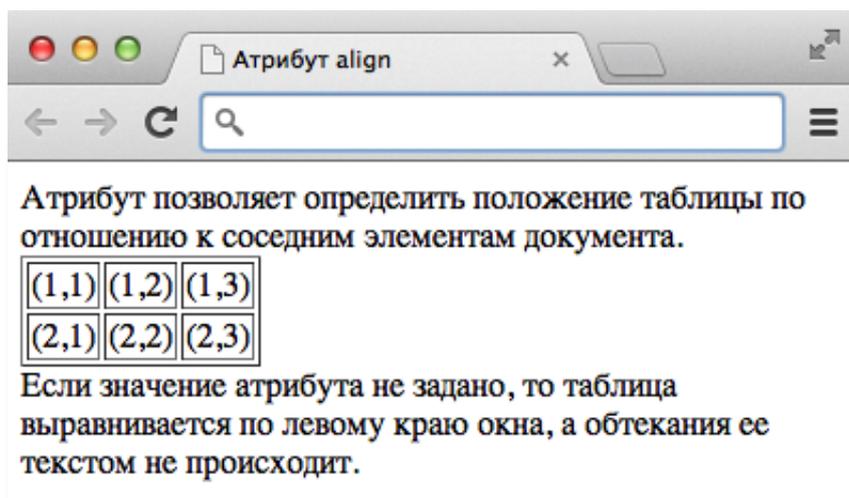
Атрибуты `border` и `bordercolor`

Первый атрибут задает толщину рамки вокруг таблицы, а второй – цвет этой рамки.



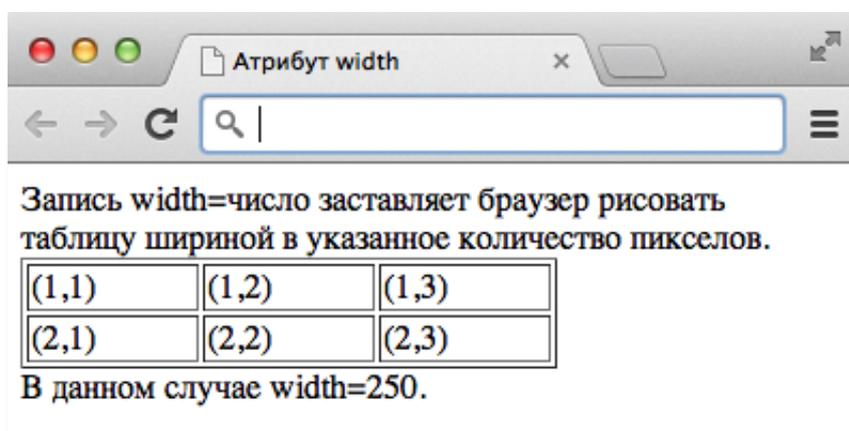
Атрибут align

Атрибут имеет для таблиц тот же смысл, что и для рисунков.



Атрибут width

Если атрибут не задан, браузер рисует таблицу минимальных размеров вокруг ее данных.



Можно задавать ширину таблицы в процентном отношении к ширине окна браузера. В этом случае используют запись width=процент.



Если для таблицы требуется больше места, значение `width` игнорируется. Если размер "уводит" таблицу за правую границу окна, то браузер добавит к окну горизонтальную полосу прокрутки. Желательно создавать документ так, чтобы горизонтальной полосы прокрутки не было. Во-первых, полоса прокрутки уменьшает полезный размер окна; во-вторых, ухудшает восприятие документа.

Атрибуты `cellpadding` и `cellspacing`

Первый атрибут задает расстояние между рамкой таблицы и содержимым ячеек, а второй атрибут – между рамками соседних ячеек. Когда атрибуты не заданы, браузер устанавливает их в соответствии с настройками по умолчанию, обычно: `cellpadding=1` и `cellspacing=2`.

Атрибуты `bordercolordark` и `bordercolorlight`

Атрибуты позволяют нарисовать объемную рамку. Первый атрибут задает цвет ее нижней и правой стороны, второй – левой и верхней. Некоторые браузеры эти атрибуты не поддерживают.

Атрибуты тега `<TR>`

HTML-таблица состоит из строк, каждая из которых задается тегом `<TR>...</TR>`.

Изменяя атрибуты тега `<TR>`, можно установить некие свойства сразу для всех ячеек строки.

Атрибут	Значения	Описание
<code>align</code>	left, center, right	Выравнивание по горизонтали
<code>valign</code>	top, middle, bottom, baseline	Выравнивание по вертикали
<code>bgcolor</code>	цвет	Цвет фона
<code>bordercolor</code>	цвет	Цвет линий рамки

Атрибуты тегов `<TD>` и `<TH>`

Каждая табличная строка состоит из ячеек⁴, которые последовательно описываются тегами `<TD>...</TD>` (обычная ячейка) и (или) `<TH>...</TH>` (ячейка-заголовок):

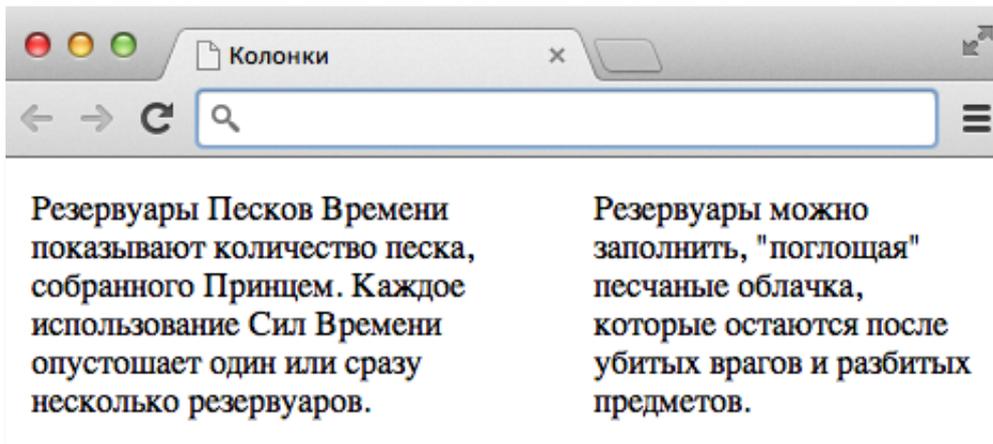
Содержимое обычной ячейки по умолчанию горизонтально выравнивается по левому краю, а ячейки-заголовка – по центру. Текст в ячейке-заголовке выделяется полужирным шрифтом.

Атрибут	Значения	Описание
<code>align</code>	left, center, right	Выравнивание по горизонтали
<code>valign</code>	top, middle, bottom, baseline	Выравнивание по вертикали
<code>width</code>	число или процент	Ширина ячейки
<code>bgcolor</code>	цвет	Цвет фона
<code>background</code>	файл	Фоновая картинка
<code>bordercolor</code>	цвет	Цвет линий рамки
<code>colspan</code>	число	Ширина большой ячейки (в столбцах)
<code>rowspan</code>	число	Высота большой ячейки (в строках)

⁴ В любую ячейку таблицы можно вложить другую таблицу.

Колонки

Используя таблицы, можно выводить на экран информацию в виде нескольких колонок.

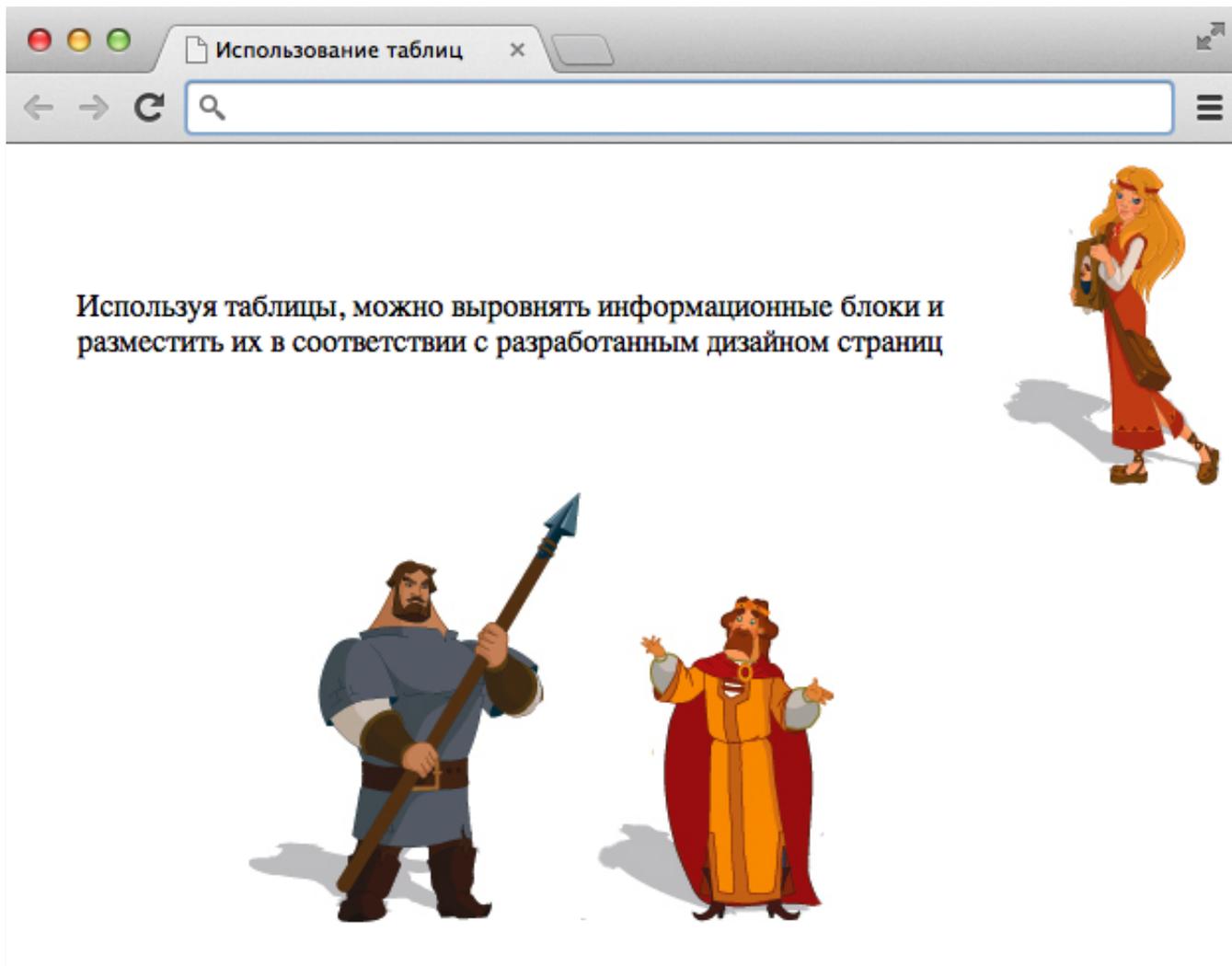


Фрагмент HTML- программы:

```
<TABLE border=0 cellpadding=5>
  <TR>
    <TD>
      Резервуары Песков Времени показывают количество песка, собранного Принцем.
      Каждое использование Сил Времени опустошает один или сразу несколько резервуаров.
    </TD>
    <TD>
      Резервуары можно заполнить, "поглощая" песчаные облачка, которые остаются после
      убитых врагов и разбитых предметов.
    </TD>
  </TR>
</TABLE>
```

Использование таблиц при дизайне страниц

Таблицы – инструмент, позволяющий расположить информацию так, как задумал дизайнер. Пример страницы, сконструированной с использованием таблицы:



Бегущая строка

Бегущая строка на странице задается тегом `<MARQUEE>...</MARQUEE>`.

Можно перемещать различные элементы – изображения, текст и т.д. Перемещение можно задать не только по горизонтали, но и по вертикали. В этом случае указываются размеры области, в которой будет происходить движение.

Атрибут behavior

Устанавливает способ движения содержимого внутри элемента. Значение по умолчанию: `scroll`.

<code>behavior=alternate</code>	Содержимое элемента перемещается между правым и левым краем элемента.
<code>behavior=scroll</code>	Содержимое элемента перемещается в направлении, заданным параметром <code>direction</code> , затем скрывается за пределами области, после чего начинает движение с начала.
<code>behavior=slide</code>	Содержимое элемента перемещается в направлении, заданным параметром <code>direction</code> , доходит до края области и останавливается.

Атрибут direction

Содержимое контейнера `<MARQUEE>` может перемещаться в четырех направлениях, не только влево и вправо, но также вверх и вниз. Значение по умолчанию: `left`

<code>direction=down</code>	Движение сверху вниз.
<code>direction=left</code>	Движение справа налево.
<code>direction=right</code>	Движение в правую сторону.
<code>direction=up</code>	Движение вверх.

Атрибут loop

Устанавливает, сколько раз скроллировать содержимое элемента `MARQUEE`.

Значение атрибута может быть натуральным числом или `-1` (для бесконечного воспроизведения движения). Значение по умолчанию: `-1`.

Атрибуты height u width

Элемент `MARQUEE` отображается как прямоугольник, ширина и высота которого устанавливаются, соответственно, с помощью параметров `width` и `height`. Прокрутка содержимого будет происходить в пределах заданных размеров. Допускается использовать значения в пикселах или процентах. Значение по умолчанию: ширина – `100%`, высота – `12` пикселей.

Атрибут bgcolor

Устанавливает цвет фона под содержимым элемента `MARQUEE`.

По умолчанию совпадает с цветом фона веб-страницы.

Специальные символы

Поскольку символы "<" и ">" воспринимаются браузером как начало и конец тега, то возникает вопрос: как показать эти символы на экране?

Для этих некоторых других символов предусмотрено специальное кодирование.

Код строится из символа "&" (амперсанд), фиксированного имени и завершающего символа ";"

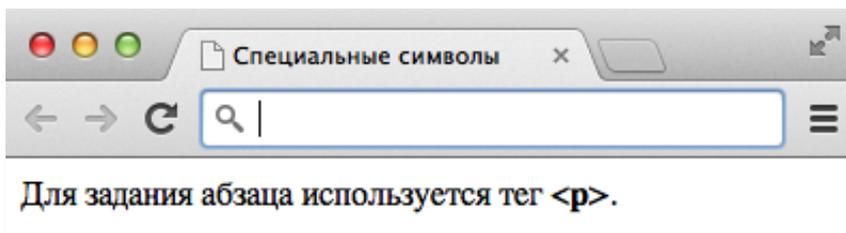
&имя;

Не для всех символов в стандарте HTML предусмотрены мнемонические имена, но символ можно закодировать по его числовому коду:

&#код;

Примеры:

меньше	<	<	<
больше	>	>	>
амперсанд	&	&	&
копирайт	©	©	©



Фрагмент HTML- программы:

```
<BODY>
```

```
Для задания абзаца используется тег &lt;<B>p</B>&gt;
```

```
</BODY>
```