Лекция1. HTML - язык гипертекстовой разметки

Знакомство с HTML.

HTML-файл имеет расширение htm или html. Браузер определяет тип файла именно по его расширению. HTML-файл должен подчиняться определенным правилам, описанным в стандарте HTML. Рассмотрим в качестве примера произвольный HTML-файл.

Всякий HTML-файл состоит из *элементов.* Элемент начинается с ярлыка. Что такое ярлык HTML? *Ярлык* - это слово, помещенное в угловые скобки. Ярлыки бывают *открывающими* и *закрывающими.* Закрывающий ярлык отличается от открывающего тем, что он начинается с символа косая черта / сразу после угловой скобки. Ярлык - смысловой аналог английского слова tag. Не каждый элемент HTML с необходимостью должен включать в себя все указанные компоненты, но открывающий ярлык должен присутствовать всегда. Открывающий ярлык может иметь атрибуты. Эти атрибуты описывают свойства элемента, к которому относится открывающий ярлык.

Различные элементы HTML могут иметь разную структуру. Это могут быть и текст, и ссылки на рисунки, и программы, и вставляемые элементы. Подробное описание HTML-элементов можно найти в документации. Сейчас для нас важно, что всякий HTML-файл должен содержать в себе элемент <HTML>. . .</HTML>. Этот элемент включает в себя всю информацию, содержащуюся в HTML-файле.

HTML-страничка может быть создана при помощи простейшего текстового редактора. Откроем Блокнот (Notepad) и напишем:

```
<HTML> Это простая HTML-страница.</HTML>
```

Запомним файл под именем first.htm и загрузим файл в браузер.

Как правило, в HTML-страницах помимо ярлыка <HTML> используется несколько других обязательных ярлыков. Среди них ярлык <head> (и обязательный </head>) для написания заголовка HTML-странички. Здесь можно указать название странички. Название должно быть помещено между ярлыками <titie> и </titie>. В стандартных браузерах заголовок будет показан в верхней части главной рамки основного окна.

```
<HTML>
<head>
<title>
На этой станице в HTML-файле содержатся все ярлыки
</title>
</head>
<body>
Все основные ярлыки присутствуют здесь.
</body>
</HTML>
```

В файле firstl.htm мы использовали заголовок, расположив его в верхней части файла между ярлыками <head> и </head>. В заголовке может быть указано название страницы с помощью ярлыков <title> и </titie>. В блоке заголовка часто используется мета-комментарий <meta>.. .</meta>. После заголовка мы использовали другую пару ярлыков: <body> и </body>, между которыми располагается основной текст документа.

Текст

Главным компонентом содержимого HTML-страниц является текст. Для форматирования текста в HTML используются специальные ярлыки. Наличие пробелов, табуляций, переводов строки и других специальных символов в тексте документа не влияет на вид представления текста

в окне браузера.

Для отделения одного параграфа от другого используется ярлык . В начале каждого параграфа мы будем ставить ярлык . Закрывающий ярлык необязателен. Для оформления иерархии заголовков в HTML используются ярлыки <h1>, <h2>, <h3> и т. д. Парные закрывающие ярлыки необходимо помещать сразу после окончания текста соответствующего заголовка.

```
<HTML>
<head>
<title>
Простой форматированный
текст
</title>
</head>
<body>
<p>
<h1>Глава 1. Форматирование
текста </h1>
<p>
<h2>Что такое HTML?</h2>
</HTML>
```

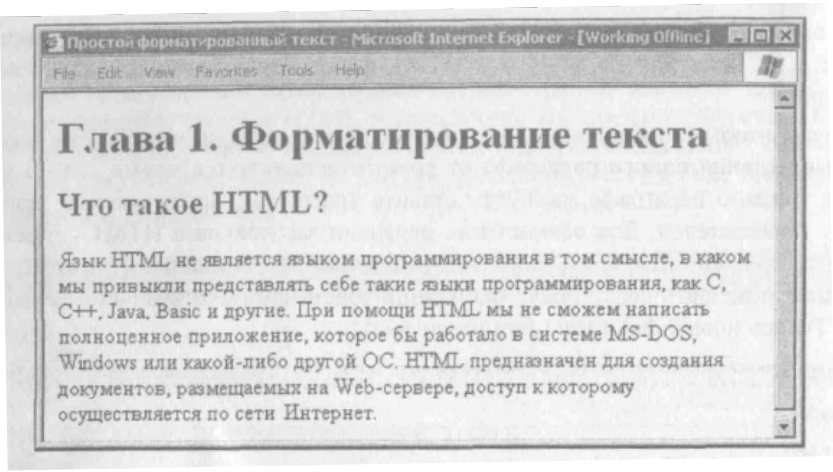


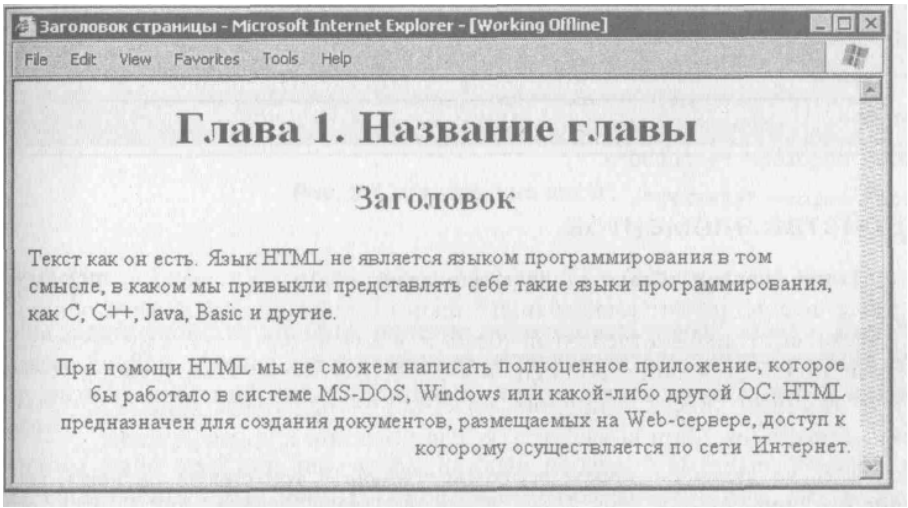
Рис. 2.4. Форматированный текст

Свойства элементов

В ярлыках можно указать значения свойств соответствующих элементов. Давайте выровняем заголовки нашего текста по центру, а фрагмент основного текста выделим в отдельный абзац и выровним по правому краю. Для этого используется свойство `align` в ярлыках <h1> и <h2>, а также в ярлыке в обычном тексте основного содержания параграфа (рис. 2.5.). Измененный файл text2.htm выглядит так, как показано в примере ниже.

Существует множество различных свойств у многих ярлыков. Так, для ярлыка <body> мы можем задать цвет фона - свойство `bgcolor`. Цвет фона можно указать либо в виде стандартного слова, обозначающего цвет, например, `white` или `yellow`, либо в виде шестнадцатеричной величины `RRGGBB`, где параметры `RR` показывают интенсивность красного цвета, `GG` - интенсивность зеленого, `BB` - интенсивность синего цвета в аддитивной модели цветовосприятия. Белому цвету соответствует значение `#FFFFFF = "White"`.

В качестве примера изменим вид нашей странички на "негатив". Для этого необязательно задавать цвет для каждого ярлыка по отдельности. Здесь можно использовать общий для всего фрагмента текста ярлык <font>, с помощью которого задаются параметры используемого шрифта, в том числе его цвет. Файл text3.htm выглядит следующим образом.



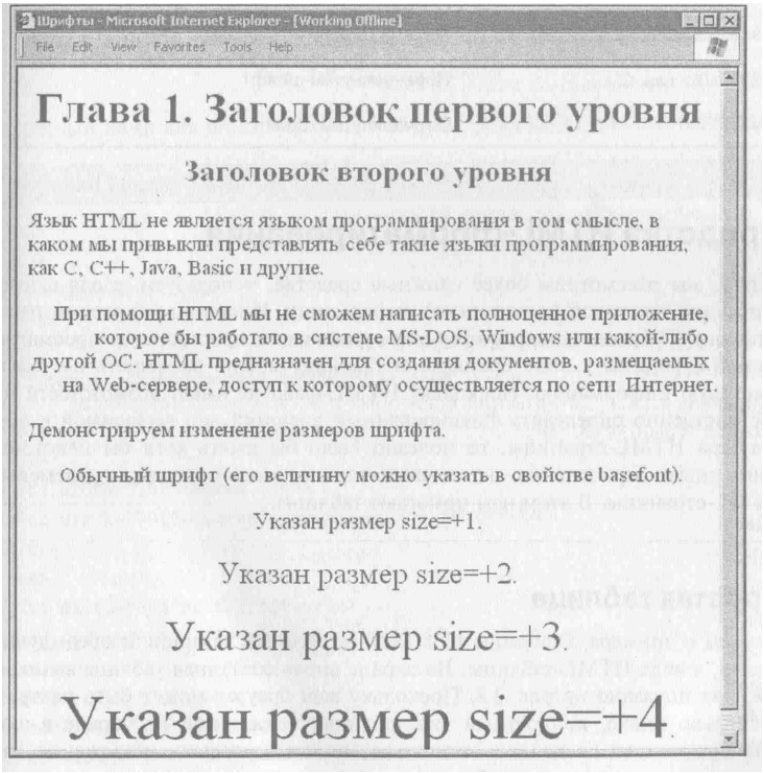

```
<HTML>
<head>
```

```
<title>
Цвет и текст
</title>
</head>
<body bgcolor="#000000">
<font color="#FFFFFF">
<p>
<h1 align=”center”> Глава 1. Название главы </h1>
<p>
<h2 align=”center”> Заголовок второго уровня </h2>
<p align=”right”> Текст.
</font>
</body>
</HTML>
```

Шрифт

Что бы задать свойства шрифта, полезно использовать ярлык `<font>`. Внутри угловых скобок ярлыка можно использовать различные свойства шрифта: его цвет, размер, внешний вид. Обязательно использовать закрывающийся ярлык `</font>`.

Изменим файл text2.htm так, чтобы каждый параграф выглядел по свему, имел свой собственный цвет. В конце файла продемонстрируем, как можно изменять размер шрифта.



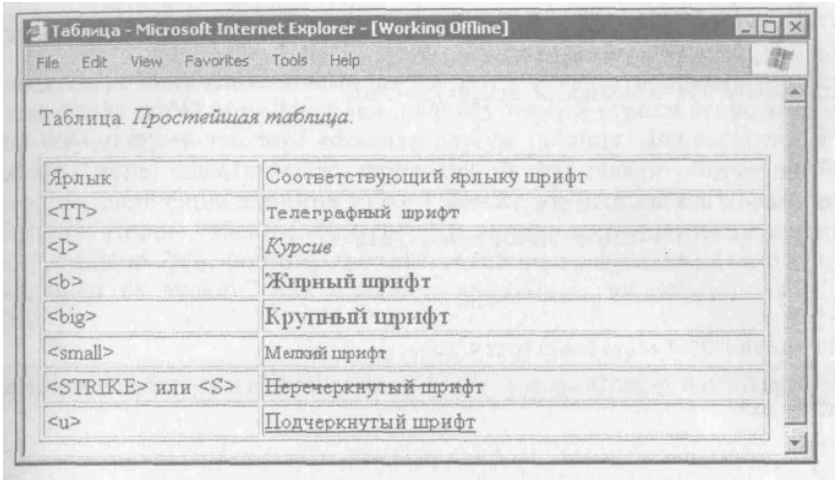

```
<HTML>
<head>
<title>
Шрифты
</title>
</head>
<body bgcolor="white">
<p>
<font color=”blue”>
<h1 align=”center”> Глава 1.
Название главы </h1>
</font>
<p>
<font color=”green”>
<h2 align=”center”> Заголовок
второго уровня </h2>
</font>
<font color=”maroon”>
<p align=”right”> Текст.
</font>
<p>
Демонстрируем изменение
размеров шрифта.
<p align=”center”> Обычный шрифт (его величину можно указать в
basefont).
<p align=”center”> <font size=+1> Указан размер size=+1. </font>
<p align=”center”> <font size=+2> Указан размер size=+2. </font>
</body>
</HTML>
```

Средства HTML-форматирования

Сейчас мы рассмотрим более сложные средства, используемые для форматирования текстовой информации. И начнем мы с рассмотрения таблиц. Таблицы используются для организации пространства просматриваемой страницы в окне браузера. В таблицах можно размещать не только текстовую информацию. Поскольку HTML-файл не имеет возможности непосредственно определять фиксированный внешний выводимой в окне браузера HTML-страницы, то полезно было бы иметь хотя бы некоторые примерные ограничения на относительное местоположение элементов HTML-страницы. В этом нам помогают таблицы.

Простая таблица

Начнем с примера. Оформим таблицу соответствия шрифтов и соответствующих им ярлыков.




```
<HTML>
<head>
<title>Таблица</title>
</head>
<body>
<p> Таблица. <i>Простейшая таблица</i>.</p>
<table border="1" width="100%">
 <tr>
  <td width="30%"> Ярлык </td>
  <td width="70%”> Соответствующий ярлыку шрифт</td>
 </tr>
 <tr>
  <td width="30%">&lt;TT&gt;</td>
  <td width=”70%”;<TT>Телеграфный шрифт</td>
 </tr>
 <tr>
  <td width="30%">&lt;I&gt;</td>
  <td width="70%"><i>Курсив</i></td>
 </tr>
 <tr>
  <td width="30%">&lt;b&gt;</td>
  <td width="70%"><b>Жирный шрифт</b></td>
 </tr>
 <tr>
  <td width="30%">&lt;big&gt;</td>
  <td width="70%"><big>Крупыный шрифт</big></td>
 </tr>
 <tr>
  <td width="30%">&lt;small&gt;</td>
  <td width="70%"><small>Мелкий шрифт</small></td>
 </tr>
 <tr>
```

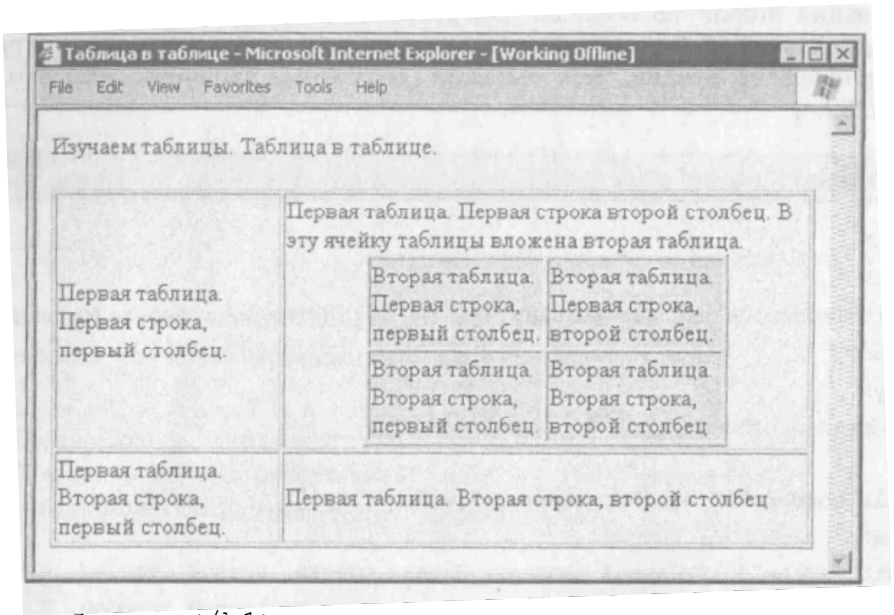
```
  <td width="30%">&lt;STRIKE&gt;  или  &lt;s&gt;</td>
  <td width="70%"<s>Перечеркнутый шрифт</s><td>
 </tr>
 <tr>
  <td width="30%">&lt;u&gt;</td>
  <td width="70%"><u>Подчеркнутый шрифт</u></td>
 </tr>
 </table>
 </body>
 </HTML>
```

Что нового мы видим с тексте этого файла? Появился ярлык <table>. Его присутствие говорит о том, что между этим открывающим ярлыком и соответствующим ему закрывающим ярлыком </table> располагается таблица. Таблица состоит из строк и столбцов. Каждая строка начинается ярлыком <tr> и заканчивается ярлыком </tr>. Каждый столбец описывается внутри строки с помощью открывающего ярлыка <td> и закрывающего ярлыка </td>. Вот и все основные компоненты таблицы.

В ярлыках <table>, <tr>, <td> можно указать свойства. В нашем случае мы задали параметр ширины таблицы в виде width="100%". Это означает, что таблица займет всю ширину окна браузера. В параметрах столбцов мы также указываем ширину столбца в процентах от общей ширины таблицы. Если изменить значения ширины столбцов, внешний вид таблицы изменится.

Вложенные таблицы

Таблицы могут быть вложенными друг в друга. В качестве примера рассмотрим код файла table3.htm. Здесь мы в ячейку первой строки второго столбца первой объемлющей таблицы вставили другую таблицу. Для того чтобы одна таблица четче выделялась на фоне другой, мы задали разные значения параметров цвета фона для этих таблиц. Как выглядят вложенные таблицы, показано на рис. 2.9.




```
<HTML>
<HEAD>  <TITLE>  Заголовок страницы </title> </head>
<BODY BGCOLOR="#FFA4FF">
< H1 Align="Center"> Изучаем таблицы </h1>
<table width="100%" border="1" bgcolor="#E8E8E8">
 <Tr>
  <Td ALIGN="center" width="30%"> Первая таблица. Первая строка, первый столбец.</Td>
  <Td width="70%">
   <Table Width="70%" Border="3" BORDERCOLOR="#000000" Align="center" Bgcolor="#FFFF00">
    <Tr>
     <Td width="50%" BORDERCOLOR="#FFFF00"> Вторая таблица. Первая строка. Первый столбец.</Td>
     <Td width="50%"> Вторая таблица. Первая строка. Второй
```

```
столбец.</Td>
    </Tr>
    <Tr>
     <Td   width="50%">   Вторая   таблица.   Вторая   строка.   Первый
столбец.</Td>
     <Td   width="50%">   Вторая   таблица.   Вторая   строка.   Второй
столбец.</Td>
    </Tr>
   </Table>
   </Td>
 </Tr>
 <Tr>
  <Td width="30%" ALIGN="center"> Первая таблица. Вторая строка. Первый
столбец.</Td>
  <Td width="70%" ALIGN="center"> Первая таблица. Вторая строка. Второй
столбец.</Td>
 </Tr>
</Table>
</body>
</html>
```

Вложенные таблицы позволяют со значительной степенью гибкости располагать элементы на страничке. Если установить параметр толщины рамки таблицы `border`, равным нулю, то рамки станут не видимы и пользователь не будет подозревать о том, что страница скомпонована при помощи таблиц.

Свойства элемента `<table>`

`align`	Выравнивание, относительное положение таблицы внутри элемента, в котором она находится
`bgcolor`	цвет фона таблицы
`width`	Ширина таблицы
`cols`	Количество столбцов в таблице
`border`	Толщина рамки таблицы
`cellspacing`	Пробел между ячейками
`cellpadding`	Пробел от границы ячейки до содержимого ячейки

Это лишь малая часть свойств элемента `<table>`. Подробная информация содержится в справочных системах и в документации по языку HTML.

Объединение ячеек в таблице

Если вам приходилось работать с текстовым процессором Word, то вы знаете, что в таблице несколько соседних ячеек можно объединить. HTML позволят производить такое объединение ячеек. Для объединения ячеек используются параметры `rowspan` и `coispan`. Им присваивается целое число, равное количеству объединенных ячеек. Эти параметры задаются в ярлыке `<td>`.

Ссылки

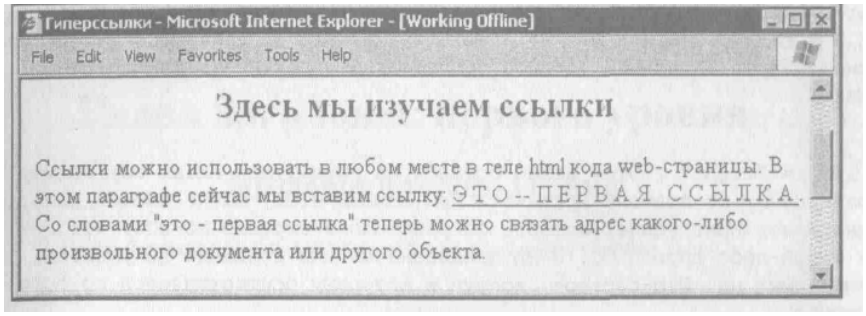
Наличие гиперссылок - одно из важных свойств HTML. Собственно механизм ссылок и делает язык HTML тем средством, которое необходимо для удобной работы в сети Интернет. Механизм ссылок может быть различным. Рассмотрим на примерах то, как работают простейшие ссылки.

Текстовая ссылка

С помощью HTML мы имеем возможность организовать ссылку из любой части документа. Для ссылки используется ярлык, называемый *якорем,* anchor. В HTML, вероятно по той причине, что ссылка - это один из наиболее важных элементов Web-страниц, используется максимально сокращенный вариант слова anchor, от этого слова осталась лишь начальная буква. Для создания ссылки используется открывающий ярлык <a> и закрывающий ярлык .

Между ярлыками <a> и расположен фрагмент HTML-странички, с которым связана ссылка. Это может быть текст, это может быть таблица, это может быть рисунок.

Для начала мы создадим простую текстовую ссылку. Откроем какой-либо текстовый редактор и введем текст:

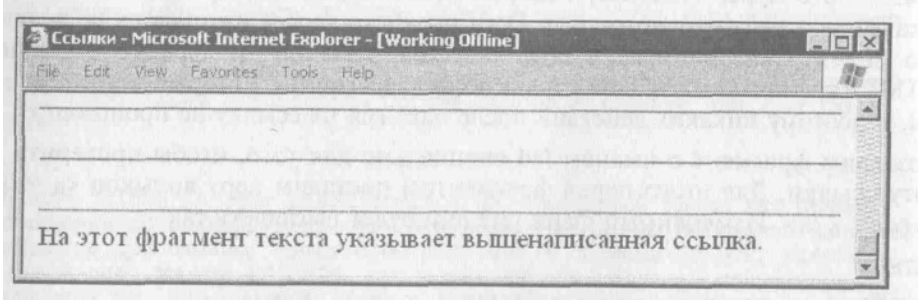



```
<HTML>
<head>
<title>
Гиперссылки
</title>
</head>
<body bgcolor= "#FFFFFF">
<p>
<h1 align=”center”> Глава1. </h1>
<p>
<h2 align=”center”> Изучаем ссылки. </h2>
<p>
Ссылки можно использовать в любом месте в теле HTML кода.
В этом параграфе мы вставим ссылку: <a href=”#fed”> Это первая
ссылка.</a>
<p align= "right"><br>При помощи HTML мы не сможем написать полноценное
при ложение, которое бы работало в системе MS-DOS, Windows или какой-
либо другой ОС. HTML предназначен для создания документов, размещаемых
на Web-сервере, доступ к которому осуществляется по сети Интернет.
</body>
</HTML>
```

Адрес ссылки

Ссылки создаются для того, чтобы при щелчке на ней можно было просмотреть документ, на который она указывает. Однако если в предыдущем примере мы щелкнем на ссылке, то ничего не произойдет. Дело в том, что в свойстве `href` для нашей ссылки указан несуществующий адрес. В свойстве `href` указывается адрес документа, на который указывает ссылка. Href - это hyper reference, гиперссылка. В прошлом примере мы указали в качестве адреса параметр `#fed`. Если параметр адреса начинается со знака *#,* это значит, что данная ссылка является ссылкой на фрагмент текущей HTML-страницы. На нашей странице нет фрагмента, который бы носил имя `fed`, и поэтому никаких действий после нажатия на ссылку не происходит.

Создадим фрагмент с именем `fed` специально для того, чтобы проверить работу ссылки. Для этого перед фрагментом поставим пару ярлыков . Измененный файл ref2.htm будет выглядеть так:




```
<HTML>
<head>
<title>
Гиперссылки
</title>
</head>
<body bgcolor=
```

```
"#FFFFFF">
<p>
<h1 align=”center”> Глава1. </h1>
<p>
<h2 align=”center”> Изучаем ссылки. </h2>
<p>
Ссылки можно использовать в любом месте в теле HTML кода.
В этом параграфе мы вставим ссылку: <a href=”#fed”> Это первая
ссылка.</a>
<p align= "right"><br>При помощи HTML мы не сможем написать полноценное
при ложение, которое бы работало в системе MS-DOS, Windows или какой-
либо другой ОС. HTML предназначен для создания документов, размещаемых
на Web-сервере, доступ к которому осуществляется по сети Интернет.
<a name= "#fed"></a>. На этот фрагмент текста указывает ссылка.
</body>
</HTML>
```

В приведенном выше примере мы использовали ярлык `<br>`. Это непарный ярлык, для него не требуется указания закрывающего ярлыка. Этот ярлык указывает на необходимость перехода к новой строке в месте расположения ярлыка. Далее в этой главе мы рассмотрим и другие ярлыки, с помощью которых происходит управление и форматирование текста.

Чтобы организовать ссылку на какой-либо документ, расположенный в сети Интернет, достаточно указать полный адрес этого документа в параметре `href` ссылки. Так, ссылка

```
<a href= "http://www.yahoo.com"> Это поисковая система Yahoo</a>
```

указывает на главную страницу поисковой системы Yahoo.

Ссылки и рисунки

Вначале научимся вставлять рисунок в HTML-файл. Внедрить файл с изображением в HTML-страницу несложно. Для этого используется ярлык `<img>`. Изображение - по-английски image (образ), сокращенный вариант этого слова используется в качестве HTML-ярлыка. В случае, если пользователь имеет очень медленную связь с Интернетом, он может отключить автоматическую загрузку изображений в браузер. Тогда вместо картинки мы можем показать пользователю текст. Для этого нужно в ярлык `<img>` вставить свойство `alt`, которому присваиваем текстовую строку. Именно эта строка будет показана в окне браузера вместо рисунка, если рисунок по тем или иным причинам не был загружен:

```
<img src="picl.jpg" alt="Текст, выводимый вместо изображения">
```

Окно браузера будет выглядеть примерно так, как показано на рис. 2.14.

Фон страницы

Рисунки можно использовать не только в качестве элементов, объектов страницы, документа, но и в виде фона, на котором будет расположена страница. Файл рисунка с фоном указывается в ярлыке `<body>`:

```
<body bgcolor="white" background="back.gif">
```

Если одной копии рисунка не хватает, чтобы покрыть весь фон документа, т. е. если размер окна браузера больше размера рисунка, то рисунок фона воспроизводится столько раз, сколько необходимо для того, чтобы покрыть все окно браузера.

Свойства рисунков

В ярлыке `<img>` можно задать значения свойств (атрибутов), указанных в табл. 2.4.

Таблица 2.4. *Свойства элемента* `<img>`

scr	URL-адрес рисунка
alt	текст, который будет показан вместо картинки, если картинка не загружена
align	выравнивание
height	высота в пикселах
width	ширина в пикселах
border	ширина границы картинки
hspace	горизонтальный отступ
vspace	вертикальный отступ
usemap	URL-адрес карты ссылок
ismap	указатель использования серверной карты ссылок

Свойство align позволяет определить местоположение картинки в окне браузера. Данное свойство может принимать следующие значения:

- bottom
- middle
- top
- left
- right

Параметр middle соответствует выравниванию центра рисунка по середине вертикали от текущей базовой линии. Значению top соответствует выравнивание по верхней границе рисунка с учетом верхней границы текущего текста.

Параметры left и right описывают плавающие рисунки. Расположение текста и вставляемого рисунка в потоке HTML-кода не имеют значения. Рисунки являются плавающими элементами страницы, их взаимное расположение и расположение по отношению к тексту определяются параметрами left и right. Обычно текст "обтекает" изображения. Для создания отступов можно использовать рамки. Ширина рамки задается с помощью параметра border:

```
<img src="picl.jpg"  align="left" border="5">
<img src="picl.jpg"  align="right" border="25">
```

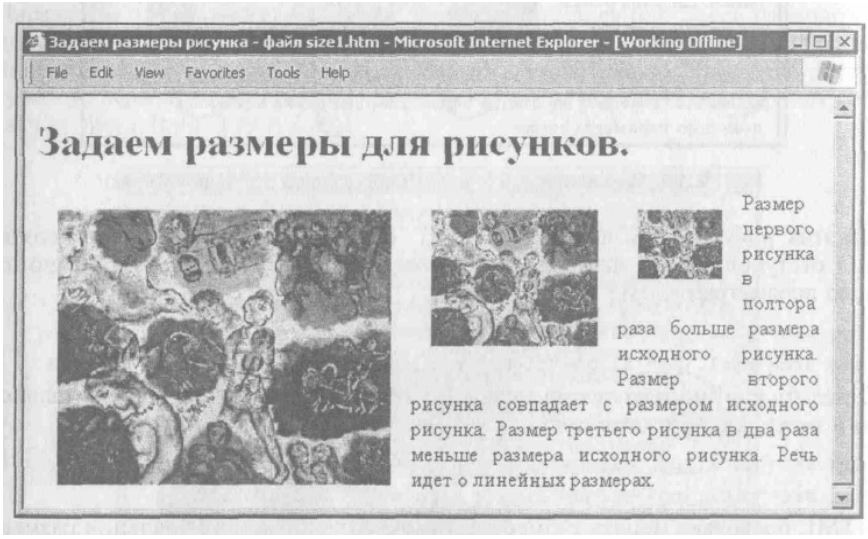
Отделить картинки от текста можно с указанием параметров горизонтального и вертикального отступов:

```
<img src="picl.jpg" align="left" vspace="15" hspace="5">
<img src="picl.jpg" align="right" vspace="5" hspace="15">
```

HTML позволяет менять размер картинок.

```
<HTML>
<head>
<title>
Задаем размеры рисунка - файл sizel.htm
</title>
</head>
<body bgcolor="white">
<h1>Задаем размеры для рисунков.</h1>

<img src="picl.jpg" align="left" vspace="15" hspace="15" width="450">
```




```
<img src="picl.jpg" align="left" vspace="15" hspace="15">
```

```
<img src="picl.jpg" align="left" vspace="15" hspace="15" width="150">
<p align="justify">Размер первого рисунка в полтора раза больше размера
исходного рисунка.
Размер второго рисунка совпадает с размером исходного рисунка.
Размер третьего рисунка в два раза меньше размера исходного рисунка.
Речь идет о линейных размерах.
</body> </HTML>
```

Изображения-карты

Создадим простой HTML-файл `map.html`:

```
<map name= "mapname">
<area [shape="shape"] coords='x,y, ...'  [href=URL|nohref]>
</map>
```

Свойства:

`shape` - форма определяемой области;
`coords` - местоположение области и ее размер;
`href` - URL, на который ссылается область;
`nohref` - область является "мертвой зоной".

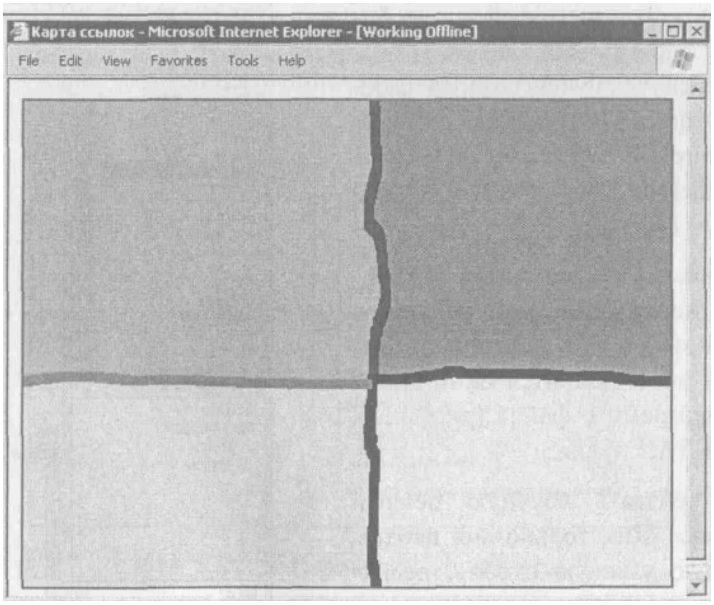
Форма `shape` может быть задана одним из четырех способов:

`default` - вся область, которая не определена с помощью других значений, приведенных ниже;
`rect` - прямоугольная область;
`circle` - круг;
`poly` - многоугольная область.

Прямоугольная область определяется четырьмя координатами:
левая; верхняя; правая; нижняя.

Для круга координат три:
горизонтальная координата центра;
вертикальная координата центра;
радиус.

Для многоугольника координаты перечисляются в последовательности `x1,y1,x2,y2,x3,y3....`




```
<HTML>
<head>
<title>Карта  ссылок</title>
</head>
<body>
<img    src="pic.jpg"    usemap=
"#mapname">
<map name="mapname">
<area                  shape=rect
coords="0,0,275,220"
href="ref1.html">
<area shape=rect coords="275,0,450,220" href="ref2.html">
<area shape=rect coords="0,220,275,340" href="ref3.html">
```

```
<area shape=rect coords="275,220,450,340" href="ref4.html">
</map>
</body>
</HTML>
```

Загрузив файл map.html в браузер, мы получим картинку, изображенную на рис. 2.26.

Каскадные таблицы стилей (Cascad Style Sheets)

Каскадные таблицы стилей были предложены w3c(WWW Consorcium) в рамках разработки спецификации HTML 3.0. Однако, реализованы в реально действующих навигаторах они были только в 1997 году.

Идея положенная в основу таблиц достаточно проста. HTML-документ - это множество вложенных в друг друга контейнеров, каждый из которых имеет свои свойства по представлению информации. Многие контейнеры можно сгруппировать в классы однотипных контейнеров, например, заголовки или параграфы. Свойства контейнера, перечисляются в качестве атрибутов тага начала контейнера. При этом у большинства контейнеров, начиная с версии HTML 3.0 набор этих атрибутов типизирован.

Контейнер STYLE(<style type="...">.....</style>) служит для определения таблицы описания стилей. `<style type="text/css">`

```
<html>
<head>
style type="text/css"><!--
p {color:blue; font-family: times; font-size:10pt;}
h1 {color:black; font-size:12pt; font-style:Arial; text-align: center;} -->
</style>
</head>
<body>
<h1>Test Style Shits in Communicator</h1>
<p> This is a first part of the document
</body>
</html>
```

Контейнер LINK в контексте описателей стилей применяется для определения внешнего файла с описаниями стилей для данного документа. Например, внешний файл может содержать следующее описание стилей:

```
/* contents of the external style sheets file  css.htm*/
p {color:blue; font-family: times; font-size:10pt;}
h1 {color:black; font-size:12pt; font-style:Arial; text-align: center;}
/* the end of style sheets definition */
```

Для применения этого описателя стилей в заголовок документа необходимо включить следующий таг:

```
<html>
<head>
<link REL=STYLESHEET TYPE="text/css" HREF=http://localhost/css.htm>
</head>
<body>
The body of the document should be here.
</body>
</html>
```

Контейнер SPAN применяется для переопределения стиля отображения текущего фрагмента текста и в некотором смысле аналогичен контейнеру FONT.

```
<HTML>
<HEAD>
</HEAD>
<BODY bgcolor=lightyellow>
<center>
```

```
        <p><span class=" color:red; font-size: 24pt;">C</span>обществу глобальных
компьютерных сетей Internet в 2000 году исполнилось 30 лет.
        </BODY>
        </HTML>
```

В данном примере, контейнер SPAN применен сразу после тага начала параграфа <p>, что позволяет выделить первую букву в отображаемом абзаце:

Свойства контейнеров, управляемые описателями стилей

Первую группу свойств составляют свойства шрифтов:

`font-size, font-family, font-weight, font-style.`

Вторую группу свойств составляют свойства текста:

`line-height, text-decoration, text-transform, text-align, text-indent.`

Третью группу свойств составляют свойства блоков текста:

`margin-left, margin-right, margin-top, margin-bottom, margin, padding-top, padding-right, padding-bottom, padding-left, paddings, border-top-width, border-bottom-width, border-left-width, border-right-width, border-width, border-style, border-color`

Четвертую группу составляют описатели цвета фона и цвета текста:

`color, background-image, background-color.`

Фреймы

Фреймы - это английское слово, в переводе означающее *рамки.* HTML-страница, в которой используются фреймы, выглядит разделенной на несколько частей, которые окаймлены границами, как рамками.

Создание и использование фреймов

Часто возникает вопрос о том, как могут быть связаны друг с другом, казалось бы, совершенно разные вещи, такие как фреймы и язык JavaScript. Не будем спешить и остановимся вначале на рассказе о том, что представляют собой фреймы и для чего они используются. После этого рассмотрим вопрос о том, как могут быть взаимосвязаны фреймы и язык JavaScript.

Окно браузера может быть разделено на несколько фреймов. Каждый фрейм показывает свой документ, как правило, это обычный HTML-документ. Для того чтобы создать фреймы, мы пользуемся двумя ярлыками, а именно ярлыком `<frameset>` и ярлыком `<frame>`. Страница, разбитая на два фрейма, может быть описана при помощи HTML следующим образом:

```
<HTML>
<frameset rows="50%,50%">
 <frame src="pagel.htm" name="framel">
 <frame src="page2.htra" name="frame2">
</frameset>
</HTML>
```

Этот код порождает два фрейма. В ярлыке `<frameset>` указано свойство `rows` (rows - это ряды). Это свойство указывает на то, что два фрейма расположены рядами, *друг над другом.* Верхний фрейм загружается страницей pagel.htm, а нижний фрейм - страницей page2.htm.

Если вы захотите расположить фреймы столбцами, то вместо свойства `rows` в ярлыке `<frameset>` укажите `cols`. Процентные величины "50%,50%" указывают относительные размеры окон фреймов. Если вы не хотите утруждать себя вычислениями того, каким должен быть размер второго окна, чтобы в целом получилось 100%, то можно указать размер окна в пикселах, при этом символ % опускается.

Каждый фрейм имеет свое уникальное имя, которое задается при помощи свойства name в ярлыке `<frame>`. При помощи этого имени фрейм легко доступен при описании на JavaScript.

Можно использовать несколько последовательных описаний фреймов при помощи ярлыков `<frameset>`, при этом фреймы могут быть расположены друг в друге. Вот пример такого

расположения фреймов:

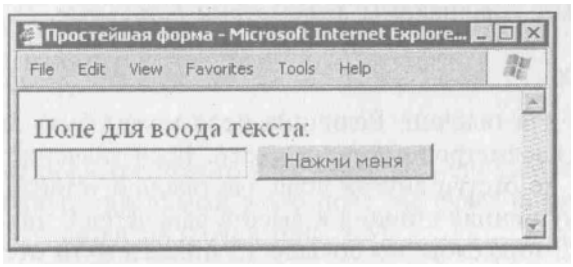
```
<frameset cols="50%,50%">
 <frameset rows="50%,50%">
  <frame src="cell.htm">
  <frame src="cell.htm">
 </frameset>
 <frameset rows="33%,33%,33%">
  <frame src="cell.htm">
  <frame src="cell.htm">
  <frame src="cell.htm">
 </frameset>
</frameset>
```

С фреймами можно осуществлять довольно замысловатые операции, если использовать язык сценариев. Динамическому HTML посвящена следующая лекция.

Формы

Создадим простейший файл с формой forml.html.

```
<HTML>
<head>
<title>Простейшая форма</title>
</head>
<body>
<form name= "first">
Поле для ввода текста: <br>
<input type=text  name =  "textl">
<input type="button" name="buttonl" value="Нажми меня">
<p>
</form>
</body>
</HTML>
```



Загрузим файл в браузер, получится картинка, показанная на рис. 2.34.

В коде HTML мы использовали ярлык `<form>`, он открывает форму. Форма - это контейнер, содержащий элементы, образующие тело формы. Требуется закрывающий ярлык. В своем примере мы указали имя для формы - это значение параметра name.

Элементы, составляющие тело формы

Важными компонентами формы являются поля для ввода текста и кнопки. В нашем случае поле для ввода текста описано при помощи ярлыка

```
<input type=text name = "textl">
```

Здесь используется элемент `<input>`. Большинство элементов тела формы - это элементы `<input>`. Элементы `<input>` бывают разнообразными, все зависит от значения свойства `type`. В случае, когда `type=text`, элементом будет текстовое поле для ввода текста (однострочного)..

Атрибуты свойства *type*

Атрибут type может принимать следующие значения:

- button;
- checkbox;
- file;
- hidden;
- image;

- password;
- radio;
- reset;
- submit;
- text.

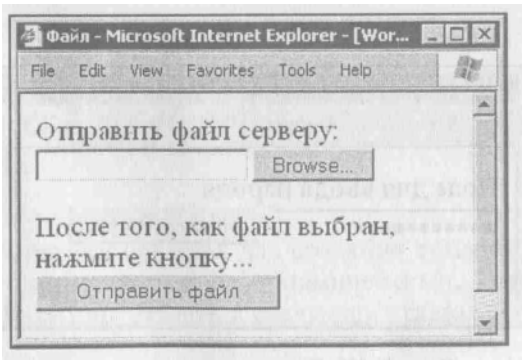
С двумя из них мы уже знакомы, рассмотрим остальные.

Тип *Checkbox*

Описывает поле для галочки. Величина поля может быть задана путем указания значений параметров `height` и `width`. Если значение оказывается более 20 пикселов, то отступ внутри поля для галочки устанавливается равным 4 пикселам, а внутренняя ширина и высота равняется 8 пикселам. Если значение меньше 20 пикселов, но больше 13 пикселов, то отступ внутри поля для галочки равен половине разности от указанной ширины (или высоты) минус 13. При размерах менее 13 пикселов отступ равен 0.

Тип *File*

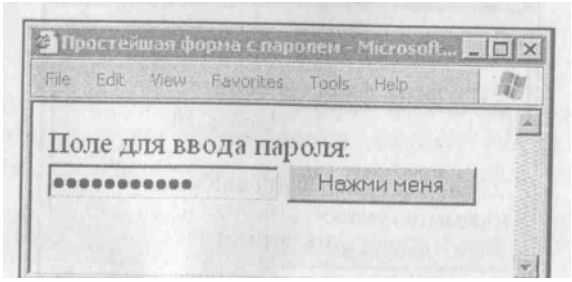
Используется для отправки файла серверу. При этом необходимо указать приложение, которое будет обрабатывать форму, указав его в атрибуте action ярлыка `<form>`. Рассмотрим это на примере файла form2.html.




```
<HTML>
<head><title>Файл</title></head>
<body>
<form name= "form"
action="scripts/form.php"
enctype="multipart/form-data"
method="post">
Отправить  файл серверу:   <br>
<input type=file name = "file1">
<p> После того, как файл выбран, нажмите
кнопку.
<input type= "button" name = "buttonl" value = "Отправить файл">
<p>
</form>
</body>
</HTML>
```

Кроме атрибута `action`, при посылке файла серверу необходимо указать значение атрибута `enctype` в виде `multipart/form-data`. Методом отправки может быть либо `get`, либо `post`. Окно браузера с формой для отправки файла серверу показано на рис. 2.35 и 2.36.

При отправке форм сервер получает информацию в виде пар, каждая из которых представляет собой имя и соответствующее этому имени значение. Эти пары обрабатываются сервером.

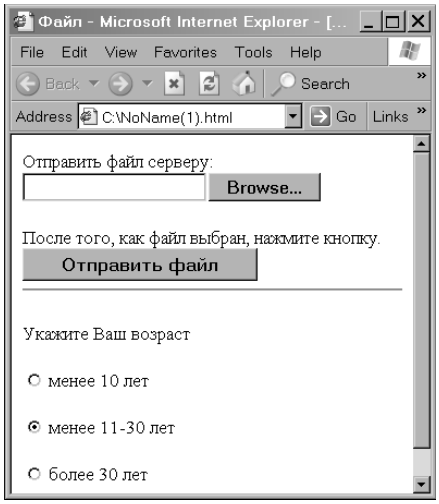


Тип *Password*

Можно в качестве поля текстового ввода данных использовать поле для ввода пароля. Тогда текст, вводимый в это поле, не будет виден в окне браузера, вместо него выводятся звездочки (рис. 2.37).

Тип *Radio*

Тип <input type=radio> создает переключатели, которые используются для ограничения пользовательского выбора. Пользователь может выбрать лишь один вариант из нескольких. Для этого несколько переключателей объединяют в группу, все переключатели одной группы имеют одинаковое имя name. При посылке формы значение выбранного переключателя отсылается, как значение имени. Для отметки того или иного переключателя, который будет использоваться по умолчанию, служит свойство checked (см. пример файла form3.html).




```
<HTML>
<head><title>Файл</title></head>
<body>
<form name= "form"
action="scripts/form.php"
enctype= "multipart/form-data"
method="post">
Отправить файл серверу:   <br>
<input type=file name = "file1">
<p> После того, как файл выбран, нажмите кнопку.
<input type= "button" name = "buttonl" value = "Отправить файл">
<hr>
<p> Укажите Ваш возраст
<p> <input type=radio name="radio"> менее 10 лет
<p> <input type="radio" name="radio" checked> менее 11-30 лет
<p> <input type="radio" name="radio" > более 30 лет
<p>
</form>
</body>
</HTML>
```