МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Т.В. Зудилова, М.Л. Буркова

Web-программирование HTML

Учебное пособие

Санкт-Петербург

2012

УДК 004.655, 004.657, 004.62

Т.В. Зудилова, М.Л. Буркова

Web-программирование HTML - СПб: НИУ ИТМО, 2012. – 70 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине “Web-программирование PHP - технологии”.

Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров направления: 210700 Инфокоммуникационные технологии и системы связи.

Рекомендовано к печати Ученым советом факультета инфокоммуникационных технологий, протокол №4 от 13 декабря 2011г.

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Оглавление

Введение

В результате курса, проводимого под руководством преподавателя, студенты познакомятся с:

- технологиями и концепцией создания статических web-страниц;
- принципами разработки структуры web-страниц;
- основными элементами языка;
- средствами оформления страниц;
- способами передачи данных от удаленного пользователя на сервер.

Цель курса

По окончании данного курса студенты смогут:

- создавать несложные статические сайты и наполнять их форматированным и структурированным содержимым;
- иметь представление об основных конструкциях языка и использовать эти знания для изучения более сложных средств разработки Web-приложений;
- знать основные понятия сети Internet;
- использовать стилевое форматирование для оформления HTML-документов.

1. Обзор возможностей языка HTML

Термин HTML (HyperText Markup Language) означает "язык разметки гипертекстов", с помощью которого верстаются Web-страницы. В самом начале развития WWW регулировать стандарты разработки была призвана международная общественная организация, включающая в себя представителей фирм-разработчиков и исследовательских институтов — W3C (World Wide Web Consortium). Более полную информацию по этому языку можно посмотреть по адресу: http://www.w3.org.

Характеристики HTML:

- Разработан специально для Web;
- Открытый стандарт;
- В HTML включен гипертекст;
- В HTML включена поддержка мультимедиа;

Со времени создания первой версии HTML претерпел некоторые изменения. Известны спецификации 2.0, 3.0, 3.2, 4.0, 5.0. Текущую спецификацию HTML всегда можно найти на сервере W3C (http://www.w3.org/).

Все что необходимо, чтобы прочитать HTML-документ - это Web-браузер, который интерпретирует элементы HTML и воспроизводит на экране документ в виде, который ему придает автор. Основное преимущество HTML заключается в том, что документ может быть просмотрен на Web-браузерах различных типов и на различных платформах.

HTML-документы могут быть созданы при помощи любого текстового редактора или специализированных HTML-редакторов и конвертеров.

С другой стороны можно использовать специальные программы – редакторы HTML текстов.

1.1. Структура документа

Основными конструкциями языка являются тэги. Все тэги HTML начинаются с "<" (левой угловой скобки) и заканчиваются символом ">" (правой угловой скобки). Как правило, существует стартовый тэг и завершающий тэг.

<title> Заголовок документа </title>

Завершающий тэг выглядит также, как стартовый, он отличается от него прямым слэшем перед текстом внутри угловых скобок. Некоторые тэги, такие, как <hr> (тэг, определяющий горизонтальную линию), не требуют завершающего тэга.

HTML не реагирует на регистр символов, на синтаксис. Тэги либо распознаются браузером, либо нет.

Когда Web-браузер получает документ, он определяет, как документ должен быть интерпретирован. Самый первый тэг, который встречается в документе, должен быть тэгом <html>. Простейший HTML-документ будет выглядеть так:

```
<html>
  <head>
        <title>.....</title>
  </head>
  <body>
        ……..
  </body>
</html>
```

Основные элементы страницы

Заголовочная часть документа <head> .

Элементы, находящиеся внутри раздела head (кроме названия документа, записываемого с помощью раздела title), не видны на экране (во всяком случае, напрямую). Элементы, содержащиеся внутри раздела head документа, нужны для того, чтобы:

- Дать документу название.
- Определить отношения между несколькими документами.
- Дать указание браузеру создать форму для поиска.
- Добавить динамическую составляющую.

Стартовый тэг <head> помещается непосредственно перед тэгом <title>, а завершающий тэг </head> размещается сразу после окончания описания документа. Например:

```
<head>
<title> Список сотрудников </title>
</head>
```

Заголовок документа <title> .

Раздел title служит для того, чтобы дать документу название. Оно обычно показывается в заголовке окна браузера.

Название документа записывается между тэгами <title> и </title> и представляет собой текстовую строку.

Тело документа <body> .

Тело документа должно находиться между тэгами <body> и </body>. Атрибуты этого тега определяют общий облик вашего документа. Они перечислены в Таблице 1.

Таблица 1.

Атрибут	Назначение
background	Указывает на URL-адрес изображения, которое используется в качестве фонового.
bgcolor	Определяет цвет фона документа
bgproperties	Если установлено значение fixed, фоновое изображение не прокручивается.
alink	Определяет цвет активной ссылки.
link	Определяет цвет еще не просмотренной ссылки.
vlink	Определяет цвет уже просмотренной ссылки.
text	Определяет цвет текста.
topmargin	Устанавливает границу верхнего поля в пикселах.
leftmargin	Устанавливает границу левого поля в пикселах.

Комментарии.

Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются браузером. Синтаксис комментария:

<!-- Это комментарий -->

Комментарии могут встречаться в документе где угодно и в любом количестве.

Элемент address.

Этот элемент служит для идентификации автора документа. Сюда же обычно помещаются сведения об авторских правах. Этот элемент располагается либо в начале, либо в самом конце документа. Элемент address состоит из текста, помещенного между тэгами <address> и </address>. Текст, заключенный между этими тэгами, обычно показывается в окне браузера курсивом.

Цветовое оформление документа

Если цвета составных частей документа не определены автором, то используются цвета по умолчанию. Они определяются установками программы просмотра. В HTML цвета определяются цифрами в шестнадцатеричном коде. Цветовая система базируется на трех основных цветах – красном, зеленом и синем – и обозначается RGB. Для каждого цвета задается шестнадцатеричное значение в пределах от 00 до FF, что соответствует диапазону 0 – 255 в десятичном исчислении. Затем эти значения объединяются в одно число, перед которым ставится символ #. Например, число #800080 обозначает фиолетовый цвет. Для простоты в HTML определены 16 стандартных цветов, которые вместе с их шестнадцатеричными кодами приведены в Таблице 2.

Таблица 2.

Цвет	Код	Цвет	Код
Black (черный)	#000000	Silver (серебряный)	#C0C0C0
Maroon (темно-бордовый)	#800000	Red (красный)	#FF0000
Green (зеленый)	#008000	Lime (известь)	#00FF00
Olive (оливковый)	#808000	Yellow (желтый)	#FFFF00
Navy (темно-синий)	#000080	Blue (синий)	#0000FF
Purple (фиолетовый)	#800080	Fuchsia (фуксия)	#FF00FF
Teal (чирок)	#008080	Aqua (аква)	#00FFFF
Gray (серый)	#808080	White (белый)	#FFFFFF

Атрибут bgcolor отвечает за цвет фона документа. Атрибут text определяет цвет текста документа. Атрибут link используется браузером для показа ещё непросмотренных ссылок. Атрибут vlink служит для показа уже просмотренных ссылок. Как правило, их окрашивают более темным оттенком того же цвета, что и непросмотренные ссылки. Атрибут alink определяет цвет ссылки, активной в текущий момент.

Установка полей

Атрибут leftmargin устанавливает расстояние между левым краем текста и левым краем окна браузера, которое измеряется в пикселах. Атрибут topmargin служит для установки расстояния между верхним краем текста и верхним краем окна браузера.

Специальные символы

Некоторые специальные символы не входят в базовую часть таблицы кодов ASCII. К ним относятся буквы алфавитов части европейских языков, математические и некоторые другие символы. Однако они тоже могут быть введены в ваш HTML-документ при помощи символа & (амперсант) и имени символа. Например, пробел: .

1.2. Форматирование документов

Любые опубликованные материалы имеют определенную структуру.

Разделение текста на абзацы

Поместите открывающий тэг в начало каждого нового абзаца вашего текста, и программа просмотра отделит абзацы друг от друга пустой строкой. Использование закрывающего тэга необязательно.

Атрибут align, имеющий значения, представленные в Таблице 3. По умолчанию происходит выравнивание по левому краю.

Таблица 3.

Значение	**Функция**
left	Выравнивание текста по левой границе окна браузера.
center	Выравнивание по центру окна браузера.
right	Выравнивание по правой границе окна браузера.

В HTML несколько стоящих подряд тэгов <p> не дают дополнительного пространства между абзацами.

Перевод строки

Для того чтобы перейти на следующую строку в любом нужном вам месте текущей строки, в HTML существует тэг разрыва строки
. Он заставляет программу просмотра выводить стоящие после него символы с начала новой строки. В отличие от тэга абзаца, тэг
 не добавляет пустую строку.

Пример:

<body>
Эта и следующая строка
разделены тэгом конца абзаца
<p>
А эти две строки разделены тэгом
конца строки.

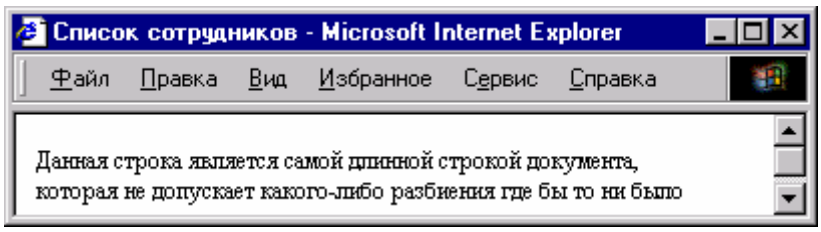
Разница видна?
</body>

Бывают случаи, когда возникает надобность в операции противоположного назначения – запретить перевод строки. Текст, заключенный между тэгами <nobr> и </nobr>, будет гарантированно располагаться в одной строке без переноса на другую.

Во избежание неприятностей с элементом <nobr> вы можете предложить браузеру читателя альтернативное место перевода строки при помощи тэга <wbr> ("мягкий" перевод строки). Эта инструкция будет выполнена только в том случае, если браузер не сможет вывести вашу фразу одной строкой в пределах окна просмотра.

Пример:

<nobr> Данная строка является самой длинной строкой документа,<wbr> которая не допускает какого-либо разбиения где бы то ни было </nobr>



В данном случае перевод строки будет произведен только после запятой.

Структурирование текста

Для удобства читателей текст рекомендуется разбить на логические части, каждая из которых посвящена отдельной теме.

Заголовки.

Элемент "заголовок" является контейнером и поэтому должен иметь открывающий (<h1>) и закрывающий (</h1>) тэги. HTML располагает шестью уровнями заголовков: h1 (самый верхний), h2, h3, h4, h5 и h6 (самый нижний). Программа просмотра выводит каждый из них, но вы не можете точно знать, в каком именно виде. Это обстоятельство является частью философии HTML: вы, как автор, отвечаете за содержание, а читатель определяет окончательный облик документа. Предназначение заголовков – показать структуру вашего документа. Точно так же, как и в теге "абзац", в заголовках можно использовать атрибут align.

Горизонтальные линии.

Элемент <hr> позволяет провести рельефную горизонтальную линию в окне большинства программ просмотра. Этот тэг не является контейнером, поэтому не требует закрывающего тэга. До и после линии автоматически вставляется пустая строка.

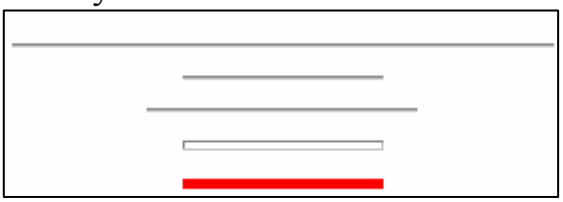
Таблица 4.

Атрибут	Назначение
align	Выравнивает по краю или центру; имеет значения left, center, right.
width	Устанавливает длину линии в пикселах или процентах от ширины окна браузера; в последнем случае добавляется символ %
size	Устанавливает ширину линии в пикселах.
noshade	Отменяет рельефность линии.
color	Указывает цвет линии. Используется формат RGB или стандартное имя.

Пример:

```
<body>
<hr align=center>
< hr align=center width=100>
< hr align=center width=50%>
< hr align=center width =100 size=5>
```

< hr align=center width =100 size=5 color=red>
</body>



Использование предварительно отформатированного текста.

Наиболее употребляемым является контейнер <pre>. Текст внутри него может записываться в любой форме. Поддерживаются также тэги <p> и
. Универсальность этого контейнера позволяет создавать таблицы и ровные колонки текста.

Текст внутри контейнера <pre> может содержать любые элементы физического и логического форматирования. Однако запрещено использование тэга <address> и тэгов заголовка.

Самым большим недостатком контейнера <pre> является возможность вывода преформатированного текста только моноширинным шрифтом.

Пример:

```
<pre>
  HR ALIGN=CENTER
    HR ALIGN=CENTER WIDTH=100
      HR ALIGN=CENTER WIDTH=50%
    HR ALIGN=CENTER WIDTH=100 SIZE=5
  HR ALIGN=CENTER WIDTH=100 SIZE=5 COLOR=RED
</pre>
```

```
HR ALIGN=CENTER
  HR ALIGN=CENTER WIDTH=100
    HR ALIGN=CENTER WIDTH=50%
  HR ALIGN=CENTER WIDTH=100 SIZE=5
HR ALIGN=CENTER WIDTH=100 SIZE=5 COLOR=RED
```

Цитата <blockquote>.

Данный тэг предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный тэгом <blockquote>, отступает от левого края документа на 8 пробелов.

<body>
На открытии данной конференции глава представительства произнес:
<blockquote>
Сегодня один из величайших дней для нашей компании.
</ blockquote>
</body>

При отображении браузером текст будет выглядеть так:

На открытии данной конференции глава представительства произнес:

Сегодня один из величайших дней для нашей компании.

1.3. Форматирование текста

Логическое форматирование

Элементы логического форматирования.
Таблица 5.

<cite>	Используется для выделения цитат или названий книг и статей, при этом текст обычно выводится курсивом. <cite>Tom Sawyer</cite> remains one of the classics of American literature.
<code>	Применяется для вывода небольшого куска программного кода (для больших листингов используется тэг <pre>) шрифтом фиксированной ширины.
	Этот элемент обычно используется для выделения важных фрагментов текста. Браузеры обычно отображают такой текст курсивом. The actual line reads, "Alas, poor Yorick. I knew him, Horatio."
<kbd>	Элемент, выделяющий шрифтом фиксированной ширины текст, вводимый пользователем с клавиатуры. To run the decoder, type <kbd>Restore</kbd> followed by your password.
<samp>	Используется для выделения нескольких символов шрифтом фиксированной ширины. The letters <samp>AEIOU</samp> are the vowels of the English language.
	Этот элемент обычно используется для выделения важных фрагментов текста. Браузеры обычно отображают такой текст полужирным шрифтом. The most important rule to remember is Don't panic!
<var>	Используется для отметки имен переменных. Обычно такой текст отображается курсивом. The sort routine rotates on the <var>I</var>th element.

Все перечисленные элементы являются контейнерами и требуют закрывающего тэга.

Физическое форматирование

Последней "инстанцией" определения внешнего вида документа является программа просмотра читателя. Вы имеете ограниченные возможности повлиять на этот процесс с помощью элементов физического форматирования, список которых приведен ниже. Физическое форматирование считается “не рекомендованным“ оформления Web-страниц.

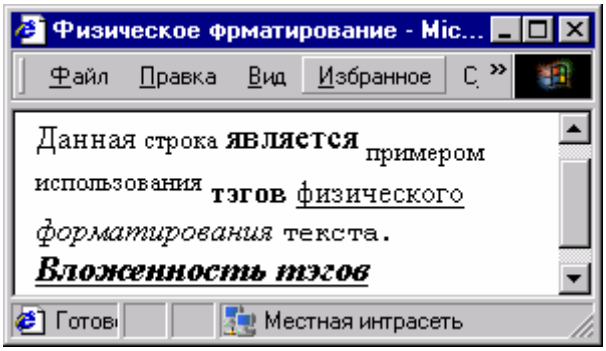
В последних версиях языка вместо него используются каскадыне таблицы стилей. Но тем не менее, тэги физического форматирования можно встретить еще во многих документах.

	Выделяет текст полужирным шрифтом.
	This is in bold text.
	Выделяет текст курсивом.
	This is initalic text.
<tt>	Выводит текст шрифтом фиксированной ширины.
	This is in <tt>teletype</tt> text.
<u>	Элемент подчёркивания.
	This text is <u>underlined</u>.
<strike>	Элемент зачеркивания. Отображается текст, перечеркнутый горизонтальной линией.
	This is a <strike>strikethough</strike> example.
<big>	Выводит текст шрифтом большего размера.
	This is <big>big</big> text.
<small>	Выводит текст шрифтом меньшего размера.
	This is small text.
<sub>	Сдвигает текст ниже уровня строки и выводит его (если возможно) шрифтом меньшего размера.
	This is a _{subscript}.
<sup>	Сдвигает текст выше уровня строки и выводит его (если возможно) шрифтом меньшего размера.
	This is a ^{superscript}.

Все элементы физического форматирования являются контейнерами, т. е. требуют закрывающего тэга. Элементы физического форматирования могут быть вложенными друг в друга.

```
<body>
  Данная
  <small>строка</small>
  <b><big>является</big></b>
  <sub> примером</sub>
```

```
  <sup>использования</sup>
  <b>тэгов</b>
  <u> физического</u>
  <i> форматирования </i>
  <tt>текста.</tt>
 <b> <i>  <u>
<big>Вложенность тэгов</big>
 </u> </i> </u>
</body>
```



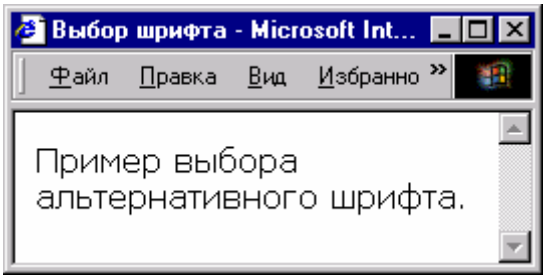
Шрифты.

Элемент font представляет собой контейнер, т. е. требует как открывающего, так и закрывающего тэгов. После стартового тэга обязательно указание атрибутов, без которых элемент не оказывает никакого влияния на текст, помещенный в контейнер. Элемент font может использоваться внутри любого другого текстового контейнера.

Атрибут face данного элемента позволяет вам указать тип шрифта, которым программа просмотра выведет ваш текст. Параметром атрибута служит название шрифта, которое должно в точности совпадать с названием шрифта, имеющегося у пользователя. Если нужного шрифта нет, программа проигнорирует запрос и будет использовать шрифт, установленный по умолчанию..

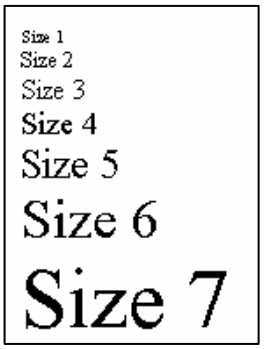
Атрибут face позволяет указать как один, так и несколько шрифтов (через запятую). Весь список будет просмотрен слева направо, и первый из имеющихся на машине пользователя будет использован для вывода документа.

```
<html>
<head>
<title>Выбор шрифта</title>
</head>
<body>
<font face="Verdana", "Arial", "Helvetica">
Пример выбора альтернативного шрифта. </font>
</body>
</html>
```



Атрибут size служит для указания размера шрифта в условных единицах от 1 до 7. Принято считать, что размер "нормального" шрифта соответствует числу 3. Размер может быть указан как абсолютной величиной (size=5), так и относительной (size=+2).

```
<body>
<font size=1>Size 1</font><br>
<font size=-1>Size 2</font><br>
<font size=3>Size 3</font><br>
<font size=4>Size 4</font><br>
<font size=+2>Size 5</font><br>
<font size=6>Size 6</font><br>
<font size=+4>Size 7</font><br>
</body>
```



Атрибут color устанавливает цвет шрифта, который может быть указан как в формате RGB, так и стандартным именем.

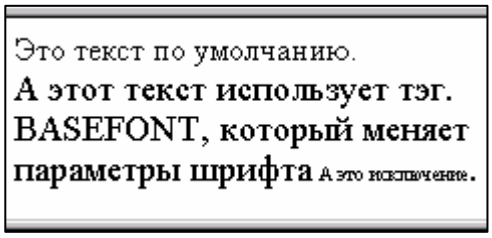
```
<body>
<font color="#FF0000">Этот текст будет выведен красным цветом</FONT><BR>
<font color="green"> Этот текст будет выведен зеленым цветом </FONT><BR>
</body>
```

Тэг <basefont> служит для указания размера, типа и цвета шрифта, стандартных для данного документа. Эти величины обязательны для всего документа, если только не переназначаются при помощи элемента font. После закрытия элемента font значения тэга <basefont> восстанавливаются. Значения атрибутов basefont могут быть изменены другим тэгом

<basefont> в любом месте документа. Это не контейнер и закрывающего тэга не существует.

Тэг <basefont> использует те же самые атрибуты, что и элемент font.

```
<body>
Это текст по умолчанию.<br>
<basefont size=4 face="georgia">
А этот текст использует тэг.<br>
BASEFONT, который меняет параметры шрифта
<font size=-3> А это исключение</font>.<br>
</body>
```



1.4. Списки

HTML имеет специальные элементы-контейнеры для представления данных в виде списков. Основными типами списков являются нумерованные и маркированные списки, списки определений. Для получения дополнительных эффектов различные типы списков могут вкладываться друг в друга.

Упорядоченный (нумерованный) список.

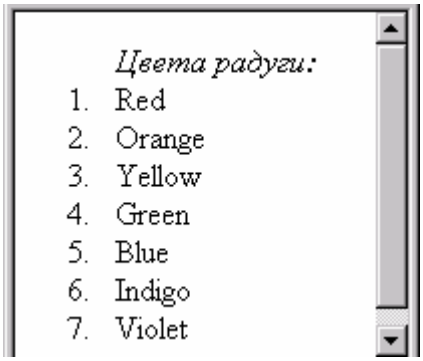
В HTML список состоит из тэга-контейнера списка, определяющего его тип, и стандартных тэгов <li>, предваряющих каждый пункт списка. Упорядоченный список используется для нумерованного перечисления отдельных пунктов или указания последовательности каких-то действий. Когда программа просмотра встречает тэг упорядоченного списка, она последовательно нумерует пункты списка: 1, 2, 3 и т.д.

Упорядоченный список открывается тэгом <ol>, а каждый его пункт начинается стандартным тэгом <li>. Для создания заголовка списка используется специальный тэг <lh>. Список закрывается тэгом . Открывающий и закрывающий тэги обеспечивают перевод строки до и после списка, отделяя таким образом список от остального текста.

Тэги абзаца можно использовать для отделения пунктов списка друг от друга. Внутри списка можно применять тэги стилей (как физические, так и логические, например, или <i>), и другие элементы HTML.

```
<body>
<ol>
 <lh><em>Цвета радуги:</em><br>
 <li>Red
 <li>Orange
 <li>Yellow
```

```
  <li>Green
  <li>Blue
  <li>Indigo
  <li>Violet
</ol>
</body>
```



Упорядоченные списки допускают вложения друг в друга

Атрибуты тэга <ol> позволяют устанавливать вид маркеров элементов списка, а также задавать начальный маркер списка. В Таблице 6 приведены атрибуты этого тэга и их назначение.

Таблица 6.

Атрибут	**Назначение**
compact	Представляет список в более компактном виде.
type=A	Устанавливает маркер в виде прописных букв.
type=a	Устанавливает маркер в виде строчных букв.
type=I	Устанавливает маркер в виде больших римских цифр.
type=i	Устанавливает маркер в виде маленьких римских цифр.
type=1	Устанавливает маркер в виде арабских цифр.
start=n	Устанавливает начальный маркер в текущем списке.

Неупорядоченный (маркированный) список.

В HTML существует возможность создания неупорядоченных списков, т. е. таких, в которых отношения между пунктами не определены. (Списки такого типа называют также ненумерованными или маркированными.)

Неупорядоченный список вместо буквенной или цифровой нумерации предполагает использование различных символов, называемых маркерами списка (bullets). Как и в упорядоченных списках, здесь также обеспечивается перевод строки до и после списка, а также допускается вложенность списков. Список располагается внутри контейнера <ul>. Программы просмотра создают автоматический отступ для вложенных

списков и сами разнообразят маркеры, вид которых зависит от типа программы.

Как и в случае тэга <ol>, для тэга <ul> в HTML можно устанавливать вид маркеров в неупорядоченных списках при помощи атрибута type, который допускает три значения: disc, square и circle

Как и тэг <ol>, тэг <ul> имеет атрибут compact, позволяющий выводить неупорядоченный список в более компактном виде.

Список определений.

Списки определений, также называемые словарями специальных терминов (глоссариями), являются особым видом списков HTML. Они представляют текст в форме словарной статьи, состоящей из определяемого термина и абзаца, раскрывающего его значение. Элемент списка определений dl является контейнером и обеспечивает отделение списка от остального текста пустыми строками. Внутри контейнера тэгом <dt> помечается определяемый термин, а тэгом <dd> – абзац с его определением. Тэги <dt> и <dd> не являются контейнерами и поэтому не имеют закрывающих тэгов. Базовый шаблон списка определений выглядит следующим образом:

```
<dl>
<dt>Термин
<dd>Определение данного термина
</dl>
```

Списки определений могут включать другие элементы HTML. Часто применяются элементы стилей (физические и логические).

Комбинирование списков

Бывают ситуации, когда в список одного типа требуется включить список (или списки) другого типа. HTML позволяет производить любое комбинирование типов списков.

```
<body>
<ol>
  <lh><em>Planets of the Solar System:</em><br>
  <li>Mercury
  <ul> <ul>
<li>Roman god of commerce, travel, and thievery
<li>Dictionary Definition
<dl>
<dt>Mercury
<dd>The smallest of the planets and the one
nearest the sun, having a sidereal period of
revolution about the sun of 88.0 days at a
mean distance of 58.3 million kilometers (36.2
```

```
million miles) and a mean radius of appropriately
2,414 kilometers (1,500 miles).
</dl>
  </ul> </ul>
  <li>Venus
  <ul> <ul>
<li>Roman goddess of sexual love and physical beauty
<li>Dictionary Definition
<dl>
  <dt>Venus
  <dd>The second planet from the sun, having an
average radius        of 6,052 kilometers (3,760 miles),
a mass 0.815 times that of Earth, anda sidereal
period of revolution about the sun of 224.7 days
at a mean distance of approximately 100.1 million
kilometers (67.2 million miles).
                  </dl>
  </ul> </ul>
</ol>
</body>
```

Planets of the Solar System:

1. Mercury
   - Roman god of commerce, travel, and thievery
   - Dictionary Definition

   Mercury

   The smallest of the planets and the one nearest the sun, having a sidereal period of revolution about the sun of 88.0 days at a mean distance of 58.3 million kilometers (36.2 million miles) and a mean radius of appropriately 2,414 kilometers (1,500 miles).
2. Venus
   - Roman goddess of sexual love and physical beauty
   - Dictionary Definition

   Venus

   The second planet from the sun, having an average radius of 6,052 kilometers (3,760 miles), a mass 0.815 times that of Earth, anda sidereal period of revolution about the sun of 224.7 days at a mean distance of approximately 100.1 million kilometers (67.2 million miles).

Отступ для каждого списка устанавливает программа просмотра, однако, если это необходимо, для его увеличения можно добавить "пустой" список:

```
<ol>
<li>Простой список
<li> Простой список
</ol>
```

нужно написать

```
<ol>
<ol>
<li>Список с увеличенным отступом
<li> Список с увеличенным отступом
</ol></ol>
```

1. Простой список
2. Простой список

    1. Список с увеличенным отступом
    2. Список с увеличенным отступом

Дополнительные возможности форматирования списков

Вы можете создать собственные маркеры для использования в ненумерованных списках.

Контейнер ul информирует браузер о необходимости интерпретировать заключенный в нем текст как неупорядоченный список. Если не нужны стандартные маркеры, то тэги <li> не используется. Вместо него нужно записать код, определяющий ваш новый маркер, например:

<img src= "cube.gif" align=top>Red

Тэг <img> указывает на графический файл используемого маркера и метод выравнивания изображения.

1.5. Гиперссылки

Значение ссылок во Всемирной паутине трудно переоценить. Читая книгу, вы всегда имеете ее под рукой. Работая в WWW, вы понятия не имеете, где находится та или иная нужная вам страница. Поэтому ссылки здесь являются единственной возможностью перейти от одного документа к другому.

Гипертекст и гипермедиа

Гипертекстовый документ – это документ, содержащий ссылки на другие документы, позволяющие при помощи нажатия кнопки мыши быстро перемещаться от одного документа к другому.

Гипермедийнь документ основан на гипертекстовом документе, но в дополнение к тексту содержит разнообразную графику, видео и аудиообъекты. В таких документах в качестве ссылок часто используются изображения.

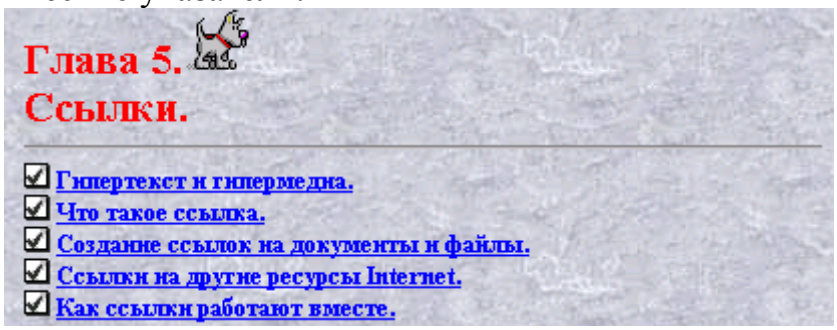
Ссылка состоит из двух частей. Первая из них – это то, что вы видите на Web-странице; она называется указатель (anchor). Вторая часть, дающая инструкцию браузеру, называется адресной частью ссылки (URL-адресом). Когда вы щелкаете мышью по указателю ссылки, браузер загружает документ, адрес которого определяется URL-адресом.

Указатели.

Указатели бывают двух типов – текстовые и графические.

Текстовые указатели представляют собой слово или слова, подчеркнутые прямой линией.

На рисунке приведен пример Web-страницы, содержащей текстовые и графические указатели.



Графические указатели в принципе мало отличаются от текстовых. Они не подчеркиваются и не выделяются цветом, но вокруг них можно сделать рамку.

Маркеры списка (Bullets).

Графические указатели часто имеют вид точки, звездочки или другого маркера.

Как правило, строка рядом с маркером также является указателем, имеющим тот же самый URL-адрес.

Второй частью ссылки является URL-адрес. Это не что иное, как адрес Web-страницы, которая будет загружена, если щелкнуть на указателе ссылки кнопкой мыши. Указание адреса может быть относительным или абсолютным.

Относительные указатели.

URL-адрес файла, расположенный на том же компьютере, что и документ, в котором находится указатель этой ссылки, называется относительным. Это означает, что если браузер загрузил страницу, находящуюся по адресу http://www.mysite.com/page, то относительный адрес /picture подразумевает адрес http://www.mysite.com/page/picture, т. е. подкаталог, расположенный на той же машине.

Абсолютные указатели.

URL-адрес, полностью определяющий компьютер, каталог и файл, называется абсолютным. В отличие от относительных адресов, абсолютные адреса могут ссылаться на файлы, расположенные на других компьютерах.

Создание ссылок на документы и файлы

Для создания ссылки необходимо сообщить браузеру, какой элемент страницы является указателем, и указать адрес документа, на который ссылаетесь. Оба действия выполняются при помощи тэга <a>.

Тэг <a> имеет атрибут href, в котором размещается URL– адрес.

anchor

Для облегчения работы с относительными указателями в HTML введен тэг <base>. Он располагается в начале документа и содержит URL-адрес, относительно которого в документе построена вся адресация. Это указание влияет на любой нижестоящий тэг, включая <a>, <img> и т.д. Если, например, вы вставите строку

<base href="http://www.server.com">

вся относительная адресация в дальнейшем будет базироваться на этом адресе. Так, относительный указатель на файл "images/face.gif будет подразумевать адрес http://www.server.com/images/face.gif. Если тэг <base> отсутствует, относительная адресация строится на URL-адресе текущего документа.

Некоторые программы просмотра выводят в маленьком окошке текст, содержащийся в атрибуте title тэга <a>, если задержать курсор мыши на указателе ссылки.

 Me Page

Создание графического указателя аналогично созданию текстового указателя. Вместо текста в тэге <a> нужно разместить имя файла изображения. В следующей строке кода атрибут href опять указывает адрес домашней страницы, только вместо текстового анкера при помощи тэга <img> создается графический указатель, представляющий собой маленькую каринку

<img src="comp.jpg">

Кроме ссылок на другие документы, часто бывает полезно включить ссылки на разные части текущего документа.

Для построения внутренней ссылки сначала нужно создать указатель, показывающий место назначения. Для этого нужно разместить там указатель и дать ему имя при помощи атрибута name тэга <a>. При этом атрибут href не используется, и браузер не выделяет содержимое тэга <a>.

Middle Section in Web page

После того как указатель получил имя, можно приступить к созданию ссылки на него. Для этого, вместо указания в атрибуте href адреса документа, как это делалось ранее, поместим туда имя указателя со специальным префиксом (#), говорящим о том, что это внутренняя ссылка.

Jump to the middle

Теперь, если пользователь щёлкнет кнопкой мыши на словах Jump to the middle, программа просмотра выведет среднюю часть документа, причем указатель ссылки будет расположен в верхней строке окна.

Файлы и другие встраиваемые объекты

Когда пользователь щелкает мышью на ссылке, указывающей на другую Web-страницу, она выводится непосредственно в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать потом.

Связь с электронной почтой (e-mail).

Если вас интересует отклик читателей на содержание вашего документа, вы захотите поместить на странице свой адрес e-mail.

Send me E-mail.

После щелчка мышью на ссылке на ваш адрес браузер откроет собственное окно для работы с электронной почтой.

Связь с FTP.

Ссылка на сайт FTP похожа на другие гипертекстовые ссылки. Вместо http: Нужно поставить ftp:, а вместо URL-адреса – //sitename/path. Вы должны удостовериться, что сайт, на который вы ссылаетесь, разрешает анонимный доступ. Практически все браузеры работают с FTP без всяких проблем. Ваша ссылка может иметь следующий вид:

You can get the FAQ <a href="ftp://ftp.mysite.com/pub/FAQ">here< /a>.

Если не указывать имя файла, браузер выведет перечень всех файлов в каталоге.

Таблица 7.

Ссылка на	Формат	Пример
Web-страницу	http://sitename/	http://www.mysite.com/
e-mail	mailto:address	mailto:me@mysite.com
Newsgroup	news:newsgroupname	news:news.newusers.questions
FTP	ftp://sitename/	ftp://ftp.mysite.com/
Telnet	telnet://sitename/	telnet://bbs.mysite.com/

1.6. Таблицы

Для лучшего представления информации вы можете использовать таблицы. Элемент table представляет собой тэг-контейнер, в котором размещается содержимое таблицы.

Таблица строится по строкам: для обозначения строки используется контейнер tr, для обозначения заголовков столбцов (строк) – контейнер th, а для данных в ячейках – контейнер td. Заголовки выделяются полужирным шрифтом и центрируются в своих ячейках. Данные имеют обычный шрифт и выравниваются по левой стороне ячейки.

Таблица 8.

Тэг	Описание
<table></table>	Контейнер таблицы.
<tr></tr>	Контейнер строки таблицы. Допускается отсутствие закрывающего тэга.
<td></td>	Контейнер ячейки таблицы. В ячейку можно включить другую таблицу. Закрывающий тэг может быть опущен.
<th></th>	Контейнер заголовка, располагающегося обычно в первой строке, либо в первом столбце таблицы. Закрывающий тэг также необязателен.

Пример:

```
<table border=2>
<tr>
<th>Colors</th><th>Of</th><th>The Rainbow</th>
</tr>
<tr>
<td>Red</td><td>Orange</td><td>Yellow</td>
</tr>
<tr>
<td>Green</td><td>Blue</td><td>Violet</td>
</tr>
</table>
<hr>
<table border=2>
<caption>My Favorite Groups</caption>
<tr><th>Rock</th>
<td>Pink Floyd</td>
 <td>Led Zepplin</td>
<td>The Dobbie Brothers</td></tr>
<tr><th>Soft</th><td>Simon and Garfunkel</td>
<td>Peter, paul, & Mary</td>
<td>Neil Young</td></tr>
<tr><th>New Age</th><td>Enya</td>
<td>Clannad</td>
<td>Steamroller</td></tr>
</table>
```

Colors	Of	The Rainbow
Red	Orange	Yellow
Green	Blue	Violet

My Favorite Groups

Rock	Pink Floyd	Led Zepplin	The Dobbie Brothers
Soft	Simon and Garfunkel	Peter, paul, & Mary	Neil Young
New Age	Enya	Clannad	Steamroller

Если надо разнообразить заголовки, то можно применить тэги физического форматирования.

Таблицы всегда должны быть прямоугольными. Другие формы не допускаются.

Размещение данных внутри ячеек

При помощи атрибутов align и valign можно по-разному размещать данные относительно границ ячейки. Эти атрибуты используются совместно с элементами <caption>, <tr>, <TH> и <td> в самых различных комбинациях. Ниже приведены значения атрибутов для перечисленных элементов.

Таблица 9.

<caption>	Атрибут align может иметь значения top и bottom (по умолчанию – top); размещает заголовок таблицы сверху или снизу.
<tr>	Атрибут align может принимать значения left, center и right (по умолчанию – left для данных и center для заголовков); он определяет горизонтальное выравнивание данных в ячейках и действует на всю строку, если не отменяется тем же атрибутом в отдельной ячейке. Атрибут valing может иметь значения top, bottom, middle и baseline (по умолчанию – middle); он регулирует положение данных относительно верхней и нижней границ ячейки и влияет на всю строку, если не отменяется таким же атрибутом в отдельной ячейке, baseline применяется ко всем элементам строки и выравнивает их по базовой линии.
<th>	Атрибут align может принимать значения left, center и right (по умолчанию – center). Атрибут valing может иметь значения top, bottom и middle (по умолчанию – middle).
<td>	Атрибут align может принимать значения left, center и right (по умолчанию – left). Атрибуту valing может иметь значения top, bottom и middle (по умолчанию – middle).

Применение этих атрибутов:

```
< table border>
  <caption align=bottom>A Really Ugly Table</caption>
  <tr>
<th></th><th>##########</th><th>##########</th>
<th>##########</th>
  </tr>
  <tr align=right>
<th>Row 1</th><td>XX<br>XX</td><td align=center>X
</td><td>XXX</td>
  </tr>
  <tr valing=baseline>
<th aling=left>Second Row</th><td>XXX<br>XXX</td><td>XXX</td>
<td>XXX<br>XXXXX<br>XXX</td>
  </tr>
  <tr aling=left>
<th>This Is<br>The Bottom Row of <br>The Table</th>
<td valing=bottom>XXXXX</td>
<td valing=top>XXX<br>XXXXX</td>
<td valing=middle>XXXXX</td>
  </tr>
</table>
```

	##########	##########	##########
Row 1	XX XX	X	XXX
Second Row	XXX XXX	XXX	XXX XXXXX XXX
This Is The Bottom Row of The Table	XXXXX	XXX XXXXX	XXXXX

A Really Ugly Table

При отсутствии атрибута border рамка не прорисовывается. Такие таблицы можно использовать для табличной верстки старниц.

Объединение ячеек

Смежные ячейки таблицы могут объединяться с целью размещения большего количества данных. Например, в таблице из пяти строк и пяти столбцов все ячейки первой строки можно объединить и поместить в этой строке красивый заголовок таблицы. Возможно также объединение нескольких строк или создание пустой прямоугольной области.

Для соединения двух смежных ячеек в одном столбце нужно использовать атрибут rowspan тэга <th> или <td>, например:

<td rowspan=2>

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут colspan тех же тэгов, например:

<td colspan=2>

Пустые ячейки

Существует разница между пустой ячейкой и ячейкой с невидимым содержанием

Таблица 10.

width	Определяет ширину всей таблицы в пикселах, либо в процентах от ширины окна браузера. Может также использоваться для отдельной ячейки.
height	Определяет высоту всей таблицы в пикселах, либо в процентах от высоты окна браузера. Может также использоваться дня отдельной ячейки.
border	Устанавливает ширину рамки таблицы в пикселях, например, border=2.
cellpading cellspasing	Добавляют свободное пространство между данными внутри ячейки и ее границами и, соответственно, между ячейками внутри всей таблицы. Если рамка отсутствует, то результат их действия одинаков.

Использование цветов

Вы можете изменить цвет фона ячейки при помощи атрибута bgcolor перед размещением в ней текста или изображения, а также использовать атрибут bordercolor для изменения цвета рамки ячейки.

Тэги <table>, <td>, <th> и <tr> допускают использование в них указанных атрибутов.

Теги <table> и <tr> поддерживают аттрибут background.

Альтернативные способы представления табличных данных

- Список
- Использоваие изображения.
- Предварительно отформатированный текст.

<pre>

```
+------------------+---------------------+--------------------+
| Offense   | Defense   | Goalie  |
+------------------+---------------------+--------------------+
```

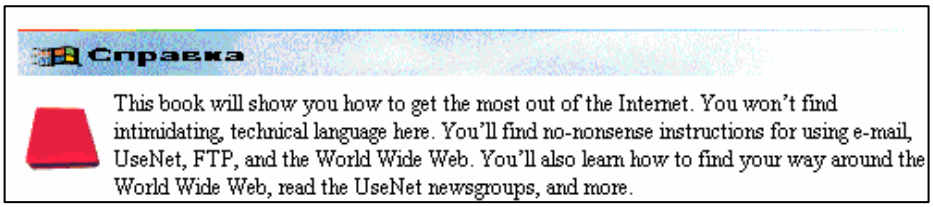
Husmann	O’Donnell	
Popplewell	
McGilly	Longo	Weinberg
Donahue	Seymour	
Camillo	Walsh	
+------------------+----------------------+--------------------+
</pre>

Offense	Defense	Goalie
Husmann	O’Donnell	
Popplewell		
McGilly	Longo	Weinberg
Donahue	Seymour	
Camillo	Walsh	

Использование изображения в качестве заголовка таблицы

Вы можете украсить свою таблицу, поместив в ее заголовок изображение вместо текста.

```
<table width=500 cellspacing=0 cellpadding=2 border=0>
<tr>
<th colspan=2>
 <img src="cloud.gif" border=0 height=25 width=500>
</th>
</tr>
<tr>
<td valing=top>
  <img src="an59.gif">
</td>
<td valing=top>
  This book will show you how to get the most out of the
  Internet. You won’t find intimidating, technical language
  here. You’ll find no-nonsense instructions for using e-mail,
  UseNet, FTP, and the World Wide Web. You’ll also learn how
  to find your way around the World Wide Web, read the UseNet
  newsgroups, and more.
</td>
</tr>
</table>
```



1.7. Использование графики

Изображения могут сделать текст вашего документа более содержательным.

Изображения помогают лучше передать суть и содержание документа.

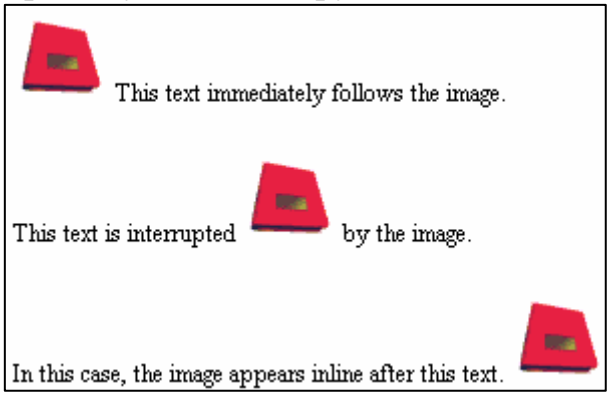
Вставка изображения в документ

Для вставки изображения на страницу следует воспользоваться тэгом <img> совместно с атрибутом src, поместив их в надлежащее место вашего HTML-документа:

<img src="URL">

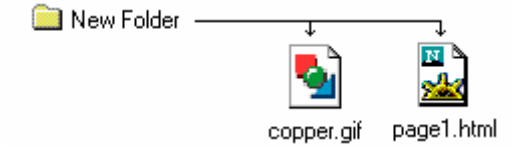
По умолчанию браузер выводит изображение немедленно после текста или другого объекта, описанного предыдущими инструкциями

```
<body>
<p>
<img src="book.gif">
This text immediately follows the image.
<p>
This text is interrupted
<img src="book.gif">
by the image.
<p>
In this case, the image appears inline after this text.
<img src="book.gif">
</body>
```

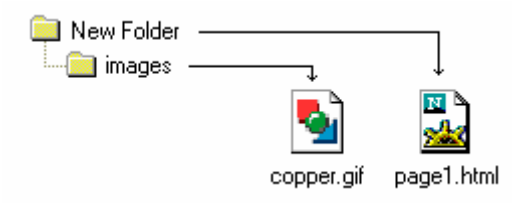


Предусмотрите хранение файлов графики в одном определенном каталоге (не корневом) вашего Web-сайта. Тогда вы сможете использовать относительную адресацию в комбинации с тэгом <base>, а не указывать полный URL-адрес.

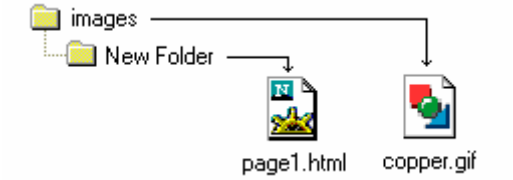
Атрибут src определяет не только какое изображение, но и где хранится это изображение. В примере, который приведен ниже. src="copper.gif" означает, что браузер будет искать изображение с именем copper.gif в той же самой папке (или каталоге), где непосредственно расположен html документ.



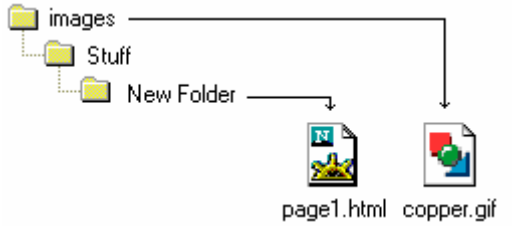
- src="images/copper.gif" означает, что изображение из папки нижнего уровня, по сравнению с папкой HTML-документа, который запросил его. Это можно продолжить по уровням вниз настолько это необходимо.



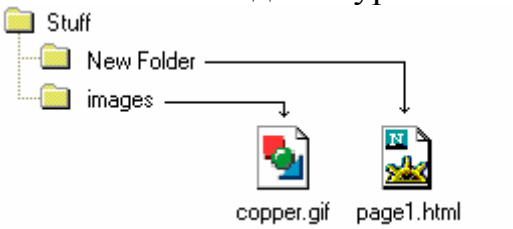
- src="../copper.gif" означает, что изображение находится в папке верхнего уровня, по сравнению с папкой HTML-документа, который запросил его.



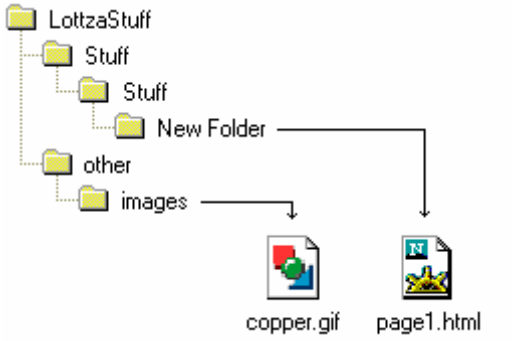
- src="../../copper.gif" означает, что изображение в папке на два уровня выше, чем папка HTML-документа, который запросил его.



- src="../images/copper.gif" означает, что изображение и HTML-документ в папках одного уровня.



- src="../../../other/images/copper.gif" - …



Выравнивание текста по краю изображения

По умолчанию, когда изображение вставляется в строку текста, строка выравнивается по низу изображения. Изменить эту установку можно при помощи атрибута align тэга <img>.

Таблица11.

Значение	Описание.
top	Выравнивает текст по верху изображения.
middle	Выравнивает текст по середине изображения.
bottom	Выравнивает текст по низу изображения.

```
<body>
  <p>
<img src="comp.jpg" align=top>
This text is aligned with the top of the image.
  </p>
  <p>
<img src="comp.jpg" align=middle>
This text is aligned with the middle of the image.
  </p>
  <p>
<img src="comp.jpg" align=bottom>
This text is aligned with the bottom of the image.
  </p>
</body>
```

Позиционирование изображения на странице

По умолчанию программа просмотра выводит изображение в текущей строке. Текст не "обтекает" его. Однако при помощи атрибута align тэга <img> изображение можно сделать "плавающим", т. е. заставить текст расположиться вокруг изображения.

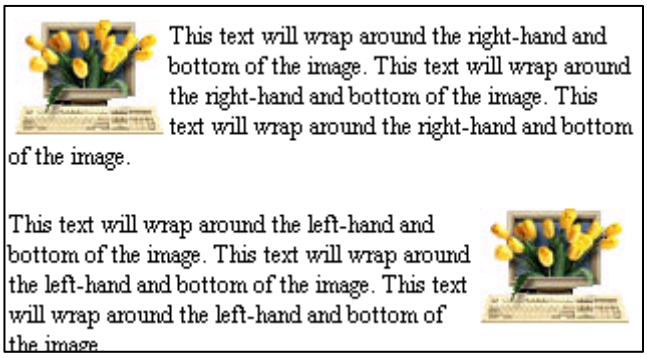
Таблица 12.

Значение	**Описание.**
left	Обтекаемое текстом изображение прижато к левой стороне окна браузера.
right	Обтекаемое текстом изображение прижато к правой стороне окна браузера.

```
  <p>
<img src="comp.jpg" align=left>
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
<img src="comp.jpg" align=right>
This text will wrap around the left-hand and bottom of the image.
This text will wrap around the left-hand and bottom of the image.
This text will wrap around the left-hand and bottom of the image.
  </p>
```

При помощи тэга <img> программе просмотра можно сообщить размеры изображения, которое затем размещено на странице:

<img src="comp.jpg">
<img src="comp.jpg" width=140 height=100>



Для указания размеров изображения (в пикселях) служат атрибуты height и width тэга <img>. Если указанные размеры не совпадают с размерами изображения, программа просмотра изменяет масштаб изображения.

Альтернативное описание изображения:

<img src="filename" alt="Description">

Если браузер читателя не выводит изображение, на его месте будет помещено альтернативное описание. Если изображение выводится, это описание будет находиться на месте иллюстрации до начала ее загрузки. Еще лучше использовать эту возможность совместно с указанием размеров.

Если указатель мыши задержать на иллюстрации на одну-две секунды, этот же текст будет выведен в специальном всплывающем окошке в виде подсказки.

Помещение изображения в рамку.

Эта простая операция предполагает применение атрибута border тэга <img>. По умолчанию программа просмотра использует рамку, которая указана в соответствующей ссылке. Введите ширину рамки в пикселах, как показано в примере:

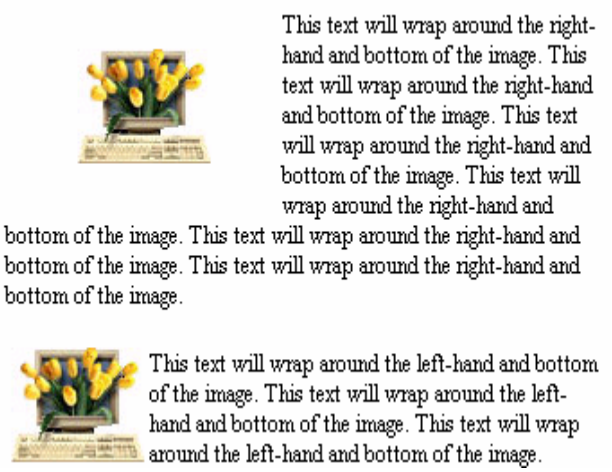
<body>
<img src="comp.jpg" border=0>
<img src="comp.jpg" border=5>
<img src="comp.jpg" border=10>
</body>

Отделение изображения от текста.

Вам может не понравиться, что текст слишком близко подходит к изображению Если это так, используйте атрибуты vspace и hspace для

указания расстояния (по вертикали и горизонтали) между кромкой текста и краями иллюстрации.

```
<body>
  <p>
<img src="comp.jpg" aling=left hspace=40 vspace=20>
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
This text will wrap around the right-hand and bottom of the image.
  </p>
  <p>
<img src="comp.jpg" align=left>
This text will wrap around the left-hand and bottom of the image.
This text will wrap around the left-hand and bottom of the image.
This text will wrap around the left-hand and bottom of the image.
  </p>
</body>
```



Карта ссылок

Сегодня многие Web-страницы располагают интересной разновидностью меню – изображениями-картами, т.е. изображениями, чувствительными к нажатию кнопки мыши (imagemaps – от англ. слов image – изображение и map – карта, план.).

Разные части изображения на странице могут быть связаны с разными URL-адресами. Так как пользователь должен знать, где расположены "чувствительные" области изображения, они часто выделяются рамками, тоже являющимися частью изображения.

Изображения-карты бывают двух типов: обслуживаемые сервером и программой-клиентом (браузером).

Синтаксис определения изображения-карты, обслуживаемой клиентом следующий:

<map name="mapname" >

<area [shape="shape"] coords="x,y,..." [href'="URL"] [nohref]>

</map>

Определение начинается тэгом <map> и заканчивается тэгом </map>. Для того чтобы ссылаться на это определение позже в тэге <img>, оно должно иметь имя, задаваемое при помощи атрибута name.

Для задания чувствительных областей используется тэг <area>.

Атрибуты тэга <area>.

Таблица 13.

Атрибут	Описание
shape	Определяет форму чувствительной зоны. Имеет значения rect, poly, circle, rect.
coords	В этом атрибуте перечисляются через запятую пары координат – x1,y1,x2,y2. Между парами также ставится запятая.
href	Определяет URL-адрес ссылки. Относительные адреса задаются относительно документа, содержащего тэг <map>. Если в этом же документе используется тэг <base>, адресация рассчитывается относительно URL, указанного в этом тэге.
nohref	Указывает нечувствительную зону, т.е. зону, не связанную с другими документами или ресурсами. Атрибуты nohref и href взаимоисключающие.

Ссылка на изображение-карту:

<map name=mymap>

<area shape=rect coords="0,0,100,100" href=item1.html>

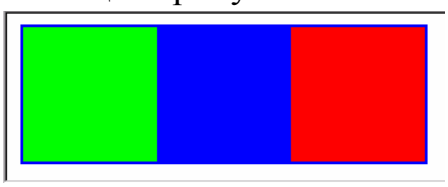
<area shape=rect coords="101,0,200,100" href=item2.html>

<area shape=rect coords="201,0,300,100" href=item3.html>

</map>

<img src=mymap.gif usemap=#mymap>

Этот тэг загружает изображение mymap.gif Атрибут usemap указывает на имя определения изображения-карты, которое было присвоено при помощи атрибута name тэга <map>.



1.8. Вставка объектов мультимедиа

Вставка видео

Для вставки видео-фрагментов в код HTML-страницы и его дальнейшего просмотра средствами браузера используется уже известный Вам тэг <img> с новым атрибутом dynsrc. Например, строка кода:

<img src=”image.gif” dynsrc=”video.avi”>

выведет в окне браузера сначала картинку image.gif и весь текст, а затем, пока пользователь будет читать текст, догрузит видеофайл и запустит его вместо картинки.

При помощи атрибута loop можно задать количество раз воспроизведения клипа. Значение этого атрибута равное –1 или символьной константе infinite, позволяет прокручивать данное изображение не ограниченное колличество раз. Данный атрибут принимает целочисленные значения.

Атрибут start позволяет указать событие после совершения которого начнется воспроизведение клипа. Например,

<img src=”image.gif” dynsrc=”video.avi” start=mouseover/fileopen>

Для просмотра видеоклипа также можно использовать универсальный тэг <embed>. Он позволяет воспроизводить и видео и аудио клипы в обоих браузерах. Используя такие атрибуты данного тэга, как height и width вы можете изменять высоту и ширину видеоклипа в пикселах. Используя ключевое слово autostart можно автоматически загружать клип на выполнение сразу после его загрузки.

Атрибут loop=yes задает бесконечное количество воспроизведений данного клипа.

Вставка аудио

Один из наиболее популярных способов вставки аудиоклипов – использование простой ссылки на звуковой файл:
<img src=”images/sound.gif” align=left>
 Мое приветствие

В данном случаи загрузка и последующее воспроизведение данного аудиофайла будет происходить только после щелчка мышкой по ссылке. На экран этот файл выведен не будет.

Также используется тег <bgsound>. При использовании этого тега в вашем HTML-документе при его загрузке сначала происходит загрузка его текстовой и графической части и лишь затем, загрузка и воспроизведение аудиофайла указанного в атрибутах тега <bgsound>.

Например**:**

<bgsound=”myaudio.wav” loop=10>

Значение атрибута loop равное infinite указывает на бесконечное количество повторений данного аудиофрагмента (пока пользователь не покинет страницу).

Для вставки звукого файла может быть использован тег <embed>. При выводе аудиоклипа броузер выводит на экран набор кнопок, позволяющий управлять воспроизведением клипа. Эти кнопки управления появятся на вашем экране только после того, как будет загружена вся HTML-страница, и скачан целиком аудио файл.

<embed src=”lion.wav” height=60 width=140 autostart=true loop=true>

Данная строка выводит звуковой файл и устанавливает размер панели управления клипа. Атрибут loop позволяет задать количество раз проигрывания клипа.

Иногда еще используется атрибут hidden=true (по умолчанию – false), который скрывает вывод на экран управляющик кнопок. В этом случаи просто звучит звук и посетители не могут им управлять.

1.9. Таблицы стилей

Таблицы стилей HTML предназначены для:

- Изменение расстояний между строками, словами и отдельными символами.
- Установка левого, правого, верхнего и нижнего полей элемента (блока текста контейнера HTML).
- Установка отступа элемента.
- Изменение размера, стиля и других атрибутов шрифта элемента.
- Установка рамки вокруг элемента.
- Включение фонового изображения и фонового цвета в элемент.

Связывание документа с таблицей стилей

Большим преимуществом таблиц стилей HTML является возможность отделить операцию форматирования от содержания документа. Сначала вы определяете, как должен выглядеть текст в том или ином месте страницы, а затем вводите сам текст. Если вы позднее решите, например, заменить цвет шрифта заголовков на синий, для этого будет достаточно поменять только стиль этих заголовков. Делать изменения в тексте нет необходимости.

Способы применения таблицы стилей.

Таблица 14.

Связывание (Linking).	Можно связать HTML-документ с таблицей стилей, хранящейся в отдельном файле.
Встраивание (Embedding).	Можно встроить таблицу стилей в HTML-документ с помощью контейнера <style>.
Задание стиля для отдельного фрагмента документа (Inline).	Можно определять элементы стиля "на лету", т. е. указывать их в тэгах HTML, например, в тэге абзаца <p>.

Создания таблицы стилей в виде отдельного файла для применения его ко всем страницам сайта.

Этот метод упрощает создание сайта. Вы можете даже разработать единую таблицу стилей, которую могли бы использовать все сайты вашей организации.

Хранить таблицу стилей следует в текстовом файле с расширением .css. Его можно создать при помощи любого текстового редактора. У вас не будет никаких трудностей. Для связывания таблицы стилей с документом HTML используйте тэг <link> следующим образом:

<link rel=stylesheet href=www.myserver.com/mysheet.css type="text/css">

Укажите в атрибуте href URL-адрес вашей таблицы стилей. Дайте атрибуту type значение "text/css", что позволит программам просмотра, не поддерживающим таблицы стилей, проигнорировать указанный адрес.

Встраивание таблицы стилей в документ.

Таблицу стилей необязательно хранить в виде отдельного файла. Ее можно встроить непосредственно в документ, однако в этом случае она будет действовать только внутри файла этого документа. Для распространения действия таблицы на другие документы ее необходимо скопировать в каждый из них.

Для включения таблицы стилей в документ воспользуйтесь контейнером <style>. Он размещается между тэгами <html> и <body>:

```
<head>
</head>
<style type="text/css">
Style definitions go here
</style>
<body>
</body>
```

Тэг <style> имеет единственный атрибут type, определяющий тип mime (Multipurpose Internet Mail Extension, стандарт электронной почты Internet). Определяйте его как "text/css" для того, чтобы браузеры, не поддерживающие таблицы стилей, могли игнорировать тэг <style>.

Ниже приведен конкретный пример тэга <style>.

<style type="text/css">
h1 {color: blue;}
</style>

Задание стиля для отдельного фрагмента документа:

Вы можете определять стиль, что называется, "на лету", оперативно внося требуемые изменения. Например, если вы определили стиль документа с заголовком одного цвета, а потом решили выделить цветом какой-то элемент заголовка, вы можете это сделать внутри тэга заголовка, не изменяя общий стиль документа.

Такой метод действует внутри тэга, где определен или переопределен стиль при помощи атрибута style. Он поддерживается всеми подчиненными тэгами тега <body>. Для оперативного определения стиля добавьте к нужному тэгу атрибут style и присвойте ему строковое значение, указывающее новый стиль:

<h1 style="color: blue">

Используя атрибут style с тэгом , можно определять стиль части документа, расположенной в контейнере . Это работает благодаря принципу "наследования. Например, если вы хотите установить цвет шрифта для целого блока тэгов синим, вы можете расположить эти тэги внутри контейнера и определить цвет шрифта текста следующим образом:

<h1>This is a heading</h1>
This is a paragraph. It will look blue in the user's browser

Для изменения стиля нескольких слов или даже символов можно использовать атрибут style совместно с тэгом , например:

This is a simple block of text

Синтаксис таблиц стилей

Таблицы стилей хранятся в текстовых файлах, удобных для редактирования. Их нетрудно создавать вручную, однако, как и для HTML-документов, существуют специальные редакторы таблиц стилей.

Таблицы стилей позволяют определять стиль для одного или нескольких тэгов. Например, вы можете создать таблицу стилей, определяющую стили для тэгов <h1>, <h2>, и . Каждое определение называется правилом (rule). Правило содержит селектор (тэг HTML), за которым следует декларация (определение стиля). Селектор является связующим звеном между определением и тэгом. Ниже приведен пример правила, указывающего стиль для каждого из тэгов заголовка <h1>:

hl {color: blue;}

Декларация заключается в фигурные скобки. Каждая декларация имеет две части: название свойства и присваиваемое ему значение, разделенные двоеточием. В HTML включены десятки свойств (font-size, font-style, color, margin-right и т. д.). Каждое свойство может принимать несколько значений, одно из которых приписывается ему по умолчанию.

В предыдущем примере было указано лишь одно свойство color. Однако ничто не мешает определить целый ряд свойств в одном тэге, отделив их друг от друга точкой с запятой:

Hl {color: blue; font-size: 12pt; text-line: center;}

В этом примере программа просмотра выведет каждый заголовок первого уровня синим шрифтом размером 12 пунктов и выровняет их по центру окна. Для всех прочих свойств будут использоваться значения по умолчанию (например, свойству font-style будет присвоено значение normal).

Группирование селекторов.

Если вы хотите определить один и тот же стиль для нескольких тэгов, вы можете перечислить их в отдельном списке:

p {font-size: 12pt;}
ul {font-size: 12pt;}
li {font-size: 12pt;}

HTML позволяет сделать то же самое и в более компактном виде – в одной строке:

p, ul, li {font-size: 12pt;}

Запятая здесь является обязательным элементом. Если она опущена, смысл правила изменится.

Комментирование таблицы стилей.

По мере усложнения таблицы стилей, скорее всего, понадобится включить в нее дополнительные сведения о назначении того или иного правила. Комментарии располагаются между символами /* и */ и игнорируются программами просмотра, например:

body {margin-left: lin;} /* Отступ на 1 дюйм */
h1 {margin-left: -lin;} /* Сдвиг влево на 1 дюйм */
h2 {margin-left: -lin;} /* Сдвиг влево на 1 дюйм */

Комментарии могут иметь вид блоков текста, дающих подробное описание стиля страницы, например:

/*------

Свойство margin-left тэга <body> установлено в 1 дюйм. Так как все внутренние тэги наследуют это значение, то вся страница будет иметь отступ в 1 дюйм. Заголовки первого и второго уровней имеют отрицательный отступ (-1 дюйм), т.е. будут прижаты к левой границе окна браузера.

-----*/
body {margin-left: lin;} /* Отступ на 1 дюйм */
h1 {margin-left: -lin;} /* Сдвиг влево: на 1 дюйм */

h2 {margin-left: -lin;} /* Сдвиг влево на 1 дюйм */

Наследование свойств.

В HTML подчиненные тэги наследуют некоторые свойства родительских тэгов. Например, все тэги контейнера <body> (<p> и <ul>) будут обладать некоторыми свойствами тэга <body>. Точно так же тэг <li> наследует свойства тэга <ul>. Рассмотрим следующий код:

```
<style type="text/css">
p{color: blue;}
</style>
<body>
<p>Hello. This is a paragraph of text. <em>This is emphasized</em><p>
</body>
```

Таблица стилей этого документа устанавливает цвет в тэге <p> синим, однако, цвет для тэга явно не определен (по умолчанию – это черный цвет). Здесь не о чем беспокоиться, так как этот тэг находится в родительском контейнере <p> и наследует таким образом синий цвет.

Применение контекстных селекторов.

Иногда возникает необходимость определения двух (и более) стилей для одного тэга. Например, может понадобиться указание двух стилей для тэга <li>: один для случая, когда он подчинен тэгу <ul>, и второй, когда он подчинен тэгу <ol>. Это возможно сделать с помощью контекстных селекторов. Контекстный селектор определяет точную последовательность тэгов, для которых будет применен тот или иной стиль. Другими словами, вы можете указать, что какой-то стиль должен применяться, например, в тэге <li> только в том случае, если этот тэг является подчиненным тэгу <ol>:

```
ol li{list-style-type: decimal;}
```

Для того же тэга <li> можно определить другой стиль, действительный только в случае подчиненности тэгу <ul>:

```
ul li {list-style-type: square;}
```

Заметьте, что список селекторов не разделен запятыми. В противном случае всем тэгам списка будет приписан один и тот же стиль.

Почему таблицы стилей HTML называются каскадными

В рекомендациях W3C таблицы стилей называются "каскадными таблицами стилей" потому, что для верстки Web-страницы можно применять не одну, а сразу несколько таблиц. При этом программа просмотра сама определяет последовательность использования таблиц и разрешает конфликты между ними по принципу каскадирования.

Как это работает? Каждому правилу браузер приписывает весовой коэффициент. При интерпретации каждого тэга программа просматривает все правила этого тэга и сортирует их по величине весового коэффициента. Выигрывает самое "весомое" правило.

Существуют следующие общие принципы разрешения конфликтов между таблицами стилей:

Таблица стилей автора страницы "весомее" таблицы стилей читателя, которая, в свою очередь, "весомее" установок браузера по умолчанию.

Старшинство типов таблиц стилей в документе (по убыванию): текущее задание стиля (inline), встраивание (embedding) связывание (linking).

Использование классов в таблицах стилей

Классом называется определение нескольких стилей одного элемента, каждый из которых может использоваться в нужном месте страницы. Например, вы можете определить три вариации стиля заголовка h1. Определение вариаций похоже на указание стиля, только к названию тэга добавляется произвольное имя класса, отделенное точкой:

h1.blue {color: blue;}

h1.red {(color: red;}

h1.black {color: black;}

Теперь, включая в документ тэг <h1>, можно указать в нём конкретный стиль при помощи атрибута class:

<h1 class=red>Red Heading</h1>

Вы можете разрешить обратиться к какому-либо классу из любого тэга, если при определении данного класса опустить в селекторе имя тэга, например:

.red {color: red;}

1.10. Блочная верстка страниц

Отличия блочной верстки от табличной

Верстка блоками, с помошью такого тега как div, имеет ряд больших преимущест по сравнению с версткой таблицами (с помощью элемента table), среди них:

1. Дизайн сверстаный блоками быстрее загружается.
2. Содержимое блоков, в отличии от содержимого ячеек таблиц отображается по мере загрузки(содержимое таблиц же напротив, отображается только тогда, когда загрузиться все содержимое таблицы).
3. Код написаный блоками имеет более читаемый вид.

Управление положением блоков на странице

Для решения этого вопроса используется такое свойство как float.

Свойство float может принимать три значения:

1. left - выравнивание элемента по левому краю страницы;

2. right - выравнивание элемента по правому краю страницы;
3. none - элемент страницы ни куда не перемещается, то есть будет там где он должен быть. Это значение используется по умолчани.

Так же нам понадобиться рассмотреть еще одно свойство - clear. Свойство clear может принимать четыре значения:

1. left - установка элемента ниже любого предыдущего, перемещенного влево блока;
2. right - установка элемента ниже любого предыдущего, перемещенного вправо блока;
3. both - установка элемента ниже любого предыдущего перемещенного блока;
4. none - нет ни каких ограничений на положение блока относительно перемещаемых блоков.

1.11. Формы

С помощью HTML вы можете создавать простые формы, предполагающие ответы типа "да" и "нет", вы можете разрабатывать сложные формы для заказов или для того, чтобы получить от своих читателей какие-либо комментарии и пожелания.

Форма представляет собой несколько полей, где пользователь может ввести некоторую информацию, либо выбрать какую-то опцию. После того как пользователь отправит информацию, она обрабатывается программой (скриптом), размещённой на сервере. Скрипт – это короткая программа, специально созданная для обработки каждой формы.

Формы HTML позволят вам получать информацию от читателей.

В форму могут заноситься мнения сторон дискуссионной группы. Круг применения форм HTML ограничивается только вашей фантазией.

Работа с тэгами форм

В HTML существует три тэга для создания различного типа полей в форме: <textarea>, <select> и <input>. Любое их количество может быть размещено в контейнере между тэгами <form> и </form>. Ниже дано их краткое описание (подробнее они будут рассмотрены в соответствующих разделах этой главы).

Таблица 15.

<textarea>	Определяет поле, в которое пользователь вводит многострочную текстовую информацию.
<select>	Позволяет пользователю сделать выбор в окне с полосой прокрутки либо в раскрывающемся меню.
<input>	Обеспечивает некоторые другие виды ввода информации: ввод одной строки текста, установку и

	сброс флажков (checkboxes), выбор переключателя (radio buttons) и нажатие кнопки для отправки данных или очистки формы.

Каждая форма начинается тэгом <form>. В нем нужно определить два атрибута, указывающих используемый скрипт и метод посылки данных:

action	Определяет URL, который примет и обработает данные формы. Если этот атрибут не определен, данные отправляются но адресу страницы, на которой помещена форма.
method	Указывает форме, как послать информацию соответствующей программе обработки (скрипту). Обычно он получает значение post, тогда информация формы посылается отдельно от URL. Если указано значение get, информация формы посылается вместе с URL. Get для посылки данных до 256 символов, а post для посылки данных длинной более 256 символов.

Например:

<form method="post" action="/cgi-bin/comment script">

...

</form>

В этом примере дано указание браузеру отправить заполненную форму для обработки скриптом comment script, расположенным в каталоге cgi-bin вашего сервера, и использовать метод посылки post.

Метод get – данные, посылаемые браузером серверу, включают модифицированный URL адрес: метод(http), servr:port и в конец добавляется символ ?, далее следует строка запроса.

Метод post – запрос серверу посылается как mime-данные, при этом пробелы заменяются символом +, поля разделяются & и символы кодируются в шестнадцатиричном коде в виде %xx.

На странице можно расположить любое число форм, однако нужно следить за тем, чтобы не поместить одну форму в другую.

Тэг <textarea> предназначен для построения поля для ввода многострочной текстовой информации. При помощи атрибутов rows и cols этого тэга можно построить поле любого размера. В контейнере textarea допускается размещать любой текст, который будет выведен в поле ввода по умолчанию.

Поле textarea удобно тем, что пользователь может ввести в него любое количество информации.

Тэг <textarea> имеет следующие атрибуты.

Таблица 16.

name	Обязательный атрибут, определяющий название информации.
rows	Устанавливает высоту поля, т.е. число строк в нём.
cols	Устанавливает ширину поля, т.е. длину строки.

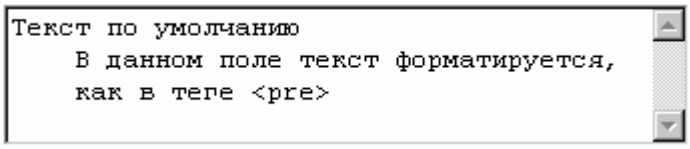
Хотя атрибуты rows и cols не являются обязательными, они не имеют определенных значений по умолчанию (для каждого браузера эти значения различны), поэтому лучше их всегда указывать.

Данный тэг имеет еще один атрибут – wrep. Значения которые он может принимать приведены в Таблице 17.

Таблица 17.

off	Переход на новую строку только при нажатии на Enter. Данные отправляются одной строкой.
soft	Автоматический перенос текста на новую строку, при достижении границы текстового поля. При этом текст на сервер передается одной строкой. Это значение по умолчанию.
hard	Визуально так же как soft, но на сервер будет отправленно несколько строк.

```
<form>
<textarea name="comments" rows=4 cols=40>
Текст по умолчанию
    В данном поле текст форматируется,
    как в теге <pre>
</textarea>
</form>
```



Тэг <select> используется для создания всплывающего меню или списка опций с полосой прокрутки. Список опций и пункты меню располагаются внутри контейнера select. Как и тэг <textarea>, тэг <select> требует обязательного определения имени в атрибуте name. Количество опций указывается в атрибуте size. Ниже перечислены атрибуты тэга <select>.

Таблица 18.

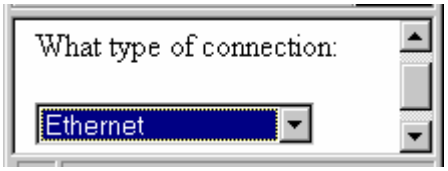
name	Определяет имя поля
size	Определяет вертикальный размер окна для опций выбора. Если атрибут опущен или его значение равно 1, выводится всплывающий список опций(раскрывающийся список). Если указано число больше единицы, то опции выводятся в окне с полосой прокрутки. Если значение атрибута больше, чем фактическое количество элементов списка, добавляются пустые строки. При их выборе пользователем возвращаются пустые поля.
multiple	Этот атрибут позволяет производить выбор сразу нескольких опций.

Список опций включается в контейнер <select> при помощи тэгов <option>. Этот тэг имеет два атрибута.

Таблица 19.

value	Указывает значение, возвращаемое программе обработки (скрипту) в случае выбора опции пользователем.
selected	Указывает на опцию, выделенную по умолчанию.

```
<form>
<select name="network">
<option selected value="ethernet"> Ethernet
<option value="token16"> Token Ring - 16MB
<option value="token4"> Token Ring - 4MB
<option value="localtalk"> LocalTalk
</select>
</form>
```



Тэг <select> можно использовать как дополнительное средство навигации. Для этого в него нужно включить список URL-адресов. После выбора одного из адресов и нажатия кнопки Отправить (Submit), скрипт, размещенный на вашем сервере или на машине читателя, загрузит требуемую страницу.

Тэг <input>, в отличие от <textarea> и <select>, является одиночным тэгом. Он предназначен для сбора информации различными способами, включая текстовые поля, поля для ввода пароля, переключатели, флажки, кнопки для отправки данных (Submit) и для очистки формы (Reset, Clear).

Тэг <input> располагает следующими атрибутами.

Таблица 20.

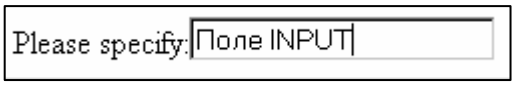
namesize	Указывает размер поля ввода в символах.
maxlenght	Определяет максимально возможное число символов, вводимых в поле.
value	Для текстового поля определяет текст, выводимый по умолчанию. Для флажков и переключателей указывает значение, возвращаемое программе обработки. Для кнопок отправки и очистки формы определяет надпись на кнопке.
checked	Устанавливает флажок или переключатель во включенное состояние по умолчанию. С другими типами тэгов <input> не употребляется.
type	Устанавливает тип поля ввода (см. ниже).

Атрибут type тэга <input> может принимать следующие значения:

1. text

Является значением по умолчанию и предполагает создание одной строки для ввода данных. Для этого типа поля ввода употребляются атрибуты name (обязательный), size, maxlenght и value.

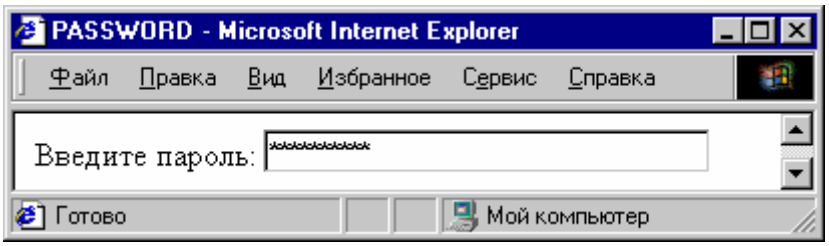
<form>
Please specify:<input type="text" name="network_other">
</form>



2. password

Позволяет заменять вводимые символы пароля звездочками. Для этого типа поля ввода используется атрибуты name (обязательный), size, maxlenght и value.

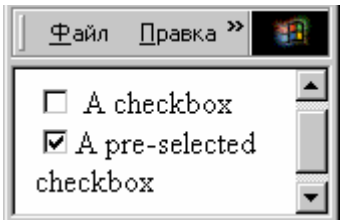
<form>
Введите пароль: <input type="password" name="secret_word" size="10" maxlenght="30">
</form>



3. checkbox

Позволяет вывести поле для установки флажка в виде маленького квадратика, в котором может быть произведена отметка опции "галочкой". Может использоваться совместно с атрибутами NAME (обязательный), VALUE и CHECKED (определяет установленный по умолчанию флажок). Флажки обычно употребляются, когда можно выбрать сразу несколько опций из числа предложенных.

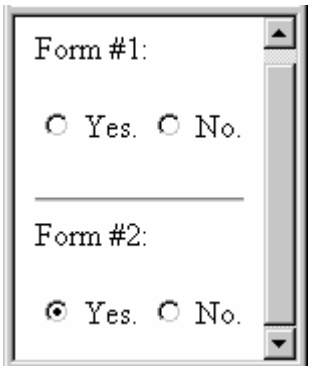
<form>
<input type="checkbox" name="checkbox1" value="checkbox_value1">
A checkbox
<input type="checkbox" name="checkbox2" value="checkbox_value2" checked>
A pre-selected checkbox
</form>



4. radio

Позволяет выбрать только одну из представленного числа опций. Переключатели можно группировать, задавая одно и то же значение атрибута name (обязательный). Также используются атрибуты value и checked. Значение атрибута value отправляется на сервер для обработки скриптом.

```
Form #1:
<form>
<input type="radio" name="choice" value="choice1"> Yes.
<input type="radio" name="choice" value="choice2"> No.
</form>
<hr>
Form #2:
<form>
<input type="radio" name="choice" value="choice1" checked> Yes.
<input type="radio" name="choice" value="choice2"> No.
</form>
```

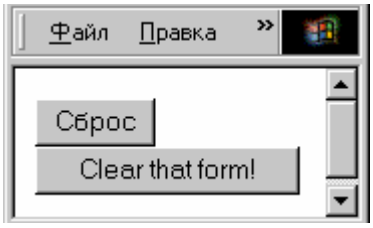


Если опций для выбора слишком много, для экономии места лучше использовать тэг <select>.

5. reset

Позволяет создать кнопку для очистки формы. Атрибут value может быть использован здесь для наименования этой кнопки (по умолчанию кнопка имеет надпись “Сброс” или “Reset”).

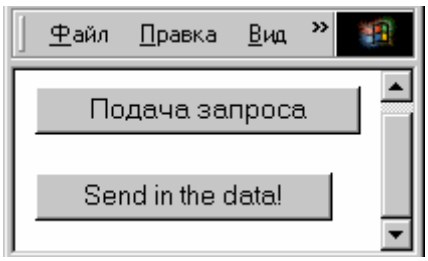
```
<form>
<input type="reset">
<br>
<input type="reset" value="Clear that form!">
</form>
```



6. submit

Используется для создания кнопки, по нажатию которой введенные данные отправляются на сервер для обработки программой-скриптом. В атрибуте value может быть указано название для этой кнопки (по умолчанию – “Submit” или “Подача запроса”).

```
<form>
<input type="submit">
<br>
<input type="submit" value="Send in the data!">
</form>
```



7. button

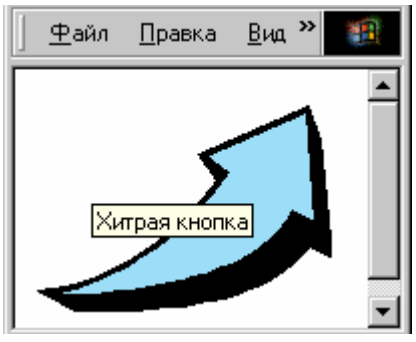
Используется для создания кнопки при нажатии на которую будет выполнятся отдельный скрипт. В качестве параметров можно задать размеры кнопки и ее подпись.

```
<form>
<input  type="button" value="TEST" width="200" height="100">
</form>
```

8. image

Используется для создания кнопки в виде картинки, при щелчке левой кнопкой мыши по которой, все данные из формы будут отправлены на сервер. Таким образом это подобие кнопки submit, только оформленной по другому.

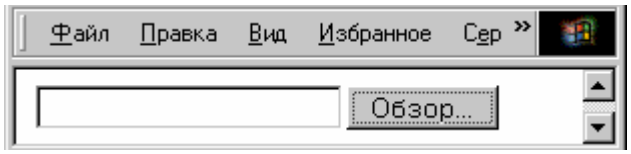
```
<form>
<input type="image" src="trigger.gif" align="right" alt="Хитрая кнопка">
</form>
```



9. file

Используется для обращения к файловой структуре диска и выбора файла, который необходимо переслать по сети. Пересылка осуществляется в двоичном коде.

```
<form>
<input type="file">
</form>
```



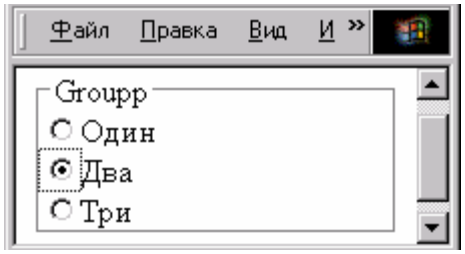
10.hidden

Используется для передачи данных на сервер из многоступенчатых форм без участия пользователя. Данные в этих полях на экране не отображаются.

Выделение нескольких элементов в группу

Парный тэг <fieldset> предназначен для выделения нескольких элементов управления в группу. В паре с этим тэгом работает парный тэг <legent>, при помощи которого задается подпись для этой группы. У тэга <legent> есть атрибут align, который может принимать значения left, right, center и предназначенный для выравнивания подписи группы относительно экрана.

```
<fieldset>
<legend align=left>Groupp</legend>
<input type="radio" name="radio1" value="1">Один<br>
<input type="radio" name="radio1" value="2">Два<br>
<input type="radio" name="radio1" value="3">Три<br>
</fieldset>
```



Размещение в документе форм нескольких типов

Расположение на странице форм разных типов способно сделать ее более выразительной и понятной.

Размещение на странице нескольких форм требует их визуального разделения. Для этого можно использовать тэг <hr>, создающий горизонтальные линии, или тэг <img> для включения узкого горизонтально расположенного изображения.

2. Практика

Постановка задачи

Необходимо создать Web-сайт, состоящий из нескольких (не менее трех) связанных между собой статических HTML-страниц и использующий следующие возможности языка HTML:

- структурирование и оформление текста;
- списки;
- табличная и блочная верстка;
- графические и текстовые ссылки;
- вставка графических объектов, видео и аудио;
- формы с полями различных типов (списки, текстовые поля, переключатели и селекторы), а также кнопками, позволяющими очистить форму и передать ее содержимое на сервер.

Оформление страниц должно выполняться при помощи каскадных таблиц стилей.

Навигация по сайту должна осуществляться с помощью меню.

Все страницы необходимо создавать в виде исходных текстов, не пользуясь графическими средствами разработки.

Для выполнения практической работы Вам необходимо иметь текстовый редактор (возможна работа в специальных редакторах Web-документов, например Adobe Dreamweaver), несколько браузеров для просмотра Ваших страниц (Internet Explorer, Mozilla Firefox, Opera и другие).

Ниже, на Рис. 1-3, приведены примеры нескольких страниц. Подробные инструкции по созданию описанного выше сайта приведены во второй части каждой практической работы (см. раздел “Задание”). В первой части каждой практической работы даны простые упражнения, на примере которых можно рассмотреть отдельные элементы языка HTML.

2.1. Практическая работа №1

Упражнение 1. Изменеие строки заголовка

1. Откройте текстовый редактор. Введите текст:

```
<html>
<head>
<title>Упражнение 1</ title>
</head>
<body>
---  Это моя страница ---
</body>
```

</html>

2. Сохраните документ с именем Ex1.html в рабочей папке
3. Вызовите браузер. Откройте созданный файл. (File → Open →…)

Упражнение 2. Использование заголовков в документе

1. Перейдите в текстовый редактор. Введите текст:

```
<html>
<head>
<title>Упражнение 2</ title>
</head>
<body>
<h1 align=center> Заголовок первого уровня</h1>
здесь идет обычный текст
<h6 align=center> Заголовок последнего уровня</h6>
</body>
</html>
```

2. Сохраните документ с именем Ex2.html в рабочей папке.
3. Вызовите браузер. Откройте созданный файл. (File → Open →…).

Упражнение 3. Логическое форматирование текста на абзацы и отделение их друг от друга горизонтальными линиями.

```
<html>
<head>
<title>Упражнение 2</ title>
</head>
<body>
<p>
Здесь
Вы
видите все
подряд<br>здесь<br>прервали<br>строку<br>
Это предложение отделено от следующего </p> Это другой параграф.
<p> Это один параграф </p><p> Это другой параграф </p>
<h1> <i>Линии</i></h1>
<hr width=50>
< hr width =100>
< hr width =20% align=left size=1>
< hr width =50% align=right size=5>
< hr width =100% align=center size=8 color="#ff0000" noshade >
</body>
</html>
```

Сохраните документ с именем Ex3.html в рабочей папке.

Упражнение 4. Использование упорядоченных и неупорядоченных списков

```
<html>
<head>
<title>Упражнение 4</ title>
</head>
<body>
<h2> <font color="green">Пример упорядоченного списка</font></h2>
Единицы измерения информации
<ol start=2 type=1>
<li>биты
<li>байты
<l>ікилобайты
<br>....
<li value=7>мегабайты </li>
<il>гигабайты
</ol>
<h2> <font color="red">Пример неупорядоченного списка</font>
</h2>
Единицы измерения информации
<ul>
<li>биты
<li>байты
<li>килобайты
<li>мегабайты
<li>гигабайты
</ul>
</body>
</html>
```

Сохраните документ с именем Ex4.html в рабочей папке.

Упражнение 5. Использование внешних гиперссылок

```
<html>
<head>
<title>Упражнение 5</ title>
</head>
<body>
Перейти к  <a href="http://www.limtu.spb.ru/">LIMTU</a>
<br>
Напишите мне  <a href="mailto:info@limtu.spb.ru">по электронной
почте</a>
<br>
```

Открытие диска или папки < a href ="C:\">C:
</body>
</html>

Сохраните документ с именем Ex5.html в рабочей папке.

Упражнение 6. Использование внутренних гиперссылок

<html>
<head>
<title>Упражнение 6</ title>
</head>
<body>
Перейти к упражнению 5
</body>
</html>

Сохраните документ с именем Ex6.html в той же папке, что и документ Ex5.html.

Упражнение 7. Использование ссылок в одном документе

<html>
<head>
<title>Упражнение 2</ title>
</head>
<body>
Глава1 , самая главная,
к ней мы еще вернемся

глава2
 глава3
можно вернуться из той же страницы
возврат
</body>
</html>

Сохраните документ с именем Ex7.html в рабочей папке.

Упражнение 8. Создание файла для возврата к определенному месту документа (якорю)

<html>
<head>

```
<title>Упражнение 8</ title>
</head>
<body>
<a href="Ex7.html#Chapt1">перейдем к 1 главе</a>
</body>
</html>
```

Сохраните документ с именем Ex8.html в той же папке, что и документ Ex7.html.

Задание 1. Создание и наполнение страниц будущего сайта

В данном задании Вы начнете создание страниц будущего сайта.

1. Выберите тему для будущего сайта, предлагается сделать сайт “о себе” или своих увлечениях.
2. Сделайте макет будущего сайта: продумайте, какие страницы войдут в сайт, как они будут взаимосвязаны, какую информацию будут содержать, и как эта информация будет располагаться на каждой странице.
3. Подготовьте наполнение страниц - текстовые фрагменты, графические файлы, видео- и аудио – файлы, приведите информацию к необходимому формату, задайте нужные размеры изображениям. Создайте папку для сайта и скопируйте туда всю подготовленную информацию.
4. В папке сайта создайте HTML-документы, соответсвующие главной и другим страницам сайта.
5. Разместите в документах текстовую информацию, отформатируйте ее нужным образом.
6. Добавьте на страницы сайта навигационное меню и организуйте переходы по гиперссылкам между страницами сайта.

2.2. Практическая работа №2

Упражнение 9. Задание фонового изображения

1. Выберите подходящее для фона изображение и скопируйте его в рабочую папку. Переименуйте его в img1.gif.

```
<html>
<head>
<title>Упражнение 9</ title>
</head>
<body background=”img1.gif”>
```

--- Это моя страница, но уже с интересным фоном ---
</body>
</html>

2. Сохраните документ с именем Ex9.html в рабочей папке.

Упражнение 10. Вставка изображений

1. Выберите подходящее изображение и скопируйте его в рабочую папку. Переименуйте его в img2.gif.

```
<html>
<head>
<title>Упражнение 10</ title>
</head>
<body>
Вставка изображения <br>
<img src="img2.gif">
< img src="img2.gif" width=82 height=68>
< img src="img2.gif" width=200 height=68>
< img src="img2.gif" width=20>
</body>
</html>
```

2. Сохраните документ с именем Ex10.html в рабочей папке.

Упражнение 11. Создания таблицы

```
<html>
<head>
<title>Упражнение 11</ title>
</head>
<body>
<table border cellspacing=0 cellpadding=10>
<caption align=top> Оргтехника
</caption>
 <tr>
        <th>Наименование</th>
        <th> Цена </th>
        <th> Количество </th>
 </tr>
 <tr>
        <th> Принтер </th>
        <td align =right> 350 </td>
        <td align =right> 2 </td>
```

```
  </tr>
  <tr>
        <th> Ксерокс</th>
        <td align =right> 1250 </td>
        <td align =right> 1 </td>
  </tr>
  <tr align=center>
        <th> Факс</th>
        <td align =right> 250 </td>
        <td align =right> 2 </td>
  </tr>
</table>
</body>
</html>
```

Сохраните документ с именем Ex11.html в рабочей папке.

Упражнение 12. Создание вложенных таблиц

```
<html>
<head>
<title>Упражнение 12</ title>
</head>
<body>
 <table border =3 width=200 height=100>
<tr>
        <td>
       <table border =3>
         <tr>
         <td>Ed</ td>
         </tr>
       </table>
        </td>
 </tr>
</table>
</body>
</html>
```

Сохраните документ с именем Ex12.html в рабочей папке.

Задание 2: Верстка одной из страниц сайта и графическое оформление страниц

В ходе выполнения данного практического задания необходимо сверстать одну из страниц сайта и добавить графическую информацию на страницы.

1. Оформите одну из страниц сайта (например, главную страницу), с помощью табличной верстки, то есть используя для взаимного расположения элементов на странице таблицы с невидимыми границами. Для этого продумайте, как будут располагаться элементы на странице относительно друг друга, создайте таблицу и расположите элементы в ее ячейках.
2. Задайте фоновые изображения на страницах, если это необходимо.
3. Поместите графическую информацию на страницы сайта, а также добавьте аудио- и видео-файлы.

2.3. Практическая работа №3

Упражнение 13. Создание и использование каскадных таблиц стилей

1. Внутренние таблицы стилей задают стиль только одному элементу документа при помощи атрибута style в HTML теге.

```
<html>
<head>
<title>Упражнение 13a</title>
</head>
<body>
<p color ="blue" size="3" face="Arial">
Пример физического форматирования.
</p>
<p style=”color:green; font-size:12pt; font-family:Arial”>
Пример встроенной таблицы стилей.
</p>
</body>
</html>
```

2. Сохраните документ с именем Ex13a.html в рабочей папке.

3. Глобальные стили (внедренные таблицы) задают вид элементов всего документа. Для этого используется тег <style type="text/css">. Он размещается в заголовке документа.

```
<html>
<head>
 <title>Упражнение 13b</title>
<style type="text/css">
h1{color:red; font-style:italic; font-size:32px;}
p{color:blue}
```

</style>
</head>
<body>
Пример внедренной таблицы стилей.
<h1> Этот заголовок написан крупным красным курсивом. </h1>
 Это предложение выделено синим цветом.
</body>
</html>

4. Сохраните документ с именем Ex13b.html в рабочей папке.

5. Использование внешних стилевых таблиц — самый экономный и обобщающий способ задания правил оформления однотипных элементов для любого количества страниц. То есть одну таблицу стилей можно использовать для форматирования многих страниц, что приводит к единообразному отображению различных документов и придает некоторую системность серверу разработчика. Осуществляется связывание отдельного файла, содержащего стилевые правила, с множеством гипертекстовых документов. Связываемый файл содержит набор правил.

body{background:grey; font-size:14pt; color:red; font-family:Arial;}
h1, p{color:green;}

6. Сохраните документ с именем example_styles.css в рабочей папке.

7. В самих же HTML документах делается ссылка на этот файл при помощи тега <link>. Выглядит это так: <link rel="stylesheet" type="text/css" href="адрес файла со стилями">

<html>
<head>
<title> Упражнение 13c</title>
<link rel="stylesheet" type="text/css" href="example_styles.css">
</head>
<body>
Пример внешней таблицы стилей
<h1> Этот заголовок выделен зеленым цветом. </h1>
 И это предложение тоже.
</body>
</html>

8. Сохраните документ с именем Ex13c.html в рабочей папке.

Упражнение 14. Использование классов в создании каскадных таблиц стилей

<html>
<head>
<title> Упражнениеи 14</title>
<style>
.blue {color:blue; font-style:italic;}
#boldunderline {text-decoration:underline; font-weight:bold;}

</style>
</head>
<body>
 Здравствуйте, это моя домашняя страница.
 Пока еще в стадии разработки ...
... Но скоро откроется
</body>
</html>

Сохраните документ с именем Ex14.html в рабочей папке.

Упражнение 15. Использование блочной верстки

1. Откройте новый HTML-документ.
2. Создайте в нем три блока разного цвета:

1

2

3

3. Задайте ширину для каждого блока:

1

2

3

4. Теперь предлагается каждому из блоков добавить свойство float со значением, например left, блоки должны выстроиться в один ряд:

1

2

3

5. Попробуйте добавить ниже еще один блок другого цвета:

1

2

3

4
4

6. Так как выше у нас блоки со свойством float, то для того, чтобы четвертый блок оказался под уже созданными тремя, надо для четвертого блока задать свойство clear:

1

2

3

4
4

7. Для того, чтобы убрать отступ между четвертым блоком и первыми тремя добавьте для всей страницы стилевые свойства margin и padding:

```
<style type="text/css">
        *{ margin:0px; padding:0px; }
</style>
```

8. Должен получиться следующий вариант страницы:

```
<html>
<head>
<title>Упражнение 15</title>
<style type="text/css">
  * { margin:0px; padding:0px; }
</style>
</head>
<body>
<div style="background:#red; width:100px; float:left;">1</div>
```

```
<div style="background:#green; width:200px; float:left;">2</div>
<div style="background:#blue; width:300px; float:left;">3</div>
<div style="background:#orange; clear:left;">4<br>4<br>4</div>
</body>
</html>
```

9. Сохраните документ с именем Ex15.html в рабочей папке.

Задание 3: Верстка последующих страниц сайта и стилевое оформление

На данном этапе создания сайта необходимо завершить верстку всех страниц сайта выбранными способами и оформить страницы, используя таблицы стилей.

1. Используйте блочную верстку для одной из отавшихся страниц сайта. Для этого сначала скомпануте блоки на странице, затем поместите в них содержание.
2. Для остальных страниц сайта используйте по желанию либо блочную, либо табличную верстку. При этом заготовьте отдельную страницу, которая может представлять собой либо страницу обратной связи, либо анкету, либо регистрацию, для создания формы.
3. Задайте с помощью внешних каскадных таблиц стилей одинаковой стилевое оформление страниц сайта.
4. Если необходимо, уже с помощью внедренных и встроенных таблиц стилей оформите отдельные элементы на страницах сайта.

2.4. Практическая работа №4

Упражнение 16. Текстовое поле и поле для ввода пароля

```
<html>
 <head>
  <title> Упражнение16 </title>
 </head>
<body>
Содержание Документа
<form>
   <input type=text  name="login" value="user" size=40>
   <input type =password name="pas" value="123" size=10 maxlenght=10>
</form>
</body>
</html>
```

Сохраните документ с именем Ex16.html в рабочей папке.

Упражнение 17. Текстовая область

```
<html>
 <head>
   <title>Упражнение17</title>
 </head>
<body>
<form>
    <textarea  name="comments" rows=5 cols=30>
       Комментарий 1
       Комментарий 2
    </textarea>
</form>
</body>
</html>
```

Сохраните документ с именем Ex17.html в рабочей папке.

Упражнение 18. Работа с флажком

```
<html>
 <head>
   <title>Упражнение18 </title>
 </head>
<body>
<form action=”URL-адрес сервера” method=post enctype=multipart/form-data>
   Выберите нужные вам опции:<br>
1 <input type=checkbox checked name="check1" value=1>
<br>
2 <input type=checkbox  name="check2" value=2>
</form>
</body>
</html>
```

Сохраните документ с именем Ex18.html в рабочей папке.

Упражнение 19. Работа с переключателями

```
<html>
 <head>
   <title> Упражнение19</title>
 </head>
<body>
<form action=”URL-адрес сервера” method=post enctype=multipart/form-data >
```

```
Выберите цвет:<br>
<input type=radio name="r1" value=1>красный<br>
< input type=radio name="r1" value=2>желтый<br>
< input type=radio name="r1" value=3 checked>синий<br>
</form>
</body>
</html>
```

Сохраните документ с именем Ex19.html в рабочей папке.

Упражнение 20. Работа со списками

```
<html>
<head>
 <title> Упражнение20</title>
</head>
<body>
<form action=”URL-адрес сервера” method=post enctype=multipart/form-data >
Владею языком программирования
  <select name=="language" multiple  size=2>
   <option value="A">Java
   < option value="C" selected>C++
   < option value="B">C#
  </select>
<br>
<input type=submit name=”ok”>
<br>
<input type=reset name=”cancel”>
</form>
</body>
</html>
```

Сохраните документ с именем Ex20.html в рабочей папке.

Задание 4: Последние страницы сайта

Сделав данное задание, Вы доделаете страницы своего сайта, состоящего из статических HTML-документов.

1. Доделаем заготовленную раннее для формы страницу. Поместим в отведенное для формы место на странице,саму форму, состоящую из различных элементов. Не забудьте добавить элементы кнопок отправки данных и сброса результатов.
2. С помощью каскадных таблиц стилей оформите вновь добавленные элементы.
3. По желанию добавьте еще несколько страниц сайта и завершите оформление.

Литература

1. Бен Хеник, HTML и CSS. Путь к совершенству. СПб: Питер, 2011 - 336 с.
2. Дунаев В., HTML, скрипты и стили. СПб: БХВ-Петербург, 2011- 816 с.
3. Дронов В., HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. СПб: БХВ-Петербург, 2011 - 416 с.
4. Чак Муссиано, Билл Кеннеди. HTML и XHTML. Подробное руководство. Символ-Плюс, 2011 - 752 с.
5. Кисленко Н.П. HTML. Самое необходимое. СПб: БХВ-Петербург, 2012 – 352 с.
6. Комолова Н., Яковлева Е. HTML, XHTML и CSS. СПб: Питер, 2012 – 304 с.
7. Роберт Агулар. HTML и CSS. Основа любого сайта. Экспо, 2010 – 320 с.
8. Шафер С., HTML, XHTMLи CSS. Библия пользователя. М.: Вильямс, 2010 – 656 с.
9. Лабберс К., Олберс Н., Салим К.. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. М.: Вильямс, 2011 – 272 с.

Интернет-ресурсы

www.w3.org

В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Кафедра Программных систем

Кафедра **Программных систем** входит в состав нового факультета **Инфокоммуникационные технологии**, созданного решением Ученого совета университета 17 декабря 2010 г. по предложению инициативной группы сотрудников, имеющих большой опыт в реализации инфокоммуникационных проектов федерального и регионального значения.

На кафедре ведется подготовка бакалавров и магистров по направлению **210700 «Инфокоммуникационные технологии и системы связи»:**

210700.62.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Бакалавр)

210700.68.10 – ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ (Магистр)

Выпускники кафедры получают фундаментальную подготовку по: математике, физике, электронике, моделированию и проектированию инфокоммуникационных систем (ИКС), информатике и программированию, теории связи и теории информации.

В рамках профессионального цикла изучаются дисциплины: архитектура ИКС, технологии программирования, ИКС в Интернете, сетевые технологии, администрирование сетей Windows и UNIX, создание программного обеспечения ИКС, Web программирование, создание клиент-серверных приложений.

Область профессиональной деятельности бакалавров и магистров включает:

- сервисно-эксплуационная в сфере современных ИКС;
- расчетно-проектная при создании и поддержке сетевых услуг и сервисов;
- экспериментально-исследовательская;
- организационно-управленческая – в сфере информационного менеджмента ИКС.

Знания выпускников востребованы:

- в технических и программных системах;
- в системах и устройствах звукового вещания, электроакустики, речевой, и мультимедийной информатики;
- в средствах и методах защиты информации;
- в методах проектирования и моделирования сложных систем;
- в вопросах передачи и распределения информации в телекоммуникационных системах и сетях;
- в методах управления телекоммуникационными сетями и системами;
- в вопросах создания программного обеспечения ИКС.

Выпускники кафедры Программных систем обладают компетенциями:

- проектировщика и разработчика структур ИКС;
- специалиста по моделированию процессов сложных систем;
- разработчика алгоритмов решения задач ИКС;
- специалиста по безопасности жизнедеятельности ИКС;
- разработчика сетевых услуг и сервисов в ИКС;
- администратора сетей: UNIX и Windows;
- разработчика клиентских и клиент-серверных приложений;
- разработчика Web – приложений;
- специалиста по информационному менеджменту;
- менеджера проектов планирования развития ИКС.

Трудоустройство выпускников:

1. ОАО «Петербургская телефонная сеть»;
2. АО «ЛЕНГИПРОТРАНС»;
3. Акционерный коммерческий Сберегательный банк Российской Федерации;
4. ОАО «РИВЦ-Пулково»;
5. СПБ ГУП «Петербургский метрополитен»;
6. ООО «СоюзБалтКомплект»;

7. ООО «ОТИС Лифт»;
8. ОАО «Новые Информационные Технологии в Авиации»;
9. ООО «Т-Системс СиАйЭс» и др.

Кафедра сегодня имеет в своем составе высококвалифицированный преподавательский состав, в том числе:

- 5 кандидатов технических наук, имеющих ученые звания профессора и доцента;
- 4 старших преподавателя;
- 6 штатных совместителей, в том числе кандидатов наук, профессиональных IT- специалистов;
- 15 Сертифицированных тренеров, имеющих Западные Сертификаты фирм: Microsoft, Oracle, Cisco, Novell.

Современная техническая база; лицензионное программное обеспечение; специализированные лаборатории, оснащенные необходимым оборудованием и ПО; качественная методическая поддержка образовательных программ; широкие Партнерские связи существенно влияют на конкурентные преимущества подготовки специалистов.

Авторитет специализаций кафедры в области компьютерных технологий подтверждается Сертификатами на право проведения обучения по методикам ведущих Западных фирм - поставщиков аппаратного и программного обеспечения.

Заслуженной популярностью пользуются специализации кафедры ПС по подготовке и переподготовке профессиональных компьютерных специалистов с выдачей **Государственного Диплома** о профессиональной переподготовке по направлениям: **"Информационные технологии (инженер-программист)"** и **"Системный инженер"**, а также Диплома о дополнительном (к высшему) образованию с присвоением квалификации: **"Разработчик профессионально-ориентированных компьютерных технологий "**. В рамках этих специализаций высокопрофессиональные преподаватели готовят компетентных компьютерных специалистов по современным в России и за рубежом операционным системам, базам данных и языкам программирования ведущих фирм: Microsoft, Cisco, IBM, Intel, Oracle, Novell и др.

Профессионализм, компетентность, опыт, и качество программ подготовки и переподготовки IT- специалистов на кафедре ПС неоднократно были удостоены **высокими наградами «Компьютерная Элита» в номинации лучший учебный центр России**.

Партнеры:

1. **Microsof**t Certified Learning Solutions;
2. **Novell** Authorized Education Center;
3. **Cisco** Networking Academy;

4. **Oracle** Academy;
5. **Sun Java** Academy и др;
6. **Prometric**;
7. **VUE**.

Мы готовим квалифицированных инженеров в области инфокоммуникационных технологий с новыми знаниями, образом мышления и способностями быстрой адаптации к современным условиям труда.

Т.В. Зудилова, М.Л. Буркова

Web-программирование HTML

ПРАКТИКУМ

В авторской редакции
Редакционно-издательский отдел НИУ ИТМО
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99
Подписано к печати
Заказ №
Тираж 100 экз.
Отпечатано на ризографе

Редакционно-издательский отдел
Санкт-Петербургского национального
исследовательского университета
информационных технологий, механики
и оптики

197101, Санкт-Петербург, Кронверкский пр., 49