

«Национальный открытый институт г.Санкт-Петербург»

Рыбакова Е.А, Путькина Л.В., Саитов А.В.

Обработка отраслевой информации. HTML – язык разметки гипертекстовых страниц

**Методические указания к выполнению
практических работ**

Рекомендовано Методической комиссией по качеству Национального
открытого института
для студентов, обучающихся по специальности
09.02.05 – Прикладная информатика (по отраслям)

Санкт-Петербург
2016

УДК 004
ББК 32.97
Р92

Методические указания разработаны на основе рабочей программы “Прикладная информатика (по отраслям)” в соответствии с требованиями государственных образовательных стандартов среднего профессионального образования.

Контрольная работа выполняется студентами на первом курсе. Методические указания по выполнению контрольной работы содержат описание четырех заданий контрольной работы для студентов специальности – 23.07.01 – Прикладная информатика (по отраслям)

УДК 004
ББК 32.97

© Рыбакова Е.А. 2016
© Пулькина Л.В., 2016
© Саитов А.В., 2016
© Национальный открытый институт 2016
© ИКЦ 2016

ВВЕДЕНИЕ

HTML (*HyperText Markup Language* — «язык гипертекстовой разметки») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах Европейского Центра ядерных исследований в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов — дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <TABLE> предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву

современным потребностям в мультимедийном и графическом оформлении.

Практическая работа 1. Создание простой Web - страницы<HR>

Цель работы: Изучение элементов HTML, использование тегов при создании Web-страницы

Методические указания

В литературе по HTML постоянно встречается слово "тег". *Тег* - это основной элемент кодирования, принятый в стандарте HTML. HTML практически все зависит от тегов. Они служат для привлечения внимания к определенным словам в документе, тем самым давая браузеру понять, что нужно выделить, где отобразить графику и где находятся гиперссылки.

Существуют следующие типы HTML-тегов: теги-контейнеры и пустые теги.

Контейнер - это пара тегов, состоящий из начального (или открывающего) и конечного (или закрывающего) тегов.

Начальный тег имеет вид:

<TAG>,

где TAG - имя реального HTML-тега.

Конечный тег имеет вид:

</TAG>.

Он отличается от начального тем, что перед именем тега косая черта. Между начальным и конечным тегами заключено содержимое контейнера. Например, элемент, представляющий собой подчеркнутый текст, заключается между тегами **<U>** и **</U>**: **<U>** подчеркнутый текст **</U>**.

Также существует возможность вложения одного контейнера в: **<U><H1>** подчеркнутый заголовок первого уровня **</H1></U>**,

Пустой тег отличается от контейнера тем, что не содержит никакой информации. У него есть только начальный тег. Пустой тег

обычно выполняет самостоятельную задачу, не связанную с конкретным текстом. Например:

<HR> - создает горизонтальную линию и не выполняет при этом никаких действий над текстом.

<P> - приводит к вставке интервала между разделами.

Рассмотрим некоторые теги:

<COMMENT> </COMMENT>

Текст комментария. Текст, помещенный внутри COMMENT, игнорируется браузером. COMMENT может располагаться в любом месте кода Web-страницы. Существует ещё один способ обозначения комментария:

<!--Строка комментария-->.

<HTML> </HTML>

Теги элемента HTML являются самыми "внешними", т.е. заключают в себе весь остальной текст и другие теги. Допускает вложение элементов HEAD, BODY и PLAINTEXT, определяющих общую структуру Web-страницы.

<HEAD></HEAD>

Область заголовка Web-страницы. Этот текст представляет собой общую информацию о файле и не отображается в самом документе. В теге <HEAD> может находиться ряд других элементов, включая следующие:

- TITLE (определяет название документа).
- META (сообщает браузеру дополнительную информацию о документе).
- BASE (задает базовый адрес документа).
- SCRIPT (определяет операторы интерпретируемых языков программирования, таких как JavaScript и VBScript)

<TITLE> </TITLE>

Строка текста, расположенная внутри, отображается не в документе, а в заголовке окна браузера. Эта особенность часто используется для организации поиска в WWW.

<META>

Этот элемент содержит служебную информацию, которая не отображается при просмотре Web-страницы. Каждый элемент META содержит два основных атрибута, первый из которых определяет тип данных, а второй - содержание. Например, адрес электронной почты:

name="Reply-to" content="Имя@Адрес"

Имя автора Web-страницы:

name="Author" content="Имя автора".

<BODY> </BODY>

Этот элемент включает в себе гипертекст, который определяет собственно Web-страницу. Это та часть документа, которую разрабатывает автор страницы и которая отображается браузером. Внутри BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы целиком.

Для задания цвета текста: *text="#RRGGBB".*

Цвет текста гиперссылок: *link="#RRGGBB".*

Точно также можно задать цвет для использованных гиперссылок: *vlink="#RRGGBB"* Можно указать изменение цвета для последней выбранной пользователем гиперссылки: *alink="RRGGBB".*

Элемент заголовка <H1> </H1>

Существует шесть уровней заголовков, которые обозначаются H1...H6. Заголовок уровня 1 - самый крупный, а уровень 6 обеспечивает самый маленький заголовок.

Выравнивание абзацев и заголовков

Текст абзаца или заголовка можно выравнивать по левому краю (это стандартная установка), по правому краю или по центру окна браузера (рис. 1). Для тега заголовка используется следующий синтаксис:

`< Hn ALIGN=выравнивание>`.

Здесь n - уровень заголовка ($n = 1...6$), параметр "выравнивание" может принимать значение LEFT (по левому краю), RIGHT(по правому краю), CENTER(по центру).

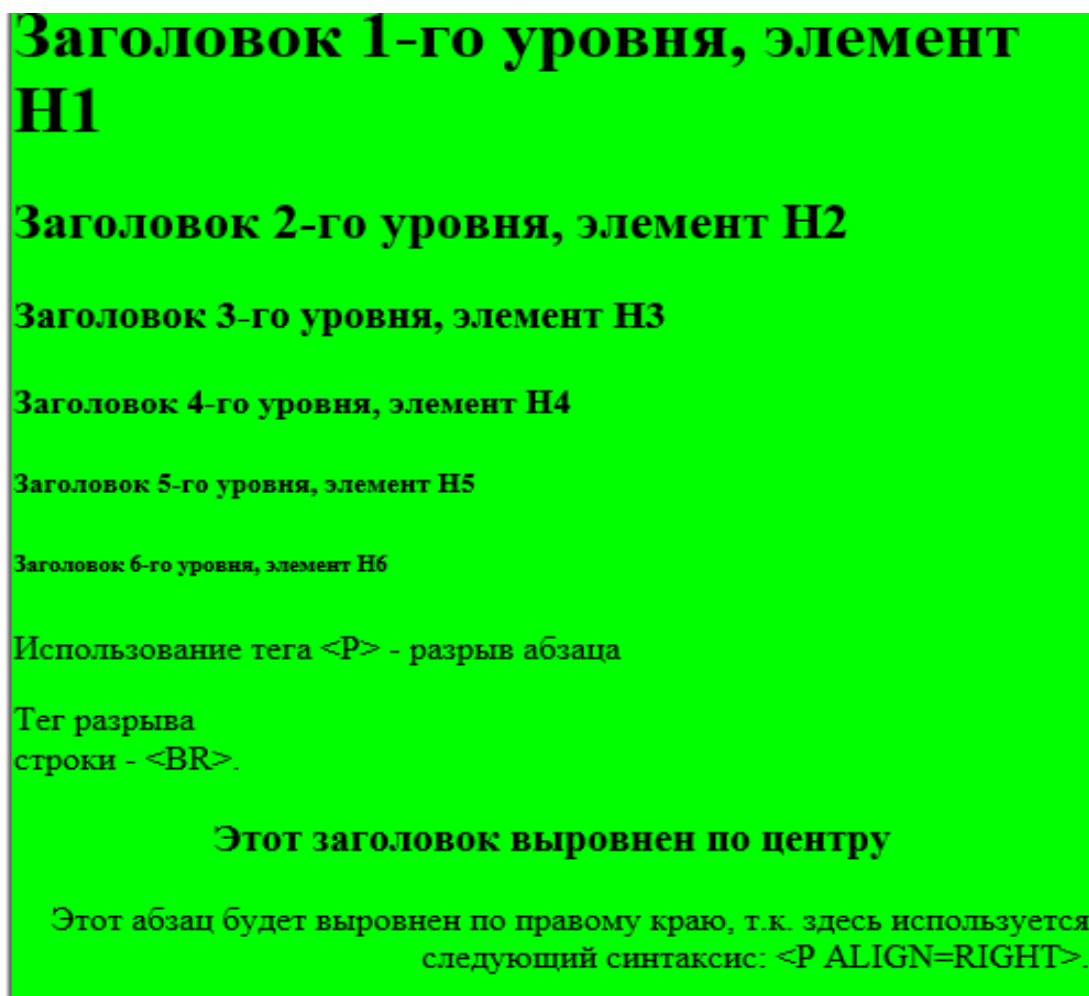


Рис. 1

Для заголовков можно использовать атрибут, задающий выравнивание влево, по центру или вправо:

- `align="left"`
- `align="center"`
- `align="right"`

Для этого используется следующий синтаксис, например: `< H4 align="left">`.

Для тега абзаца используется следующий синтаксис:

`< P ALIGN=выравнивание>`.

Здесь параметр "выравнивание" может принимать значение LEFT, RIGHT, CENTER.

Создание линий `<HR>`

Для разделения текста страницы на части можно воспользоваться горизонтальными линиями. С помощью тега `<HR>` можно создать линию с тенью, протянувшуюся вдоль окна браузера.

Горизонтальная линия (рис. 2) допускает ряд атрибутов:

- `SIZE>` - определяет толщину горизонтальной линии;
- `<WIDTH>` - определяет длину горизонтальной линии;
- `<ALIGN>` - определяет способ выравнивания линии;
- `<NOSHADE>` - определяет черную линию без эффекта

гравировки.

Для атрибута SIZE используется следующий синтаксис:

`< HR SIZE=значение>`.

Здесь "значение" - это толщина горизонтальной линии, определяемая в пикселях.

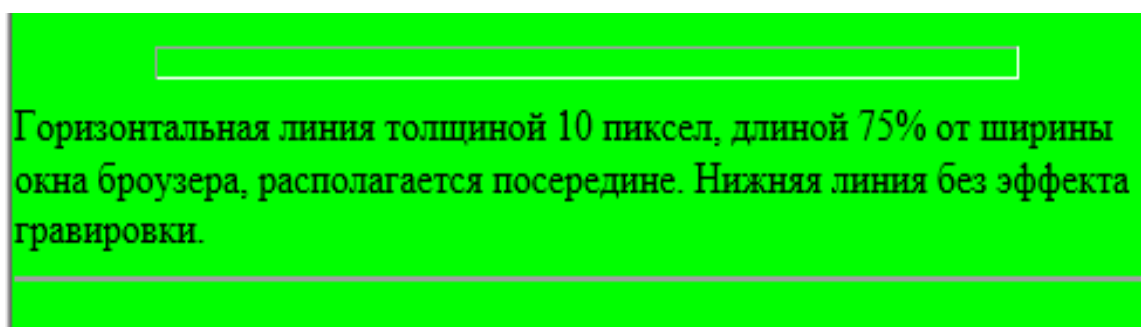


Рис. 2

Величина WIDTH может измеряться как в процентах, так и в пикселях. Поэтому используются 2 варианта синтаксиса:

- `< HR WIDTH="значение_в_процентах%">`;

- `< HR WIDTH="значение_в_пикселях">`.

Атрибут ALIGN может принимать значения LEFT, RIGHT, CENTER. Его нужно использовать в сочетании с атрибутом WIDTH, иначе линия протянется вдоль всего экрана. Пример кода, определяющего горизонтальную линию, которая начинается от левого края экрана, её длина составляет 25% от ширины окна браузера:

`< HR WIDTH="25%" ALIGN=LEFT>`.

Для атрибута NOSHADE используется следующий синтаксис:

`< HR NOSHADE>`.

Линии так же можно выбрать цвет используя тег: `color="Цвет"`.

Например:

`< HR align="left" size=20 width="75%" color="#c0c0c0">`.

Для задания цвета можно воспользоваться таблицами соответствия компьютерного шестнадцатеричного кода и цвета, которые можно найти в интернете.

Порядок выполнения работы

1. Откройте текстовый редактор Блокнот
2. Введите программу для создания Web-страницы

```

<HTML>
<HEAD>
<TITLE>Пример 1</TITLE>
</HEAD>
<BODY BGCOLOR="FFFF00" TEXT="#000000">
<H2 ALIGN=CENTER>Выставка собак</H2>
<HR SIZE=8>
На выставке представлены служебные, охотничьи и декоративные
собаки.
<HR SIZE=3 WIDTH="70%" ALIGN=CENTER>
Служебные собаки: немецкая овчарка, доберман-пинчер,
кавказская овчарка,
сенбернар, южно-русская овчарка, московская сторожевая.
<HR SIZE=3 WIDTH="50%" ALIGN=CENTER>
Охотничьи собаки: такса, спаниель, пойнтер,
русская пегая гончая, западносибирская лайка.
<HR SIZE=3 WIDTH="30%" ALIGN=CENTER>
Декоративные собаки: тойтерьер, папильон, болонка.
<HR SIZE=8>
</BODY>

```

3. Сохраните файл с именем Работа 1. НТМ в свою папку.
4. Для просмотра Web-страницы щелкните левой клавишей мыши по созданному файлу. Вы увидите результат показанный, на рис. 3

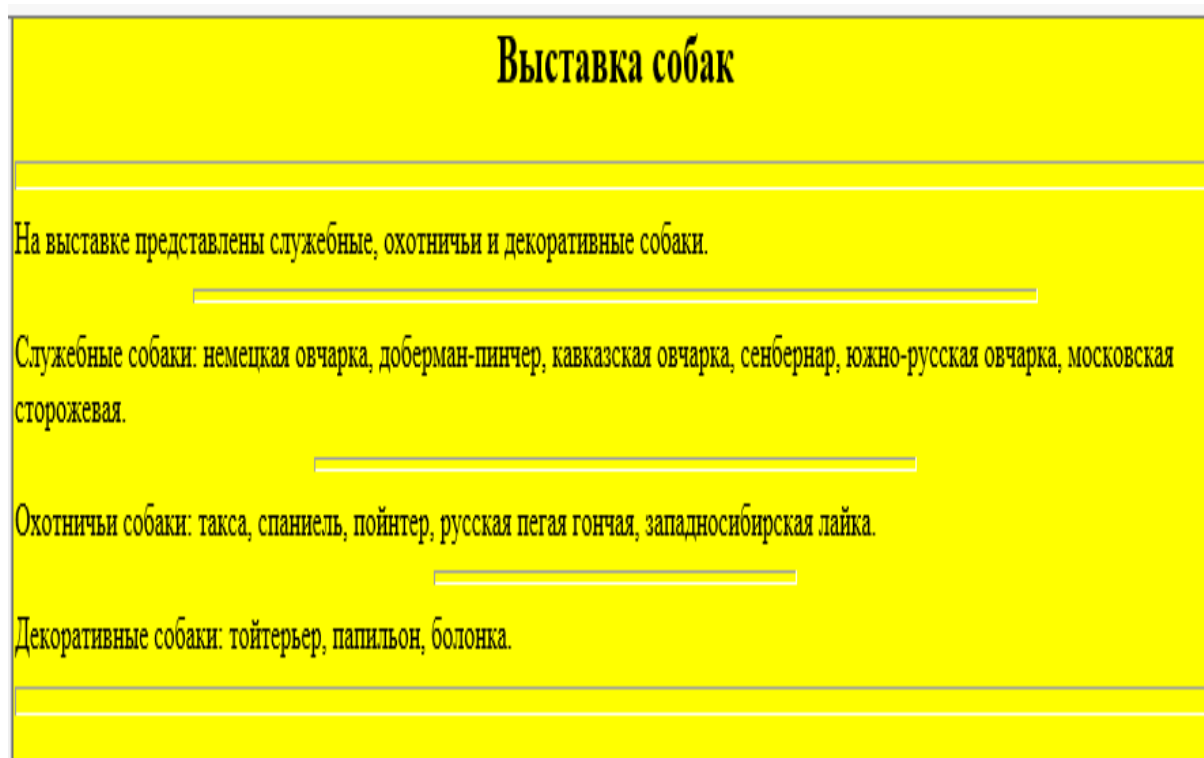


Рис. 3

Контрольные вопросы

1. Что такое теги?
2. Что такое контейнеры, приведите пример.
3. Что такое одиночный тег, как он используется, приведите пример.
4. Какие операции можно проделать над горизонтальной линией?
5. Как создать вертикальную линию?
6. Приведите синтаксис заголовка 6-го уровня, выровненного по правому краю, красного цвета.
7. Напишите базовый шаблон HTML-документа.

Практическая работа 2. Форматирование текста

Цель работы: изучение способов форматирования текста, написание текста в HTML-коде с использованием уже изученных тегов.

Методические указания

**Символы разрыва абзаца и строки <P>,
**

Тег <P> используется для вставки символа разрыва абзаца, а
 - символа разрыва строки. Вставляя символ разрыва абзаца, вы тем самым даете браузеру команду завершить текущий абзац и вставить пустую строку перед следующим абзацем. Вставляя символ разрыва строки, вы даете браузеру команду завершить текущую строку и перейти на следующую. Тег
 можно использовать для создания простых списков:

```
<HTML>
< HEAD>
< TITLE>Пример простого списка<TITLE>
< /HEAD>
< BODY>
< H3>Цитрусовые<H3>
Апельсин<BR>Лимон<BR>Мандарин<BR>
< /BODY>
< /HTML>
```

<PRE> </PRE>

Контейнер отформатированного текста <PRE> позволяет сохранить форматирование текста таким, каким оно было при наборе, поскольку символы перехода на новую строку (полученные в результате нажатия клавиши ENTER) он интерпретирует как разрывы строки. Пробелы внутри текста интерпретируются в точном соответствии с их расположением в текстовом редакторе. Теги абзацев и заголовков, случайно попавшие внутрь контейнера, не интерпретируются. Браузеры отображают отформатированный текст точно также, как он выглядит в исходной программе. Тег <PRE> удобно использовать для помещения на Web-страницу, например,

стихов. Теги <PRE> позволяют также легко представлять данные в виде таблицы. Например:

.....

< BODY>

< PRE>Данные о продажах фирмы "РОГА и КОПЫТА"

Месяцы Рога Копыта

Январь 3556 788

Апрель 5668 543

Август 7664 166

< /PRE>

< /BODY>

.....

Форматирование цитат <BLOCKQUOTE>

</BLOCKQUOTE>, <Q> </Q>

Тег <BLOCKQUOTE> предназначен только для длинных цитат. С помощью этого элемента в блоке текста создается отдельный фрагмент.

А при использовании тега <Q> цитата продолжается в той же строке, что и предыдущий текст, т.е. блок текста не разрывается на части.

Форматирование текста с помощью стилей

В HTML 4.0 предусмотрено несколько способов форматирования текста. Это явное (или абсолютное) форматирование с помощью *физических стилей*; неявное (или относительное) форматирование с помощью *логических стилей*; изменение размера шрифта.

Текст, выделенный физическим стилем, во всех браузерах отображается одинаково (курсив есть курсив, подчеркивание есть подчеркивание), а логические стили можно назвать относительными, т.к. в различных браузерах они отображаются по-разному. Браузер сам решает, как отформатировать текст оптимальным образом по отношению к остальной части Web-страницы.

Оба стиля применяются с помощью контейнеров (т.е. текст, к которому применяется стиль, заключается между начальным и конечным тегами).

Теги физических стилей

- - полужирный шрифт;
- <I> - курсив;
- <TT> - моноширинный шрифт;
- <U> - подчеркивание;
- <SUB> - подстрочный текст;
- <SUP> - надстрочный текст;
- <STRIKE> - зачеркивание.

Теги логических стилей

- - выделенный текст (обычно курсивом);
- - выделенный текст жирным шрифтом;
- <CITE> - текст в виде цитаты;
- <CODE> - текст-фрагмент HTML-кода;
- <DFN> - текст-определение;
- <SAMP> - текст как пример программы;
- <KBD> - текст, который необходимо ввести с клавиатуры;
- <VAR> - текст, определяющий переменную или значение.

Изменение размера шрифта

С помощью тега FONT можно определить размер и цвет шрифта. Атрибут SIZE позволяет указать абсолютный размер шрифта (он может принимать значения от 1 до 7) или относительный, по отношению к размеру шрифта, используемого в основной части страницы (он может принимать значения от -4 до +4). При этом используется следующий синтаксис:

.

Для изменения размера основного шрифта документа используется элемент BASEFONT. Его синтаксис выглядит следующим образом:

< BASEFONT SIZE=значение>.

Величина основного шрифта может принимать значения от 1 до 7, а стандартный для браузера размер шрифта равен 3.

Если Вы хотите узнать об использовании новых атрибутов, или что такое ЛИСТЫ СТИЛЕЙ или, то обратитесь к "Справочнику HTML".

Порядок выполнения работы

1. Введите текст работы в текстовом редакторе Блокнот

```
<HTML>
<HEAD>
<TITLE>Пример 2</TITLE>
</HEAD>
<BODY BGCOLOR="FFFF00" TEXT="#000000">
<H2 ALIGN=CENTER><U>Выставка собак <Q>"Управдом -друг человека"
</Q></U><SUB>(шутка)</SUB></H2>
<HR SIZE=8>
<B>На выставке представлены</B> <I>служебные, охотничьи и декоративные</I>
собаки.
<P>
<FONT COLOR=red SIZE=6>Служебные собаки</FONT>: немецкая овчарка,
доберман-пинчер, кавказская овчарка, сенбернар,
южно-русская овчарка, московская
сторожевая.
<P>
<FONT COLOR=blue SIZE=6>Охотничьи собаки</FONT>: такса, спаниель,
короткошерстная немецкая легавая, пойнтер, русская пегая гончая,
западносибирская лайка.<P>
<FONT COLOR=green SIZE=6>Декоративные собаки</FONT>: тойтерьер,
папильон, болонка, <S>двухцветная сибирская кошка</S>.
<HR SIZE=8>
</BODY>
</HTML>
```

2. Сохраните файл с именем Работа 2.HTM.
3. На рис. 4. показан результат

Выставка собак «"Управдом -друг человека"» (шутка)

На выставке представлены служебные, охотничьи и декоративные собаки.

Служебные собаки: немецкая овчарка, доберман-пинчер, кавказская овчарка, сенбернар, южно-русская овчарка, московская сторожевая.

Охотничьи собаки: такса, спаниель, короткошерстная немецкая легавая, pointer, русская пегая гончая, западносибирская лайка.

Декоративные собаки: тойтерьер, папильон, болонка, двуцветная сибирская кошка.

Рис. 4

Контрольные вопросы

1. Назовите известные вам способы форматирования текста.
2. В чем отличие физического стиля от логического?
3. С помощью какого дескриптора можно вывести на экран стихи (строки смещены относительно друг друга)?
4. Как вывести на экран уравнение 2-го порядка?
5. Выведите на экран простую таблицу, состоящую из заголовка 5-го уровня и 6-ти элементов.

Практическая работа 3. Создание списков

Цель работы: ознакомиться с различными типами списков, а также со способами их составления; создание вложенных и комбинированных списков.

Методические указания

Типы списков

В HTML 4.0 предусмотрены возможности создания различных типов списков. Причем каждый может использоваться как самостоятельно, так и в сочетании с другими типами списков. Существует несколько типов списков (рис. 5):

- упорядоченный (нумерованный) список (рис. 5);
- неупорядоченный (маркированный) список (рис. 5);
- список определений (рис. 6).

Пример нумерованного списка

1. Первый элемент списка
2. Второй элемент списка
3. Третий элемент списка

Пример упорядоченного списка с другим типом нумерации

- A. Первый элемент списка
- B. Второй элемент списка
- C. Третий элемент списка

Пример маркированного (неупорядоченного) списка

- Первый элемент списка
- Второй элемент списка
- Третий элемент списка

Рис. 5

Нумерованные списки

Нумерованный список создается с помощью контейнера `<OL>` `</OL>`. Каждый элемент списка определяется внутри этого

контейнера с помощью тега . При создании нумерованного списка используется следующий синтаксис:

< OL>

< LI> 1-ый элемент списка

.....

< LI> *n*-ый элемент списка

.

Пример списка определений

Первый элемент списка

Определение первого элемента

Второй элемент списка

Первое определение второго элемента

Второе определение второго элемента

Третий элемент списка

Определение третьего элемента

Пример вложенного списка

1. Первый элемент списка

2. Второй элемент списка

◦ Подэлемент 1 второго элемента

◦ Подэлемент 2 второго элемента

◦ Подэлемент 3 второго элемента

3. Третий элемент списка

Рис. 6

Каждый тег служит для определения одного элемента списка. По умолчанию элементы списка нумеруются с помощью обычных арабских цифр. Но можно сделать так, чтобы использовались различные типы букв. Существуют следующие типы нумерации упорядоченных списков:

- 1 - арабские цифры
- а - строчные буквы алфавита
- А - прописные буквы алфавита
- i - строчные римские цифры
- I - прописные римские цифры

Стиль форматирования списка определяется с помощью следующего синтаксиса:

`< OL TYPE=тип>.`

Маркированные списки

Самым распространенным типом списка является маркированный. При создании маркированных списков используются теги `<UL>` и `</UL>`. Как и в нумерованных списках, все элементы маркированных списков определяются с помощью тега `<LI>`. Когда браузер встречает очередной тег `<LI>`, он переходит на новую строку.

Чтобы изменить тип маркера для всего списка, нужно воспользоваться таким синтаксисом:

`< UL TYPE=тип>.`

А для одного элемента:

`< LI TYPE=тип>.`

Существуют следующие типы маркеров:

- Disk - заполненный кружок
- Square - заполненный квадратик
- Circle - незаполненный кружок

Списки определений

Списки определений чем-то напоминают толковые словари, т.к. каждому термину соответствует абзац определения, расположенный с отступом. Это средство незаменимо в том случае, если нужно создать список элементов с их подробным описанием.

Список определений содержится в контейнере `<DL>` `</DL>`. Элементы списка определений создаются с использованием тегов `<DT>` и `<DD>`, с помощью которых отмечаются термин и его определение. Например:

```
< DL>
< DT> термин1
< DD> определение1
< DT> термин2
< DD> определение2
< /DL>.
```

Внутри элемента <DD> могут находиться нумерованные или маркированные списки. Каждому <DT> может соответствовать несколько <DD>.

Вложение и комбинирование списков

При желании можно комбинировать различные типы списков, а также вкладывать их друг в друга независимо от того, к одному или к разным типам они относятся.

Порядок выполнения работы

1. Создайте файл с текстом как указанный на рис. 7
2. Сохраните файл с именем Работа 3. НТМ
3. Результат представлен на рис. 8.

Контрольные вопросы

1. Какие типы списков вы знаете?
2. Какие типы нумерации упорядоченных списков вы можете назвать?
3. Можно ли использовать буквы русского алфавита в упорядоченных списках?
4. Возможно ли использование нескольких <DD> в одном <DT> списка определений?
5. Обязательны ли конечные теги , , </DL>, обоснуйте.

```

<HTML>
<HEAD>
<TITLE>Пример 3</TITLE>
</HEAD>
<BODY BGCOLOR="FFFF00" TEXT="#000000">
<H2 ALIGN=CENTER><U>Выставка собак <Q>"Управдом - друг человека"
</Q></U><SUB>
(шутка)</SUB></H2>
<HR SIZE=8>
<B>На выставке представлены:</B><BR>
<OL>
<LI><FONT COLOR=red SIZE=5><I>Служебные собаки</I></FONT>
<UL>
<LI>немецкая овчарка
<LI>доберман-пинчер
<LI>кавказская овчарка
<LI>сенбернар
<LI>южно-русская овчарка
<LI>московская сторожевая
</UL>
<LI><FONT COLOR=blue SIZE=5><I>Охотничьи собаки</I></FONT>
<UL TYPE=Square>
<LI>такса
<LI>спаниель
<LI>короткошерстная немецкая легавая
<LI>пойнтер
<LI>русская пегая гончая
<LI>западносибирская лайка
</UL>
<LI><FONT COLOR=green SIZE=5><I>Декоративные собаки</I></FONT>
<UL TYPE=Disk>
<LI>тойтерьер
<LI>папильон
<LI>болонка
<LI><S>двухцветная сибирская кошка</S>
</UL>
</UL>
<HR SIZE=8>
</BODY>
</HTML>

```

Рис. 7

Выставка собак «"Управдом -друг человека"» (шутка)

На выставке представлены:

1. *Служебные собаки*

- немецкая овчарка
- доберман-пинчер
- кавказская овчарка
- сенбернар
- южно-русская овчарка
- московская сторожевая

2. *Охотничьи собаки*

- такса
- спаниель
- короткошерстная немецкая легавая
- пойнтер
- русская пегая гончая
- западносибирская лайка

3. *Декоративные собаки*

- тойтерьер
- папильон
- болонка
- двуцветная сибирская кошка

Рис. 8

Практическая работа 4. Гипертекстовые ссылки

Цель работы: изучение различных типов гиперссылок и их использование при создании Web-страниц.

Методические указания

Типы гиперссылок

Существует 3 основных типа гиперссылок:

- внутренние,
- внешние,
- относительные.

Внутренние ссылки - это ссылки на объекты в пределах одного документа. С их помощью пользователь может перемещаться по одной и той же Web-странице. Данный тип ссылок полезно использовать на длинной странице, чтобы иметь возможность быстро перемещаться между ее разделами.

Внешние ссылки - это ссылки на другие Web-серверы.

Относительные ссылки - это ссылки на другие Web-страницы (или службы Internet), расположенные на одном сервере со страницей, содержащей ссылки. Подобные ссылки называются относительными потому, что их адреса даются относительно адреса Web-страницы, на которой содержится ссылка.

По умолчанию в большинстве браузеров ссылки отображаются в виде подчеркнутых текстовых строк.

Понятие об URL и их использование

В каждой ссылке содержится URL (Uniform Resource Locator), или *унифицированный локатор ресурсов*. URL - это адрес Web-страницы, который отображается в поле "Адрес", расположенном в левом верхнем углу окна браузера.

URL состоит из двух основных элементов: типа протокола и собственно адреса.

Тип протокола определяет вид ресурса Internet, с которым устанавливается связь. Самым распространенным протоколом является http:// (HyperText Transfer Protocol - протокол передачи

гипертекста), с помощью которого по Web передаются HTML-документы.

FTP (File Transfer Protocol) расшифровывается как *протокол передачи файлов* и является средством пересылки файлов от одного компьютера к другому.

Собственно адрес - это имя файла, каталога или компьютера. Приведем пример URL:

http://www.greatstar.com/lois/index.html.

В этом URL определяется сервер, на котором находится HTML-документы, каталог и имя файла. URL состоит из нескольких компонентов:

- http:// - определяет тип используемой службы Internet;
- www - представляет собой имя Web-сервера;
- greatstar.com - определяет домен, в котором находится сервер ;
- /lois/index.html - определяет каталог на Web-сервере, в котором хранится файл.

Создание гиперссылок на другие Web-страницы

Большинство гиперссылок сами по себе являются частью *элемента привязки*, который создается с помощью тега <A>. Этот тег можно использовать в основной части HTML-документа, но только не внутри пары тегов <HEAD> </HEAD>. Самым распространенным атрибутом тега <A> является HREF. Его значением является путь к файлу (или URL файла), на который делается ссылка. Элемент привязки всегда завершается конечным тегом . Сам URL должен заключаться в кавычки и следовать за определением элемента привязки. В HTML для гиперссылки используется следующий формат:

< A HREF="URL">текст ссылки.

Чтобы связать текст "Выбор свадебных платьев" с HTML-документом под именем "платья.html", который находится в каталоге "свадьба" на Web-сервере "мой_сервер.com", HTML-код должен выглядеть следующим образом:

Выбор свадебных платьев

Связывая страницы одного сервера, нет необходимости включать в URL доменное имя. Предположим, страницы "костюм.html" и "платья.html" находятся в одном каталоге. В этом случае нужно создать следующий относительный адрес URL:

< A HREF="КОСТЮМ.HTML">Выбор костюма.

При создании гиперссылок нужно помещать их коды непосредственно в текст, ни в коем случае не прерывая изложение материала, чтобы пользователь сам решал, щелкать на ссылки, или нет. Например:

Правильный выбор свадебного
 платья .

- это сложная проблема для любой невесты. Жениху намного проще выбрать себе

костюм

для предстоящей церемонии бракосочетания.

Создание внутренних ссылок

Создавая Web-страницу, вы должны предусмотреть для пользователя возможность легко перемещаться от одного раздела к другому. Это достигается путем включения *внутренних ссылок* и имеет очень важное значение в случае длинной Web-страницы, состоящей из нескольких разделов.

Для внутренней ссылки, как и для внешней, нужно создать текст, на котором будет щелкать пользователь. И точно также нужно включить описание того места, на которое делается ссылка. Для этого место необходимо как-то обозначить. Причем имя, определенное с помощью тега , не будет отображаться в браузере.

Создавая ссылку на другую часть страницы, воспользуйтесь символом (#) и именем привязки. Для этого вы можете выбрать любое имя. В результате получится так называемая объектная привязка:

< A HREF="#торт">Рецепты тортов

Теперь нужно создать соответствующее имя объекта. После щелчка на этой ссылке браузер будет выполнять поиск по Web-странице, пока не найдет ту часть, которой присвоено имя "торт". Эта часть страницы должна быть названа соответствующим образом:

```
< A NAME="торт">Рецепты тортов-коллекция</A>  
Рассмотрим следующий пример:  
< A HREF="#торт">Рецепты тортов</A><BR>  
< A HREF="#салат">Рецепты салатов</A><BR>  
< P>  
< A NAME="торт">Рецепты тортов-коллекция</A><BR>  
(набор рецептов)<BR><BR>  
< A NAME="салат">Рецепты салатов-коллекция</A><BR>  
(набор рецептов)<BR><BR>
```

Ссылки на ресурсы мультимедиа

Ссылки на файлы .mid, .wav, .txt создаются также, как и на любые другие HTML-документы. Можно сделать так, чтобы после щелчка на некоторой ссылке раздавался какой-нибудь звук или воспроизводилась музыка. Для этого в теге <A> присвойте атрибуту HREF имя файла, в котором содержится звук или мелодия и сделайте ссылку:

Щелкните здесь , чтобы услышать новую песню.

Точно так же с помощью ссылки можно отобразить на экране текстовый файл:

Щелкните здесь , чтобы увидеть текст.

Создание ссылок на адреса электронной почты

При разработке Web-страницы многие авторы любят включать в них ссылки на свои адреса электронной почты. Это делается для того, чтобы посетители Web-страницы могли отправить авторам отзывы и замечания о странице.

Адреса состоят из 4 элементов:

- *Имя пользователя* - это имя, которое вы вводите при регистрации в системе.
- *Доменное имя* - обозначает организацию, предоставившую вам доступ к Internet.

• @ - используется в адресах электронной почты для разделения доменного и пользовательского имен.

Например, имя - lrpatter; доменное имя - greatstar.com. Для создания ссылки на правильный адрес электронной почты необходимо поставить перед ним запись mailto:. Например:

< A HREF="mailto:lrpatter@greatstar.com">Почта

Создание ссылки на FTP-сервер

Протокол передачи файлов, или FTP, чаще всего используется для копирования файлов с одного компьютера на другой. Для этого пользователя должен зарегистрироваться на удаленном компьютере (чаще всего - с гостевыми правами) и скопировать нужные файлы с FTP-сервера. Для создания ссылки на FTP-сервер нужно только указать его имя. Для этого необходимо набрать в теге привязки "ftp://имя_FTP-сервера/". Например:

FTP-сервер фирмы Microsoft.

Если вы предлагаете посетителям загрузить с вашего сервера некоторый файл, то должны точно указать путь к нему. Допустим, в вашем каталоге /users/myself/ есть программа PROGRAM.EXE, которую вы хотите распространять бесплатно. Тогда ссылка на нее будет выглядеть так:

Моя программа

Эта ссылка сообщает Web-браузеру, что нужно подключиться к FTP-серверу, найти указанный каталог и немедленно приступить к загрузке файла.

Использование тега <BASE> для упрощения URL

Тег <BASE> используется для создания *относительных URL*. Это поможет значительно упростить процесс создания Web-страниц. Сначала нужно определить основной URL-адрес с помощью тега <BASE>. Этот тег используется внутри контейнера <HEAD> </HEAD>. Например:

< HEAD>

```
< BASE HREF="http://www.имя_сервера.com/мой_каталог/">  
< /HEAD>
```

При включении в тег <HEAD> элемент <BASE> может содержать только атрибут HREF, которому в качестве значения присваивается основной URL документа. А уже на основе этого URL создаются относительные адреса. Преимущества: поскольку относительные URL указываются неявно, в случае изменения основного адреса можно быстро внести необходимые изменения. Теперь для приведенного выше примера создадим относительный URL-адрес:

```
<A HREF="товары.html"Каталог наших товаров</A>.
```

На самом деле здесь делается ссылка на гипотетический URL http://www.имя_сервера.com/мой_каталог. Таким образом, если определить основной URL, то относительно него можно создавать относительные URL.

Порядок выполнения работы

1. Найти Картинки в Интернете и скопируйте в свою папку с работами.
2. Создайте файл, как показано на рис. 9.
3. Сохраните файл с именем Работа 4.НТМ

Результат представлен на рис. 10. Для того чтобы появилась картинка необходимо щелкнуть левой клавишей мыши **по подчеркнутому тексту**

Контрольные вопросы

1. С помощью какого тега создается большинство гиперссылок и где он размещается?
2. Что такое URL, каким он может быть?
3. Как сделать ссылку на чужую Web-страницу?
4. Из каких элементов состоит адрес электронной почты и как делается на него ссылка?
5. Что такое FTP-сервер, его назначение?

```

<HTML>
<HEAD>
<TITLE>Пример 4</TITLE>
</HEAD>
<BODY BGCOLOR="FFFF00" TEXT="#000000">
<H2 ALIGN=CENTER><U>Выставка собак <Q>"Управдом - друг
человека"</Q></U><SUB>
(шутка)</SUB></H2>
<HR SIZE=8>
<B>На выставке представлены</B> <I>служебные, охотничьи и
</I><A HREF="#дек"><I>декоративные</I></A> собаки.
<P>
<FONT COLOR=red SIZE=6>Служебные собаки</FONT>: немецкая
овчарка,
<A HREF="dob.gif">доберман-пинчер</A>, кавказская овчарка,
сенбернар,
южно-русская овчарка, московская сторожевая.
<P>
<FONT COLOR=blue SIZE=6>Охотничьи собаки</FONT>: такса, спаниель,
<A HREF="leg.gif">короткошерстная немецкая легавая</A>,
пойнтер, русская пегая
гончая, западносибирская лайка.
<P>
<A NAME="дек"><FONT COLOR=green SIZE=6>Декоративные
собаки</FONT></A>:
тойтерьер, <A HREF="pap.gif">папильон</A>, болонка, <S>двухцветная
сибирская
кошка</S>.
<HR SIZE=8>
</BODY>
</HTML>

```

Выставка собак «"Управдом -друг человека"» (шутка)	
На выставке представлены служебные, охотничьи и <u>декоративные</u> собаки.	
Служебные собаки:	немецкая овчарка, <u>доберман-пинчер</u> , кавказская овчарка, сенбернар, южно-русская овчарка, московская сторожевая.
Охотничьи собаки:	такса, спаниель, <u>короткошерстная немецкая легавая</u> , пойнтер, русская пегая гончая, западносибирская лайка.
Декоративные собаки:	тойтерьер, <u>папильон</u> , болонка, двуцветная сибирская кошка.

Рис. 10

Практическая работа 5. Таблицы

Цель работы: изучение форматирования данных с помощью таблиц, их использование при создании Web-страниц.

Методические указания

Создание строк и столбцов таблицы

Код таблицы в HTML содержится в контейнере TABLE.

Каждая пара тегов <TR> и </TR> определяет одну строку таблицы. Они сообщают браузеру, что ячейки, определяемые между тегами <TR> и </TR>, относятся к одной строке таблицы.

Поскольку существует тег </TR>, теги </TD> (данные таблицы) и </TH> (заголовки таблицы) использовать необязательно. Приведем пример двух строк:

```
< TR><TD>яблоки<TD>бананы</TR>,
< TR><TD>вишни<TD>финики</TR>.
```

Выбор типа таблицы

Одной из общих характеристик таблицы является *рамка*. Для указания ширины рамки используется атрибут `BORDER=n`, где *n* - ширина рамки в пикселях. Атрибут `BORDER` помещается в тег `<TABLE>`. По умолчанию атрибуту будет присвоено значение 1. Если выбрать значение 0 или вообще опустить атрибут `BORDER`, то будет создана таблица без рамки.

Для определения *ширины таблицы* используется атрибут `WIDTH`, помещенный в тег `<TABLE>`. По умолчанию ширина таблицы вычисляется как сумма ширин всех столбцов. А ширина каждого столбца (если не выбрать другую установку) по умолчанию определяется так, чтобы в ячейке поместился самый длинный элемент данных или самая большая картинка. Если определить, что таблица занимает, например, 90% пространства экрана Web-браузера, то в ней будет больше свободного места. Для этого нужно добавить атрибут `WIDTH` в тег `<TABLE>`:

```
< TABLE BORDER=2 WIDTH="90%">.
```

Можно также присвоить атрибуту `WIDTH` фиксированное значение в пикселях.

С помощью атрибута `COLS`, помещенного в тег `<TABLE>`, можно определить количество столбцов таблицы:

```
< TABLE BORDER=0 WIDTH=600 COLS="3">.
```

Для определения *интервалов* в таблице используются атрибуты `CELLSPACING` и `CELLPADDING`, помещенные в тег `<TABLE>`. Атрибут `CELLSPACING` определяет расстояние между ячейками в пикселях (стандарт=2). Атрибут `CELLPADDING` определяет расстояние в пикселях между содержимым ячейки и ее границами (стандарт=1). Например:

```
< TABLE BORDER WIDTH=300 CELLSPACING=4  
CELLPADDING=5>.
```

Определение ячеек данных и заголовков

Существует два основных элемента таблицы: ячейки данных и ячейки заголовков. Ячейка данных определяется с помощью контейнера `TD`, в который и помещаются данные. Ячейка заголовка определяется с помощью контейнера `TH`. Текст, помещенный в этот

контейнер, отображается полужирным шрифтом. Теги <TD> и <TH> находятся внутри тега <TR> и определяют отдельные ячейки в пределах строки таблицы. В конце определения каждой строки используется тег </TR>.

Увеличение размеров ячеек с помощью атрибутов COLSPAN и ROWSPAN

Обычно в каждой строке таблицы содержится одинаковое число ячеек. Но иногда возникает необходимость растянуть ячейку на несколько строк или столбцов (рис. 11). Для определения растянутой на несколько столбцов ячейки используется атрибут COLSPAN, помещенный в тег <TH> или <TD>. Этому атрибуту нужно присвоить значение, соответствующее количеству столбцов, вдоль которых растянута ячейка.

Пример:

```
< TR>
< TH COLSPAN=2>животные
</TR>
< TR>
< TH>собаки</TH>
< TH>кошки</TH>
< /TR>
< TR>
< TD>5</TD>
< TD>14</TD>
< /TR>.
```

Если вы определили ячейку, растянутую на несколько строк, то в описании этих строк уже не нужно указывать, что соответствующие ячейки пусты. Надо просто пропустить определение этих ячеек.

Пример таблицы с ячейкой, растянутой на 2 строки

Ячейка 1	
Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5

Пример таблицы с ячейкой, растянутой на 2 столбца

		Ячейка 1
Ячейка 2	Ячейка 3	Ячейка 4
	Ячейка 5	Ячейка 6

Рис. 11

Для создания ячейки, растянутой на несколько столбцов, необходимо добавить атрибут COLSPAN в определение крайней ячейки слева в наборе объединяемых ячеек. А для получения ячейки, растянутой на несколько строк, необходимо добавить атрибут ROWSPAN в определение самой верхней ячейки.

Выравнивание данных в ячейках таблицы по горизонтали

Атрибут ALIGN позволяет выровнять содержимое ячеек и может принимать следующие значения:

- CENTER - по центру;
- JUSTIFY - текстовое содержимое ячейки будет выровнено по ширине (то есть одновременно по левому и правому полям) в том случае, если в ячейке содержится несколько слов;
- CHAR - содержимое ячейки будет выравниваться относительно символа, заданного с помощью атрибута CHAR. Например, запись ALIGN="char"; CHAR="," можно использовать для выравнивания натуральных чисел по разделителю дробной части.

Атрибут CHAROFF позволяет сдвигать данные, которые не содержат заданного для выравнивания символа (величина сдвига измеряется в символах);

- RIGHT – вправо;
- LEFT – влево.

Выравнивание данных в ячейках по вертикали

Атрибут VALIGN может использоваться в контейнере <TR>, <TH>, <TD>. С его помощью можно выровнять текст верхней или нижней границе ячейки, а также отцентрировать по вертикали. Помещенный в тег <TR>, атрибут VALIGN указывает, что все ячейки строки должны быть выровнены по вертикали относительно базовой линии. Используя атрибут WIDTH в теге <TABLE>, можно включить ячейки таблицы больше свободного пространства, чтобы сделать ее более читабельной.

Добавление подписей к таблицам

С помощью контейнера <CAPTION> к таблицам можно добавлять подписи (рис.9). По умолчанию подписи располагаются над таблицей, но их можно поместить и в другое место. Для размещения подписей под таблицей используется следующий код:

```
<TABLE>  
<CAPTION ALIGN=BOTTOM> Это подпись </CAPTION>  
<TR>.....
```

Помимо атрибута BOTTOM, в контейнере CAPTION можно использовать атрибуты:

- TOP - подпись над таблицей;
- LEFT - подпись слева от таблицы;
- RIGHT - подпись справа от таблицы.

Окрашивание таблиц и рамок

К таблице можно добавить фон, цвет которого отличается от цвета остальной части Web-страницы (рис. 12). Для этого нужно использовать тег <TABLE> с атрибутом BGCOLOR. Значением атрибутов BGCOLOR может быть как шестнадцатеричное число, так

и текст. Атрибут BGCOLOR может применяться в тегах <TABLE>, <TR>, <TD>. Например:

<TR BGCOLOR=yellow>.

С помощью тега <TABLE> и атрибутов BORDERCOLOR и BORDERLIGHT можно изменить цвет рамки таблицы. Атрибут BORDER должен быть помещен в тег <TABLE>. Чтобы изменить цвет рамки внутри таблицы, нужно использовать атрибуты BORDERCOLOR и BORDERLIGHT в теге <TR> или <TD>. Значениями этих атрибутов могут быть либо шестнадцатеричное число в виде #rrggbb, либо название цвета, например:

<TABLE BORDER=3 BORDERCOLOR=blue BORDERLIGHT=lightblue>.

Значение атрибута BORDERCOLOR влияет на верхнюю часть рамки трехмерной таблицы, а величина BORDERLIGHT - на нижнюю (на тень).

Пример таблицы с подписью, занимающей 70% ширины окна браузера

Таблица 1

Товар	Цена	
	1-9 шт.	Опт.
Корзина печенья	1,99	250,00
Бочка меда	2,00	400,00
Ложка дегтя	3,00	500,00

Рис. 12

Макетирование страниц с помощью таблиц

Определив таблицу со всем ее содержимым, но без рамки и присвоив этой таблице атрибут WIDTH="100%", получим таблицу во всю ширину окна браузера. Затем определив вложенную таблицу с рамками, можно получить великолепный макет страницы. Карты ссылок тоже прекрасно вписываются в "табличную" структуру

страницы. Чтобы вложить одну таблицу в другую, нужно включить определение вкладываемой таблицы в контейнер TD.

Пример:

```
<TABLE>
<TR>
<TD>ячейка 1
<TD>
<TABLE>
<TR><TD>одна ячейка внутри другой
</TABLE>
</TABLE>.
```

Использование атрибутов таблиц

Атрибут **FRAMES** позволяет определить, какие из внешних рамок таблицы будут отображаться на экране. Он может принимать следующие значения:

- **BORDER** - отображаются все рамки таблицы
- **VOID** - все внешние рамки удаляются;
- **ABOVE** - отображается только рамка с верхней стороны фрейма;
- **BELOW** - отображается только рамка с нижней стороны фрейма;
- **HSIDES** - отображаются все горизонтальные рамки, верхняя и нижняя;
- **VSIDES** - отображаются все вертикальные рамки, левая и правая;
- **LHS** - отображается рамка с левой стороны таблицы;
- **RHS** - отображается рамка с правой стороны таблицы;
- **BOX** - отображается рамка по всему контуру таблицы.

Атрибут **RULES** позволяет контролировать отображение внутренних рамок. Он может принимать следующие значения:

- **NONE** - все внутренние линии таблицы удаляются;
- **GROUPS** - отображаются горизонтальные линии между группами таблиц;

- ROWS - отображаются горизонтальные линии между всеми строками таблицы;
- COLS - отображаются вертикальные линии между всеми столбцами таблицы;
- ALL - отображаются линии между всеми строками и столбцами таблицы.

Порядок выполнения работы

1. Создайте файл, как показано на рис.13.
2. Сохраните файл с именем Работа 5.HTM
3. Результат представлен на рис. 14

Контрольные вопросы

1. Что нужно сделать для того, чтобы сетка таблицы была не видна?
2. Как можно выровнять содержимое ячеек?
3. С помощью каких атрибутов можно выровнять натуральные числа по разделителю дробной части и как это сделать?
4. Что нужно сделать, чтобы включить в ячейки таблицы больше свободного пространства?
5. Можно ли к таблице добавить фон, цвет которого отличается от цвета остальной части Web-страницы?
6. Как разместить подпись слева от таблицы?

```

<HTML>
<HEAD>
<TITLE>Пример 5</TITLE>
</HEAD>
<BODY BGCOLOR="ffff00" TEXT="#000000">
<H2><U>Выставка собак <Q>"Управдом - друг
человека"</Q></U><SUB>(шутка)</SUB>
</H2>
<TABLE BORDER=3 WIDTH="70%" ALIGN=CENTER>
<CAPTION ALIGN=TOP>Таблица 1</CAPTION>
<TR BGCOLOR="00ff00">
<TH ALIGN=CENTER COLSPAN=3>Собаки
</TR>
<TR BGCOLOR="00ff00">
<TH ALIGN=CENTER>служебные
<TH ALIGN=CENTER>охотничьи
<TH ALIGN=CENTER>декоративные
</TR>
<TR>
<TD ALIGN=CENTER>немецкая<BR>овчарка
<TD ALIGN=RIGHT>такса
<TD ALIGN=RIGHT>тойтерьер
</TR>
<TR>
<TD ALIGN=CENTER>доберман-<BR>пинчер
<TD ALIGN=RIGHT>спаниель
<TD ALIGN=RIGHT>папильон
</TR>
<TR>
<TD ALIGN=CENTER>кавказская<BR>овчарка
<TD ALIGN=RIGHT>пойнтер
<TD ALIGN=RIGHT>болонка
</TR>
</TABLE>
</BODY>
</HTML>

```

Рис. 13

Выставка собак «"Управдом - друг человека"»(шутка)

Таблица 1

Собаки		
служебные	охотничьи	декоративные
немецкая овчарка	такса	тойтерьер
доберман-пинчер	спаниель	папильон
кавказская овчарка	пойнтер	болонка

Рис. 14

Практическая работа 6. Карты ссылок

Цель работы: создание карт ссылок как способ связи активных зон изображения с другими Web-страницами или файлами.

Методические указания

Карты ссылок состоят из фрагментов изображения, которые являются ссылками на другие URL. После щелчка на активной зоне открывается документ, связанный с этой зоной. Существуют два вида карт ссылок: обрабатываемые клиентом и обрабатываемые сервером. Эти термины говорят о месте расположения информации, связывающей активные зоны изображения и гипертекстовые URL. Для карт ссылок, обрабатываемых клиентом, эта информация хранится в HTML-файле, содержащем код Web-страницы, на которой находится данное изображение. Для карт ссылок, обрабатываемых сервером, указанная информация хранится в файле на Web-сервере.

Выбор изображения для карты ссылок

- Изображение должно иметь четкую структуру.
- Изображение должно состоять из областей, окрашенных разными цветами.

- Размер файла не должен быть слишком большим.
- Можно использовать единую карту ссылок, состоящую из нескольких областей, или поделить ее на две или более карты.

Создание кода карт ссылок, обрабатываемых клиентом

В коде карты ссылок, обрабатываемой клиентом, нужно определить координаты всех активных зон, а также URL-адреса связанных с ними HTML-файлов. HTML-код карты ссылок состоит из следующих основных частей (рис. 15).

Спецификация изображения:

```
<IMG SRC="имя_файла_изображения"
USEMAP="#имя_карты">.
```

Объявление карты:

```
<MAP NAME="имя_карты">.
```

Описание форм активных зон, координаты и URL-адреса:

```
<AREA SHAPE="тип_формы" COORDS="число 1, число 2, ...,
число N" HREF="URL" > </MAP> .
```

Перечислим HTML-теги и их атрибуты, которые специально предназначены для описания карт ссылок, обрабатываемых клиентом:

- USEMAP - атрибут тега .Его значением является имя, которое вы присваиваете карте ссылок. Перед именем используется символ #.
- <MAP> указывает на начало определения всех активных зон карты ссылок. Атрибут NAME - это привязка к имени карты ссылок в теге .
- Координаты и URL-адреса, связанные с активными зонами карты ссылок. У тега <AREA> есть атрибуты SHAPE, ALT, COORDS, HREF. Атрибут SHAPE может принимать значения RECT (прямоугольник), CIRCLE (окружность), POLY (многоугольник).

Карты ссылок можно кодировать вручную с помощью любого текстового или HTML-редактора. Чтобы закодировать все координаты активных зон, нужно сначала их определить. Графический редактор поможет определить координаты всех углов каждой активной зоны. Затем эти значения необходимо перечислить в HTML-файле в правильном порядке. Это очень сложная задача и лучше ею не заниматься. Вместо того, чтобы кодировать карты

ссылок вручную, можно воспользоваться специальными средствами их создания, например, LiveImage, которое является одним из самых популярных инструментов создания карт ссылок.



Рис. 15

Порядок выполнения работы

1. Создайте файл, как показано на рис 16.
2. Изображения найти в Интернете или скопируйте из пособия в свою папку с работами.
3. Сохраните файл с именем Работа 6.

Синий, желтый, розовый, голубой, зеленый треугольники и центральная фигура соответствуют изображениям показанные ниже:

```

<HTML>
<HEAD>
<TITLE>Пример 6</TITLE>
</HEAD>
<BODY BGCOLOR="ffff00" TEXT="#000000">
  <H2 ALIGN=CENTER>Выставка собак-<Q>"Управдом-друг
человека."</Q></H2>
  <IMG SRC="star.gif" HEIGHT=205 WIDTH=164 ALIGN=LEFT VSPACE=4
HSPACE=4
  USEMAP="#star">
    На этой карте представлена схема расположения площадок показа собак элитных
    пород. Для того, чтобы узнать, где какие собаки выставляются,
    пощелкайте на
    карте ссылок.<BR>
  <MAP NAME="star">
    <AREA SHAPE=POLY COORDS="31,149,50,98,9,54,70,88,101,27,97,78,143,88,
    93,102,93,180,69,121"
    HREF="dob.gif">
    <AREA SHAPE=POLY COORDS="11,40,89,23,67,72" HREF="mos.gif">
    <AREA SHAPE=POLY COORDS="113,20,151,82,108,71" HREF="pap.gif">
    <AREA SHAPE=POLY COORDS="103,111,153,95,100,185" HREF="tax.gif">
    <AREA SHAPE=POLY COORDS="63,134,87,194,26,160" HREF="toi.gif">
    <AREA SHAPE=POLY COORDS="10,73,40,100,18,156" HREF="leg.gif">
  </MAP>
</BODY>
</HTML>

```

Рис 16



Результат представлен на рис.

17

Выставка собак-«"Управдом-друг человека."»



На этой карте представлена схема расположения площадок показа собак элитных пород. Для того, чтобы узнать, где какие собаки выставляются, пощелкайте на карте ссылок.

Рис. 17

Контрольные вопросы

1. Какие виды карт ссылок вы знаете?
2. Каким должно быть изображение, используемое для карты ссылок?
3. Какие значения может принимать атрибут SHAPE?
4. Назовите основные части HTML-кода карты ссылок.

Практическая работа 7. Графика в Web

Цель работы: изучение способов размещения графических объектов на разрабатываемой Web-странице, создание изображений-ссылок.

Методические указания

Использование тега для вставки графических изображений

Чтобы вставить графический файл на Web-страницу, нужно включить гипертекстовую ссылку на него (так же, как при создании гиперссылки на другую страницу). Причем изображение, на которое вы ссылаетесь, может находиться как на том же сервере, на котором расположена ваша страница, так и в любом другом месте Internet.

Для создания URL изображения используется элемент IMG. Он является для браузера сигналом к тому, что на данное место нужно вставить изображение. < IMG> - это одиночный тег, имеющий следующий синтаксис:

< IMG SRC=URL_изображения>.

SRC обозначает место, где сохраняется изображение. Изображение может находиться на вашем компьютере, на вашем Web-сервере, на любом Web-сервере в Internet. Вместо параметра *URL_изображения* нужно указать реальный адрес файла изображения.

Вместо *URL_изображения* может стоять либо полный URL изображения (например: <http://www.greatstar.com/lois/picture.gif>), либо ссылка на относительный URL графического файла, указывающий на положение файла относительно каталога, в котором находится Web-страница.

Образец HTML-кода процедуры добавления к странице встроенного изображения:

Это фотография моей собаки .

Можно указать и полный URL, например:

< IMG SRC="http://www.someserver.com/lois/моя_собака.gif">.

Если изображение находится, например, в подкаталоге *графика*, то это будет выглядеть так:

.

Выравнивание текста относительно изображений

Web-браузеры ничего не делают для того, чтобы равномерно распределить на странице текст и графику. Поэтому существуют средства контроля за положением изображений относительно текста (рис.18)

Поля вокруг изображений

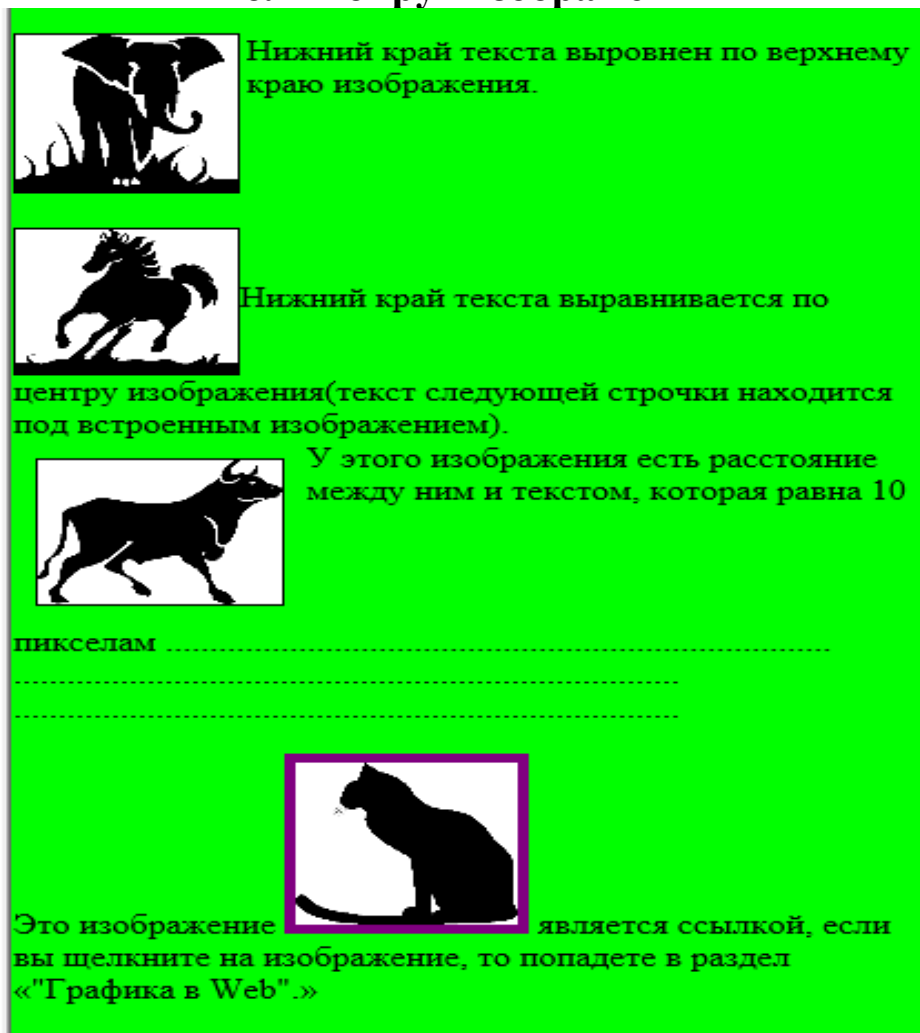


Рис. 18

Стандартные значения атрибута ALIGN:

- TOP - нижний край текста выравнивается по верхнему краю изображения

- **MIDDLE** - нижний край текста выравнивается по центру изображения

- **BOTTOM** - нижний край текста выравнивается по нижнему краю изображения

Для атрибута **ALIGN** используется следующий синтаксис:

`<IMG ALIGN=значение SRC=URL_изображения>`.

По умолчанию в теге `<IMG>` атрибуту **ALIGN** присваивается значение **BOTTOM**.

Создание плавающих изображений

С помощью значений **LEFT** и **RIGHT** атрибута **ALIGN** можно определить, вдоль какого поля должно "плавать" изображение. Например:

`<IMG SRC="моя_собака.gif" ALIGN=LEFT>`.

Для определения расстояния между текстом и плавающими изображениями, а также между краем окна и изображениями используются атрибуты **VSPACE** и **HSPACE**. Атрибут **VSPACE** служит для определения пустого пространства над и под плавающим изображением, а атрибут **HSPACE** - для определения пустого пространства справа и слева от плавающего изображения. Величины **VSPACE** и **HSPACE** измеряются в пикселях в теге `<IMG>`.

Определение ширины и высоты изображения

В HTML высоту и ширину изображения можно определить с помощью атрибутов **HEIGHT** и **WIDTH** тега `<IMG>`.

Во время загрузки Web-страницы необходимо ждать, пока загрузятся изображения, чтобы затем получить возможность прочитать текст, т.к. при обращении к Web-странице браузер сначала загружает все встроенные изображения и только потом - окружающий их текст. Если сообщить браузеру размер в пикселях изображений, содержащихся на вашей Web-странице, то он сможет подготовить макет страницы и разместить текст до того, как будет окончена загрузка всех изображений. Посетитель страницы сможет щелкнуть на какой-либо текстовой ссылке еще до окончания загрузки изображений. Например:

```
< IMG SRC="моя_собака.gif" HEIGHT=400 WIDTH=450  
ALIGN=LEFT>
```

Обтекание изображений текстом

Для определения способа обтекания текста вокруг изображения используются тег
 и атрибут CLEAR.

Для того чтобы удалить текст, находящийся справа от изображения, и поместить его под изображением, необходимо записать:

```
< BR CLEAR=LEFT>..
```

Для удаления текста, находящегося слева от изображения, и размещения под изображением, воспользуйтесь записью:

```
< BR CLEAR=RIGHT>.
```

Чтобы удалить весь текст, расположенный слева и справа от изображения, и разместить его внизу, необходимо записать:

```
< BR CLEAR=ALL>.
```

Типичный HTML-код изображения-ссылки выглядит примерно так:

```
< A HREF="URL"><IMG SRC="имя_файла_изображения"> </A>.
```

В данном случае пользователь щелкает не на строке, а на изображении.

Еще один полезный атрибут тега , особенно для изображений-ссылок - это BORDER. По умолчанию изображения-ссылки заключаются в рамку шириной 2 пикселя. Если увеличить рамку, то станет более очевидно, что изображение является ссылкой, хотя BORDER можно использовать и для обычных изображений. Пример использования атрибута BORDER в теге :

```
<IMG SRC="моя_собака.gif" BORDER=4>
```

Если присвоить значение 0, то рамка вокруг изображения будет отсутствовать.

Альтернативы изображениям-ссылкам

Обычно пользователи, имеющие "медленное" подключение к Internet, отключают в браузере функцию загрузки изображений. Существует простое решение данной проблемы с помощью атрибута ALT элемента IMG. Атрибут ALT позволяет определить текстовую

строку, которая служит альтернативой для браузеров, не поддерживающих графику. Например:

```
<IMG SRC="моя_собака.gif" ALT="фотография собаки">
```

Использование графики, цвета и текста

Добавление на Web-страницы цветных фоновых изображений

Если вы хотите создать Web-страницу, производящую на зрителя хорошее впечатление, то вам не обойтись без цветных фоновых изображений (рис. 19). Сначала надо сделать фоновое изображение цветным. С помощью атрибута BGCOLOR тега <BODY> можно определить цвет фона, указав шестнадцатеричное число, состоящее из шести цифр. Этот атрибут имеет следующий формат:

```
< BODY BGCOLOR=#rrggbb>
```

.....

```
< /BODY>.
```

Запись *#rrggbb* состоит из трех двухзначных шестнадцатеричных чисел, определяющих соотношения красного, зеленого и голубого цветов.

Добавление на Web-страницы графических фоновых изображений

Многие разработчики Web-страниц любят создавать единообразный фон для всех своих Web-страниц. В HTML для создания графического фона используется атрибут BACKGROUND тега <BODY>, например:

```
< BODY BACKGROUND="bkgrnd.gif">.
```

Для получения эффекта браузер "выкладывает" изображение мозаикой, заполняя пространство фона.

Изменение цвета текста и гиперссылок

Цвет основного текста можно изменить с помощью атрибута TEXT тега <BODY> (рис. 19). При этом используется следующий формат:

```
< BODY TEXT="#rrggb">
```

.....

```
< /BODY>.
```

Изменить цвета гиперссылок можно с помощью следующих атрибутов: LINK, VLINK, ALINK. Эти атрибуты устанавливают цвета для непосещенных, посещенных и активных ссылок соответственно. Для изменения цвета ссылок используется следующий формат:

```
< BODY LINK="#rrggb" VLINK="#rrggb" ALINK="#rrggb">
```

.....

```
< /BODY>.
```

По умолчанию для LINK выбирается голубой цвет, для VLINK - фиолетовый, для ALINK - красный.



Рис. 19

Изменение внешнего вида текста

С помощью контейнера и его атрибута COLOR можно изменить цвет любого символа или группы символов. Например:

```
< FONT COLOR=RED> Этот текст красного цвета</FONT>.
```

Значением атрибута COLOR может быть и шестнадцатеричное число(#rrggbb). Тег можно использовать также для изменения шрифта, отображаемого браузером. Для этого используется атрибут FACE, за которым следует список имен шрифтов, с помощью которых браузер должен отображать приведенный далее текст. Необходимо точно указывать имя шрифта. Пример:

< FONT FACE="Arial".....текст......

Порядок выполнения работы

1. найти в Интернете необходимые картинки и скопируйте их в свою папку с работами.
2. Откройте текстовый редактор Блокнот
3. Введите программу для создания Web-страницы

```

<HTML>
<HEAD>
<TITLE>Пример 7</TITLE>
</HEAD>
<BODY BGCOLOR="ffff00" TEXT="#000000">
<H2 ALIGN=CENTER><U>Выставка собак</Q>"Управдом-друг
человека"</Q></U>
<SUB>(шутка)</SUB></H2>
На выставке представлены служебные, охотничьи и декоративные
<IMG SRC="mos.gif" HEIGHT=100 WIDTH=140 ALIGN=LEFT
BORDER=1> собаки. &nbsp; Служебные собаки - это: овчарка,
доберман-пинчер, кавказская овчарка, южно-русская овчарка,
московская сторожевая. <BR CLEAR=LEFT><BR>
<IMG SRC="tax.gif" HEIGHT=100 WIDTH=140
ALIGN=MIDDLE BORDER=1 ALT="такса"> Охотничьи собаки: такса,
спаниель, короткошерстная немецкая легавая,
пойнтер, русская пегая гончая, западносибирская лайка. <BR
CLEAR=LEFT><BR>
<IMG SRC="pap.gif" HEIGHT=100 WIDTH=140 ALIGN=LEFT
VSPACE=20 HSPACE=20 BORDER=1> Декоративные собаки: тойтерьер,
папильон,
болонка.....
.....
.....
<BR CLEAR=LEFT><P>
Это изображение <a HREF="lab7.htm"><IMG SRC="cat.gif"
HEIGHT=120 WIDTH=170 BORDER=5></a> является ссылкой, если вы
щелкните на
изображение, то попадете в <Q>"Лабораторную работу
7"</Q>. <BR><BR><BR>
</BODY>
</HTML>

```

4. Результат показан на рис 20.
5. Сохраните файл с именем Работа 7.HTML

Выставка собак-«"Управдом-друг человека"» (шутка)



На выставке представлены служебные, охотничьи и декоративные собаки. Служебные собаки - это: овчарка, доберман-пинчер, кавказская овчарка, южно-русская овчарка, московская сторожевая.



Охотничьи собаки: такса, спаниель, короткошерстная немецкая легавая, пойнтер, русская пегая гончая, западносибирская лайка.



Декоративные собаки: тойтерьер, папильон, болонка



Это изображение

является ссылкой, если вы щелкните на изображение, то попадете в «"Лабораторную работу 7"».

Рис. 20

Контрольные вопросы

1. Чем отличаются встроенные изображения от плавающих?
2. Приведите синтаксис процедуры добавления к странице встроенного изображения.
3. Как определить расстояние между текстом и плавающим изображением?
4. Как сделать так, чтобы браузер сперва загружал весь текст, а затем уже все встроенные изображения?
5. Что такое изображение-ссылка?

Практическая работа 8. Фреймы

Цель работы: изучение создания и использования фреймов для оформления Web-страниц.

Методические указания

Создание фреймов

Фреймы - это еще один способ создания уникального интерфейса Web-сервера. Разделив страницу на части, каждая из которых обновляется отдельно, можно создавать для них различные элементы интерфейса. Чтобы получить страницу, содержащую фреймы, нужно сначала создать несколько документов (рис. 20): те, которые будут помещаться в фреймы и документ, определяющий общую структуру страницы.

Для создания структурного документа используется контейнер FRAMESET. Пара тегов <FRAMESET> </FRAMESET> полностью заменяет в структурном документе пару <BODY> </BODY>. В теге <FRAMESET> используются два атрибута: ROWS - страница будет разделена по горизонтали; COLS - страница будет разделена по вертикали. Для этих атрибутов используется следующий формат:

< FRAMESET ROWS="числа, проценты,* ">;

< FRAMESET COLS="числа, проценты,* ">.

Это означает, что необходимо выбрать значения в пикселях, процентах или просто *. Пример: создаются два горизонтальных фрейма - один имеет длину 50 пиксель, а другой - всю остальную часть страницы:

< FRAMESET ROWS="50,* ">.

В следующем примере создается страница, разделенная на два столбца, где левый занимает 25% экрана, а правый - остальные 75%:

< FRAMESET COLS="25%, 75%">.

В отдельном теге <FRAMESET> может использоваться только один из атрибутов - ROWS или COLS. Это означает, что фреймовая структура может состоять или только из строк, или только из столбцов. Чтобы создать строки внутри столбцов или наоборот, необходимы вложенные теги <FRAMESET>. Например:

< FRAMESET COLS="25%, 75%">

```
< FRAMESET ROWS="50%, 50%">  
< /FRAMESET>  
< FRAMESET ROWS="10%, 90%">  
< /FRAMESET>  
< /FRAMESET>.
```

Сначала страница была разделена на два столбца (левый - 25%, правый - 75%). Затем первый столбец был разделен на две строки по 50% пространства столбца. Второй столбец был разделен на две строки (первая - 10%, вторая - 90%).

Как загружать в фреймы целые страницы

С помощью тега `<FRAME>`, помещенного в контейнер `FRAMESET`, можно определить, что именно должно отображаться в конкретном фрейме. С его помощью можно указать URL или имя, связанное с конкретным фреймом. Для этого используется следующий формат:

```
< FRAMESET COLS/ROWS="числа">  
< FRAME SRC="URL">  
  
.....  
< /FRAMESET>.
```

Атрибут `SRC` используется для указания URL страницы, которая должна быть загружена в фрейм. При использовании в теге `<IMG>` атрибуту `SRC` может быть присвоен относительный либо абсолютный URL. Можно использовать абсолютные или относительные URL адреса по своему усмотрению. Каждый тег `<FRAME>` создает отдельное окно фрейма. Эти окна соответствуют определениям строк или столбцов в теге `<FRAMESET>`, но на экране никакие разделительные линии не появятся до тех пор, пока каждое окно не будет определено с помощью тега `<FRAME>`.

Настройка фреймов

Атрибуты `MARGINWIDTH` и `MARGINHEIGHT` используются для определения правого/левого и верхнего/нижнего полей текста и графики внутри фрейма соответственно (рис. 21). Оба эти атрибута принимают числовые значения в пикселях. В следующем коде между

содержимым файла text.html и границами фрейма определяется поле шириной 5 пиксель:

```
<FRAME SRC="text.html" MARGINWIDTH=5 MARGINHEIGHT=5>.
```

Атрибут SCROLLING может принимать значения yes, no, auto. Он позволяет определить, присутствуют ли в окне фрейма полосы прокрутки. По умолчанию принимается значение auto.

Атрибут NORESIZE не принимает никаких значений, он используется для того, чтобы не позволить пользователю изменить размеры окна фрейма. Пример использования атрибутов SCROLLING и NORESIZE:

```
< FRAME SRC="text.html" SCROLLING="no" NORESIZE>.
```

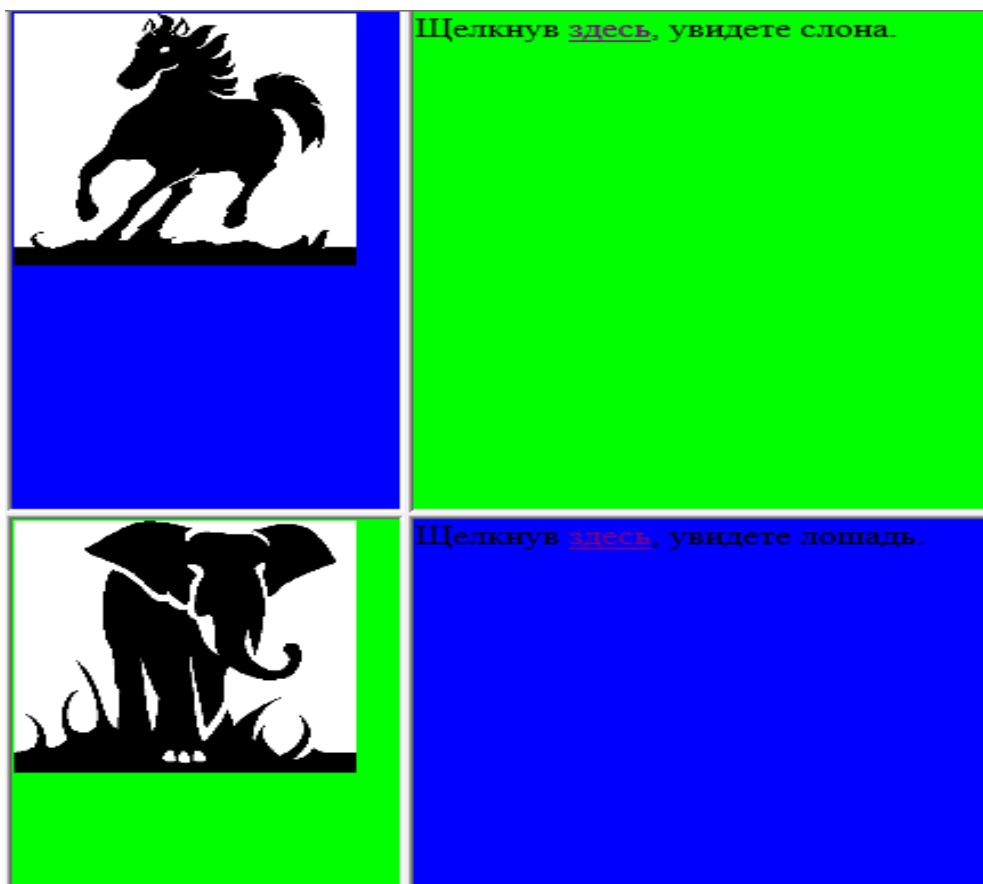


Рис. 21

Определение границ фрейма

Атрибут FRAMEBORDER можно использовать в теге <FRAMESET> либо <FRAME>. Он может принимать значения «yes»

(по умолчанию) и «no» - означает отсутствие границ. С помощью атрибута BORDER можно определить ширину рамки в пикселях. Если установить значения FRAMEBORDER="no", а BORDER="0", то границы фреймов на вашей странице будут невидимыми.

В тегах <FRAMESET> и <FRAME> можно использовать атрибут BORDERCOLOR. С его помощью определяется цвет рамок (т.е. границ фреймов). По умолчанию границы фреймов имеют тускло-серый оттенок. При определении цвета можно использовать как его название, так и числовой эквивалент.

Проблемы браузеров, не поддерживающих фреймы

Контейнер NOFRAMES предназначен для хранения HTML-кода, адресованного тем браузерам, которые не поддерживают фреймы. Браузеры, поддерживающие фреймы, игнорируют все, что содержится внутри пары тегов <NOFRAMES> </NOFRAMES>. Остальные браузеры воспринимают только текст, заключенный в контейнер NOFRAMES. Например:

<NOFRAMES> для просмотра этой страницы необходим браузер, поддерживающий фреймы </NOFRAMES>

Как присваивать фреймам имена

Чтобы контролировать процесс загрузки фреймов, нужно присвоить окнам фреймов имена. Для этого предназначен атрибут NAME, который используется в теге <FRAME>:

< FRAME SRC="URL" NAME="имя_окна">.

Если окно фрейма имеет уникальное имя, то к нему можно непосредственно обратиться из других окон фреймов:

< FRAME SRC="index.html" NAME="основное_окно">.

Загрузка страниц в именованные окна фреймов

Присвоив имена фреймам, можно указать их в гипертекстовой ссылке с помощью атрибута TARGET, который помещается в обычный тег привязки <A>. При этом используется следующий формат:

< A HREF="новый_URL" TARGET="имя_окна" текст гиперссылки.

Здесь:

- *новый_URL* - это адрес нового документа, который надо загрузить в определенное окно фрейма;
- *имя_окна* - это имя, которое вы присвоили окну с помощью атрибута NAME в теге <FRAME>.

Пример: гиперссылка, в результате щелчка на которой файл text.html загрузится в окно *Основное_окно*:

< A HREF="text.html" TARGET="Основное_окно">Текст.

Использование тега <BASE> для определения стандартных параметров

Если вы хотите, чтобы большинство ссылок на вашей Web-странице указывало на определенное окно фрейма, то для этого поместите в элемент HEAD HTML-кода страницы тег <BASE> следующим образом:

< HEAD>

< BASE TARGET="имя_окна">

< /HEAD>

И тогда не нужно указывать атрибут TARGET в каждой паре тегов <A>< /A>.

Порядок выполнения работы

Создайте файл, как показано на рис 22.

Сохраните файл с именем Работа 8.

Контрольные вопросы

1. Для чего предназначены фреймы?
2. Как создать фрейм?
3. Как упростить создание фреймов, если большинство ссылок на вашей Web-странице указывает на определенное окно фрейма?
4. Можно ли при создании фреймов во фрейме присваивать окнам фреймов одинаковые имена?
5. О чем говорит * в атрибутах ROWS или COLS?
6. Что нужно сделать, чтобы не позволить пользователю изменить размеры окна фрейма?

Охотничьи собаки 1. Спаниель 2. <u>Такса</u> 3. Пойнтер 4. Русская пегая гончая 5. Западно - сибирская лайка 6. <u>Коротко-шерстная немецкая легавая</u>	Служебные собаки 1. Немецкая овчарка 2. <u>Доберман-пинчер</u> 3. Кавказская овчарка 4. Сенбернар 5. Южно-русская овчарка 6. <u>Московская сторожевая</u>	Декоративные собаки 1. Болонка 2. <u>Папильон</u> 3. <u>Тойтерьер</u>
Выставка собак - «"Управдом - друг человека"».		

Рис. 22

СПРАВОЧНИК ПО HTML

Список тегов и атрибутов

Тег	Раздел	Подраздел
<A>	Гипертекстовые ссылки	Создание гиперссылок на другие Web-страницы
<ALIGN>	1.Основная часть Web-страницы 2.Графика в Web 3.Таблицы	1.Выравнивание абзацев и заголовков. Создание линий. 2.Выравнивание текста относительно изображений 3.Выравнивание данных в ячейках таблицы по горизонтали
ALINK	Использование графики, цвета, текста	Изменение цвета текста и гиперссылок
<ALT>	Графика в Web	Альтернативы изображениям-ссылкам
<AREA>	Карты ссылок	
	Основная часть Web-страницы	Форматирование текста с помощью стилей
BACKGROUND	1.Использование графики, цвета, текста 2.Листы стилей	1.Добавление на Web-страницы графических фоновых изображений 2.Возможности листов стилей
<BASE>	Фреймы	Использование тега <BASE> для определения стандартных параметров
BGCOLOR	Использование графики, цвета, текста	Добавление на Web-страницы цветных фоновых изображений
<BLOCKQUOTE>	Основная часть Web-страницы	Форматирование цитат с помощью тегов

		<BLOCKQUOTE> и <Q>
<BODY>	Создание Web-страницы	
BORDER	Графика в Web	Обтекание изображений текстом
BORDERCOLOR	Таблицы	Окрашивание таблиц и рамок
BORDERLIGHT	Таблицы	Окрашивание таблиц и рамок
 	Основная часть Web-страницы	Использование символов разрыва абзаца и строк
<CAPTION>	Таблицы	Добавление подписей к таблицам
CELLPADDING	Таблицы	Выбор типа таблицы
CELLSPACING	Таблицы	Выбор типа таблицы
<CITE>	Основная часть Web-страницы	Форматирование текста с помощью стилей
CLASS	Листы стилей	Использование тега <STYLE> и атрибута CLASS
CLEAR	Графика в Web	Обтекание изображений текстом
<CODE>	Основная часть Web-страницы	Форматирование текста с помощью стилей
COLOR	Использование графики, цвета, текста	Изменение внешнего вида текста
COLS	1.Таблицы 2.Фреймы	1.Выбор типа таблицы 2.Создание фреймов
COLSPAN	Таблицы	Увеличение размеров ячеек с помощью атрибутов COLSPAN и ROWSPAN
COORDS	Карты ссылок	
<DFN>	Основная часть	Форматирование текста с

	Web-страницы	помощью стилей
<DIR>	Использование новых атрибутов	Определение направления прочтения текста с помощью атрибута DIR
<DIV>	Листы стилей	Использование тега <DIV> и листов стилей
<DD>	Создание списков	Списки определений
<DL>	Создание списков	Списки определений
<DT>	Создание списков	Списки определений
	Основная часть Web-страницы	Форматирование текста с помощью стилей
FACE	Использование графики, цвета, текста	Изменение внешнего вида текста
	Основная часть Web-страницы	Изменение размера шрифта
FRAME	Фреймы	Как загружать в фреймы целые страницы
FRAMEBORDER	Фреймы	Определение границ фрейма
FRAMES	Таблицы	Использование атрибутов таблиц
FRAMESET	Фреймы	Создание фреймов
<HEAD>	Создание Web-страницы	
HEIGHT	Графика в Web	Определение ширины и высоты изображения
<Hn>	Основная часть Web-страницы	Использование заголовков для создания структуры документа
HREF	Гипертекстовые	Создание гиперссылок на

	ссылки	другие Web-страницы
HSPACE	Графика в Web	Поля вокруг изображений
<HTML>	Создание Web-страницы	
<I>	Основная часть Web-страницы	Форматирование текста с помощью стилей
	Графика в Web	Использование тега для вставки графических файлов
<KBD>	Основная часть Web-страницы	Форматирование текста с помощью стилей
<LANG>	Использование новых атрибутов	Определение языка с помощью атрибута LANG
	Создание списков	Нумерованные списки
LINK	1.Использование графики, цвета, текста 2.Листы стилей	1.Изменение цвета текста и гиперссылок 2. Использование тега <LINK> в листах стилей
<MAP>	Карты ссылок	
MARGINHEIGHT	Фреймы	Настройка фреймов
MARGINWIDTH	Фреймы	Настройка фреймов
NAME	1.Гипертекстовые ссылки 2.Карты ссылок 3.Фреймы	1.Создание внутренних ссылок в документах 2. 3.Как присваивать фреймам имена
NOFRAMES	Фреймы	Проблемы браузеров, не поддерживающих фреймы
NORSIZE	Фреймы	Настройка фреймов
NOSHADE	Основная часть Web-страницы	Создание линий в HTML

	Создание списков	Нумерованные списки
<P>	Основная часть Web-страницы	Использование символов разрыва абзаца и строк
<PRE>	Основная часть Web-страницы	Использование элемента PRE для упрощения процедуры форматирования текста
<Q>	Основная часть Web-страницы	Форматирование цитат с помощью тегов <BLOCKQUOTE> и <Q>
ROWS	Фреймы	Создание фреймов
ROWSPAN	Таблицы	Увеличение размеров ячеек с помощью атрибутов COLSPAN и ROWSPAN
RULES	Таблицы	Использование атрибутов таблиц
<SAMP>	Основная часть Web-страницы	Форматирование текста с помощью стилей
SCROLLING	Фреймы	Настройка фреймов
SHAPE	Карты ссылок	
SIZE	Основная часть Web-страницы	Изменение размера шрифта Создание линий в HTML
	Листы стилей	Использование тега для форматирования определенных элементов
SRC	Графика в Web	Использование тега для вставки графических файлов
<STYLE>	Листы стилей	Использование тега <STYLE> и атрибута CLASS
<STRIKE>	Основная часть Web-страницы	Форматирование текста с помощью стилей

	Основная часть Web-страницы	Форматирование текста с помощью стилей
<SUB>	Основная часть Web-страницы	Форматирование текста с помощью стилей
<SUP>	Основная часть Web-страницы	Форматирование текста с помощью стилей
<TABLE>	Таблицы	Выбор типа таблицы
TARGET	Фреймы	Загрузка страниц в именованные окна фреймов
<TD>	Таблицы	Создание строк и столбцов
TEXT	Использование графики, цвета, текста	Изменение цвета текста и гиперссылок
<TH>	Таблицы	Определение ячеек данных и заголовков
<TITLE>	Создание Web-страницы	
<TR>	Таблицы	Создание строк и столбцов
<TT>	Основная часть Web-страницы	Форматирование текста с помощью стилей
TYPE	Листы стилей	Использование тега <STYLE> и атрибута CLASS
<U>	Основная часть Web-страницы	Форматирование текста с помощью стилей
	Создание списков	Маркированные списки
USEMAP	Карты ссылок	
VALIGN	Таблицы	Выравнивание данных в ячейках таблицы по вертикали
<VAR>	Основная часть Web-страницы	Форматирование текста с помощью стилей

VLINK	Использование графики, цвета, текста	Изменение цвета текста и гиперссылок
VSPACE	Графика в Web	Поля вокруг изображений
WIDTH	1.Основная часть Web-страницы 2.Графика в Web 3.Таблицы	1.Создание линий в HTML 2.Определение ширины и высоты изображения 3.Выбор типа таблицы

Оглавление

ВВЕДЕНИЕ	3
Практическая работа 1. Создание простой Web - страницы<HR>	4
Элемент заголовка <H1> </H1>.....	6
Выравнивание абзацев и заголовков.....	6
Создание линий <HR>	8
<i>Порядок выполнения работы</i>	9
<i>Контрольные вопросы</i>	11
Практическая работа 2. Форматирование текста.....	12
Символы разрыва абзаца и строки <P>, 	12
Форматирование цитат <BLOCKQUOTE> </BLOCKQUOTE>, <Q> </Q>	13
Форматирование текста с помощью стилей.....	13
Изменение размера шрифта 	14
<i>Порядок выполнения работы</i>	15
<i>Контрольные вопросы</i>	16
Практическая работа 3. Создание списков.....	17
Типы списков	17
Нумерованные списки	17
Маркированные списки	19
Списки определений	19
Вложение и комбинирование списков.....	20
<i>Порядок выполнения работы</i>	20
<i>Контрольные вопросы</i>	20
Практическая работа 4. Гипертекстовые ссылки	23
Типы гиперссылок.....	23
Понятие об URL и их использование.....	23
Создание гиперссылок на другие Web-страницы.....	24
Создание внутренних ссылок	25
Ссылки на ресурсы мультимедиа	26
Создание ссылок на адреса электронной почты.....	26
Создание ссылки на FTP-сервер.....	27
Использование тега <BASE> для упрощения URL.....	27
<i>Порядок выполнения работы</i>	28
<i>Контрольные вопросы</i>	30
Практическая работа 5. Таблицы.....	30

Создание строк и столбцов таблицы	30
Выбор типа таблицы.....	31
Определение ячеек данных и заголовков.....	31
Увеличение размеров ячеек с помощью атрибутов COLSPAN и ROWSPAN.....	32
Выравнивание данных в ячейках таблицы по горизонтали.....	33
Выравнивание данных в ячейках по вертикали	34
Добавление подписей к таблицам.....	34
Окрашивание таблиц и рамок	34
Макетирование страниц с помощью таблиц	35
Использование атрибутов таблиц	36
<i>Порядок выполнения работы</i>	<i>37</i>
<i>Контрольные вопросы</i>	<i>37</i>
Практическая работа 6. Карты ссылок	39
Выбор изображения для карты ссылок	39
Создание кода карт ссылок, обрабатываемых клиентом	40
<i>Порядок выполнения работы</i>	<i>41</i>
<i>Контрольные вопросы</i>	<i>43</i>
Практическая работа 7. Графика в Web	44
Использование тега для вставки графических изображений.....	44
Выравнивание текста относительно изображений	45
Создание плавающих изображений.....	46
Поля вокруг изображений.....	45
Определение ширины и высоты изображения	46
Обтекание изображений текстом.....	47
Альтернативы изображениям-ссылкам.....	47
Использование графики, цвета и текста.....	48
Добавление на Web-страницы цветных фоновых изображений	48
Изменение цвета текста и гиперссылок	49
<i>Порядок выполнения работы</i>	<i>50</i>
<i>Контрольные вопросы</i>	<i>52</i>
Практическая работа 8. Фреймы.....	53
Создание фреймов	53
Как загружать в фреймы целые страницы	54
Настройка фреймов	54

Определение границ фрейма.....	55
Проблемы браузеров, не поддерживающих фреймы	56
Как присваивать фреймам имена.....	56
Загрузка страниц в именованные окна фреймов	56
Использование тега <BASE> для определения стандартных параметров	57
<i>Контрольные вопросы</i>	57
Приложение	59
СПРАВОЧНИК ПО HTML	59
Список тегов и атрибутов.....	59