

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение
высшего профессионального образования

“Оренбургский государственный университет”

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Л.Ф. ГАЙСИНА

СОЗДАНИЕ WEB-СТРАНИЦ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ

Рекомендовано к изданию Редакционно-издательским советом
государственного образовательного учреждения
высшего профессионального образования
“Оренбургский государственный университет”

Оренбург 2004

ББК 32.973.202я73

Г 14

УДК 004.7(075)

Рецензент

кандидат технических наук, доцент Раимова А.Т.

Гайсина Л.Ф.

Г - 14 **Создание Web-страниц: Методические указания к лабораторному практикуму. - Оренбург: ГОУ ОГУ, 2004. – 76 с.**

Лабораторный практикум состоит из пяти лабораторных работ и посвящен изучению основ Web-технологий. Каждая лабораторная работа представлена комплексом заданий для ее выполнения, рассмотрены теоретические вопросы, связанные с реализацией поставленных задач, контрольные вопросы, тесты для самопроверки, приведен список учебной и научной литературы.

Методические указания предназначены для выполнения лабораторного практикума по дисциплине “Сетевые информационные технологии” для студентов специальности 220400.

ББК 32.973.202я73

© Гайсина Л.Ф., 2004

© ГОУ ОГУ, 2004

Введение

Данные методические указания посвящены основам Web-технологий. В их рамках рассматриваются вопросы создания Web-страниц как с помощью языка гипертекстовой разметки HTML, так и с помощью приложения Microsoft FrontPage 2002 – автоматизированного средства создания Web-страниц. Этот профессиональный конструктор позволяет создавать сайты, а также настраивать и поддерживать их на Web-серверах. Так первая лабораторная работа полностью посвящена изучению работы приложения Microsoft FrontPage 2002, в то время как остальные посвящены изучению языка гипертекстовой разметки HTML.

Целью данных методических указаний - дать представление об основах Web-технологий и научить студентов создавать и опубликовывать в сети Internet свои собственные статические и динамические Web-сайты.

После окончания изучения представленного указания студенты будут уметь разрабатывать общую структуру сайтов, формировать страницы, из которых будет состоять структура, добавлять интерактивные средства и эффекты мультимедиа и, наконец, размещать готовые сайты в Internet путем загрузки их на Web-сервер.

1 Лабораторная работа №1

Использование Microsoft FrontPage для создания HTML-страниц

1.1 Цель работы

Ознакомление с программой Microsoft FrontPage 2002. Изучение интерфейса, основных возможностей программы. Создание при помощи данной программы своей Web-страницы. Запись созданной страницы с помощью службы FTP на сервер.

1.2 Порядок выполнения работы

1) Изучить интерфейс программы и выделить основные его элементы. Открыть какую либо готовую Web-страницу и просмотреть её содержимое в разных режимах отображения в программе. Исследовать в соответствии с методическими указаниями пункты главного меню программы.

2) Создать свою Web-страницу, как описано в методическом указании к данной работе, сохранить её в доступном каталоге на выделенном диске. Создать каталог со своим именем, или фамилией.

3) После создания и сохранения Web-страницы, добавить на неё какой либо текст, например, несколько определений из методических указаний. Отформатировать введённый текст, выбрать другой стиль шрифта, другую гарнитуру, при этом удобно пользоваться окном «Шрифт».

4) Добавить ещё несколько абзацев на страницу. Произвести форматирование абзацев, используя окно «Абзац». Преобразовать два последних абзаца в списки, используя окно форматирования списков.

5) Добавить заголовок в начало вашей страницы, отформатировать его как заголовок. Сделать ещё один подзаголовок на странице и отформатировать его, сделав более мелким.

6) Преобразовать несколько имеющихся фраз в гиперссылки, или создать несколько новых гиперссылок. Использовать пять видов гиперссылок, в том числе на электронную почту и файлы из внешнего каталога. Настроить цвета гиперссылок (открытые, неоткрытые, под курсором).

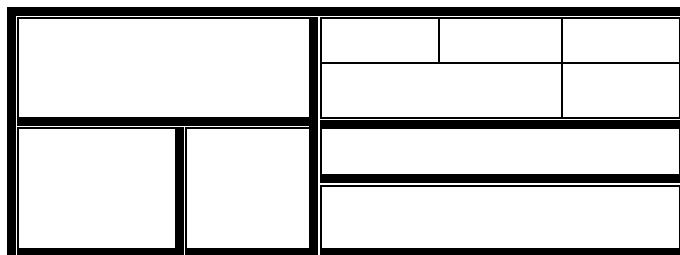
7) Использовать на странице динамические кнопки.

8) Добавить к тексту на странице несколько рисунков, видео и анимацию, фоновый звук.

9) Вставить счетчик посещений страницы и бегущую строку.

10) Использовать технологии фреймов. Разбить Web-страницу на фреймы как показано ниже.

11) Разбить один из фреймов разбить таблицей на несколько строк и столбцов, например как показано ниже.



12) Использовать формы и таблицы для организации гостевой книги, или для регистрации какой-либо информации.

13) Созданную Web-страницу загрузить при помощи службы FTP на FTP-сервер.

К сдаче лабораторной предоставляются: работающая страница на сервере с комментариями каждого тэга и его свойств (атрибутов) в исходнике.

1.3 Методические указания

Программа FrontPage 2002 предназначена для быстрого создания Web-сайтов. Ее приобретение - отдельно или в комплекте с пакетами программ Microsoft Office 2002 Professional или Developer - избавит от сложностей построения сайтов с использованием *языка HTML* (Hyper-Text Markup Language), основного языка программирования среды Web. В данной программе предстоит работать только с несложным меню и набором команд, которые схожи с другими командами из программ Microsoft Office, таких как Word, Excel и PowerPoint.

Используя FrontPage, можно создавать общую структуру Web-сайтов, формировать страницы, добавлять интерактивные средства и эффекты мультимедиа и, наконец, размещать готовые сайты в Internet, загружая файлы на Web-сервер. Программы Office распознают язык HTML так же легко, как и свои собственные форматы, поэтому с помощью FrontPage можно добавлять документы Office напрямую в новые Web-страницы.

Инструменты и функции программы FrontPage позволяют достаточно быстро построить сайт любой сложности и разместить его в сети Internet или в локальных корпоративных сетях intranet.

1.3.1 Рекомендации по планированию Web-сайта

Хотя очень много говорится о том, что сеть Web представляет собой совершенно новую среду, следует отметить, что она воспринимается

исключительно при помощи человеческого зрения. При работе с Web-сайтом надо полагаться только на визуальное восприятие.

Сайты, которые состоят из тысячи разветвлений, сразу вызывают у посетителей смущение и непонимание.

Беспорядок может быть различным. Встречаются Web-сайты, демонстрирующие неприятное явление, которое можно охарактеризовать как «ложный вход». Вместо ожидаемой домашней страницы со ссылками на продукцию или на сотрудников компании можно видеть огромный красивый рисунок или многослойную анимацию/2/.

Нельзя заставлять посетителей стучаться в дверь дважды. И уж совсем не стоит заставлять их просить, чтобы им открыли. Чтобы посетители увидели несколько рисунков, необходимо сделать эти рисунки маленькими и простыми для загрузки.

На каждой Web-странице следует особо выделять самую важную информацию. Разумеется, нужно определиться, что является важным, а что незначительным как на отдельной странице, так и на всем Web-сайте. Прежде чем начать работать над анимированными рисунками для домашней страницы, необходимо убедиться в том, какую информацию несет сайт и кто будет его посещать.

Планирование Web-сайта – пожалуй, одна из самых сложных задач. Очень нелегко правильно расположить имеющуюся информацию. Если сайт очень «глубокий», то пользователям придется открывать один уровень за другим, чтобы найти нужные сведения. Опасность заключается в том, что подобный поиск довольно часто заканчивается ничем. С другой стороны, можно создать слишком «мелкий» Web-сайт, в состав которого войдет одна домашняя страница с огромным количеством непонятных и перепутанных ссылок.

В идеале посетитель должен находить любую нужную информацию за два-три щелчка мышью. Однако этого не так просто добиться. К счастью, FrontPage позволяет перемещать страницы таким образом, что можно получить доступ к любым данным не более чем за четыре нажатия мыши.

При изучении различных средств программы FrontPage в следующих главах всегда следует помнить: необходимо убирать все лишнее, ориентироваться на низкую скорость загрузки и делать сайт простым.

1.3.2 Главное окно программы FrontPage

Главное окно программы по умолчанию открывается в обычном режиме (рисунок 1). Оно отличается от аналогичного окна в MS Office тем, что содержит вкладки выбора страниц (2), панель видов (4), кнопки режимов просмотра (5), в строке статуса индикатор процесса загрузки (6), приблизительное время загрузки (7).

Главное окно программы FrontPage включает в себя:

1) Название сайта, Web – страницы, набор вкладок страницы. Название сайта задаются при его создании. Аналогично присваивается название и странице.

2) Выбор станицы. Чтобы осуществить выбор страницы необходимо выделить страницу, чтоб отобразить её в главном окне.

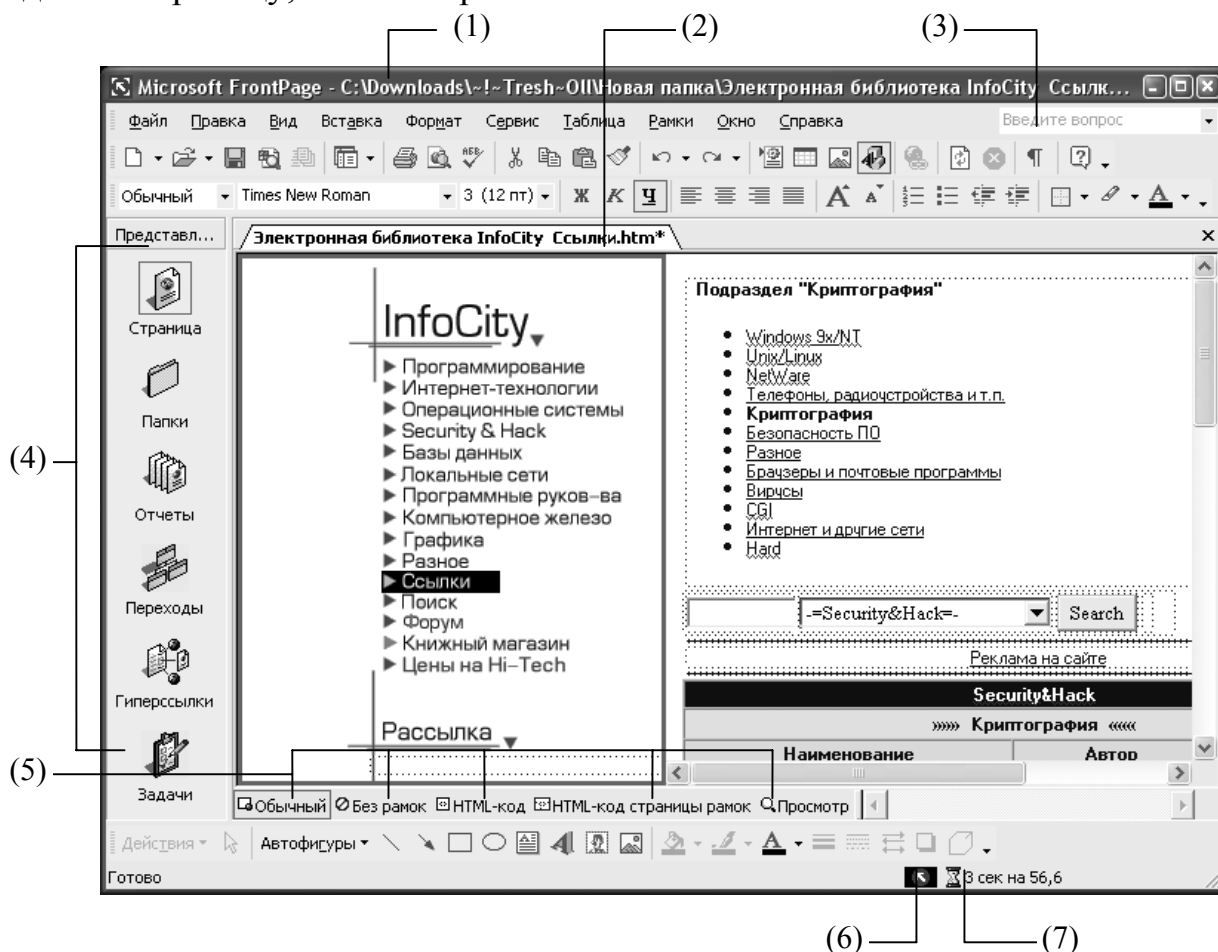


Рисунок 1 – Главное окно программы FrontPage

3) Строка поиска раздела справки FrontPage. Чтобы найти справку по любой теме необходимо ввести в строку поиска вопрос или ключевое слово и нажать на клавишу Enter, появится список ответов и тем. Необходимо щелкнуть по выбранной теме, чтобы получить по ней подробную информацию.

4) Панель видов. По умолчанию FrontPage открывается в режиме просмотра страницы. Чтобы переключиться в другой режим необходимо нажать на одну из оставшихся пяти кнопок. Режим просмотра папок "Папки" (Folders) используется для работы с файлами и папками, режим просмотра сообщений "Отчеты" (Reports) - для доступа к файлам, ссылкам и рисункам открытого Web-сайта, режим навигации "Переходы" (Navigation) - для просмотра и редактирования структуры сайтов, режим просмотра гиперссылок "Гиперссылки" (Hyperlinks) – для просмотра ссылок на любые Web-страницы, режим просмотра задач "Задачи" (Tasks) - для просмотра, создания и редактирования списка задач для сайта.

5) Режимы просмотра страниц. По умолчанию FrontPage открывается в обычном режиме, в котором можно редактировать открытую Web-страницу. Для просмотра или редактирования кодировки страницы необходимо нажать на кнопку HTML (Просмотр HTML-кодировки). Чтобы увидеть, как страница будет выглядеть в Web-браузере - нажать на кнопку “Предварительный просмотр” (Preview). Не обязательно при этом запускать Internet Explorer. Если на компьютере не установлена эта программа, то кнопка Preview будет скрытой.

6) Индикатор процесса загрузки. Эта кнопка активизируется, когда FrontPage открывает какую либо Web-страницу. После открытия всех файлов Web-страницы данная функция отключается.

7) Приблизительное время загрузки. Здесь показывается, сколько секунд будет загружаться текущая страница при скорости модемного соединения, равной 28.8 bps. Можно изменить последнюю величину, хотя заданная по умолчанию скорость соответствует средней скорости работы модемного соединения.

1.3.3 Главное меню программы FrontPage

Меню, которые расположены в верхней части главного окна FrontPage, остаются неизменными независимо от любых действий. Однако некоторые меню, например “Frames” (Фреймы), могут использоваться только при условии, что происходит работа с Web-страницами, на которых есть фреймы (рисунок 2).

Меню Файл (File) - при помощи команд в меню “Файл” можно выполнять различные действия: создавать, открывать, закрывать, сохранять, размещать и импортировать Web-страницы или сайты. Меню File также позволяет задавать настройки печати, изменять общие установки программы FrontPage и выходить из программы.

Меню Правка (Edit) - большинство команд в меню “Правка” работают так же, как в других программах, за исключением пятого раздела меню. В нем можно задавать параметры проверки Check In/Check Out system, которые позволят избежать конфликтов версий файлов в том случае, если с Web-сайтом работает несколько пользователей.

Меню Вид (View) - большинство команд в меню “Вид” повторяют те команды, которые выполняются при помощи кнопок на панели видов. Пункт меню Toolbars (Панели инструментов) позволяет включить или выключить любую из 12 панелей инструментов FrontPage.

Меню Вставка (Insert) - меню предназначено для вставки в Web-страницы любой информации от разрывов строк, дат и гиперссылок до команд Java или компонентов Office XP или Office 2000, например диаграмм и таблиц.

Меню Формат (Format) - при помощи команд в меню “Формат” можно изменять внешний вид текста, адреса HTML, а также так называемые темы FrontPage, определяющие общий дизайн вашего Web-сайта.

Меню Сервис (Tools) – в меню “Сервис” позволяет изменять многие из настроек программы FrontPage. Оно также предоставляет доступ к словарю и тезаурусу программы FrontPage.

Меню Таблица (Table) - в меню “Таблица” находятся команды, необходимые для добавления таблиц в страницу или изменения расположения строк или столбцов.

Меню Рамки (Frames) – позволяет работать с командами в меню “Рамки” можно только в том случае, если открыта Web-страница, в состав которой входят HTML-фреймы. Страницы с фреймами создаются при помощи команды “Создать страницу” (New Page) или клавиш Ctrl+N.

Меню Окно (Window) - программа FrontPage позволяет одновременно открывать несколько Web-страниц и сайтов. Меню Window служит для быстрого переключения от одного окна к другому. Также можно это делать при помощи вкладок страниц в главном окне FrontPage.

Меню Справка (Help) - это меню имеется в любой программе Windows. Оно предоставляет доступ к системе справочной информации программы FrontPage.

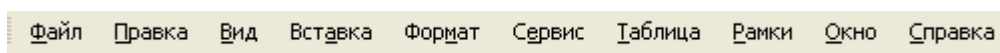


Рисунок 2 – Главное меню программы FrontPage

1.3.4 Создание, открытие, сохранение и закрытие Web-страницы

Для построения Web-страницы, можно воспользоваться одним из трех способов: создать черновик новой страницы и сохранить его, создать новую страницу на основе шаблона или открыть существующую, изменить ее и сохранить под новым именем с помощью команды “Сохранить как” (Save As).

1) Создание новой пустой Web-страницы.

а) Выбрать команду из меню “Файл” => “Создать” => “Страница или веб-узел” (File => New => Page or Web) либо щелкнуть мышью по кнопке “Новая страница” (New Page) (Ctrl+N).

б) Если выбрать команды меню “Файл” => “Страница или веб-узел” (File => New => Page or Web), то справа в главном окне программы появится область задач. Щелкнуть по гиперссылке “Пустая страница”(Blank Page) - чтобы создать пустую страницу. После щелчка по кнопке “Новая страница” (New Page) на первом шаге сразу появится пустая страница.

2) Открытие существующей Web-страницы.

а) Выбрать команду из меню “Файл” => “Открыть” (File => Open) или щелкнуть мышью по кнопке “Открыть” (Open), или нажать клавиши Ctrl+O.

б) После появления диалогового окна “Открыть файл” (Open File) необходимо перейти к тому файлу, который нужно открыть и выполнить двойной щелчок мышью, либо ввести его имя в поле “Имя” (File Name) и нажать на кнопку “Открыть” (Open).

в) В главном окне программы FrontPage будет открыта выбранная страница.

3) Сохранение Web-страницы

а) Выбрать команду из меню “Файл” => “Сохранить как” (File => Save As).

б) После появления диалогового окна “Сохранить как” (Save As), щелкнуть по кнопке перехода, чтобы перейти в директорию, в которой следует сохранить страницу.

в) В диалоговом окне “Сохранить как” (Save As) нажать на кнопку “Изменить” (Change), чтобы задать название страницы. Когда появится диалоговое окно “Указать заголовок страницы” (Set Page Title), ввести то название, которое Web-браузеры будут показывать как заголовок страницы, и нажать на кнопку ОК.

г) По умолчанию имя файла будет основано на первой строке текста в файле. Чтобы назвать файл по-другому, можно задать другое имя в текстовом поле “Имя” (File Name). Ввести после названия файла расширение .html, затем нажать на кнопку “Сохранить” (Save). Файл будет сохранен с этим названием.

4) Сохранение Web-страницы под другим именем.

а) Выбрать команду из меню “Файл” => “Сохранить как” (File => Save As).

б) После появления диалогового окна “Сохранить как” (Save As) перейти в директорию, где следует сохранить файл, затем нажать на кнопку “Сохранить” (Save), чтобы дать странице новое имя. После появления диалогового окна (Set Page Title) ввести название страницы, которое будет отображаться в браузере, и нажать на кнопку ОК.

в) Ввести имя файла в поле “Имя файла” (File Name) и нажать на кнопку “Сохранить” (Save). Файл будет сохранен под новым именем.

5) Закрытие Web-страницы.

После сохранения страницы выбрать команду из меню “Файл” => “Закрыть” (File => Close) (Ctrl+F4) или щелкнуть по кнопке с крестиком в верхнем правом углу страницы. Закроется только Web-страница, а программа FrontPage будет работать по-прежнему, и можно будет открыть другую страницу или работать над решением других задач.

1.3.5 Ввод и форматирование текста

Главная задача программы FrontPage - избавить пользователя от необходимости работать с HTML-кодировкой. Поэтому ввод и форматирование текста в FrontPage аналогично работе в любом текстовом редакторе: печатать текст, а заданное форматирование выполняется автоматически. Большинство стандартных способов работы с текстом в любом текстовом редакторе, таких как выделение, перемещение, вырезание и копирование, применимо и для программы FrontPage

В HTML-кодировке изменить шрифт слова с обычного на полужирный довольно непросто. Программа FrontPage позволяет менять стиль отображения

текста так же легко, как в текстовых редакторах, следует лишь выделить текст и нажать на кнопку. Панель инструментов “Форматирование” включает все необходимые для этого инструменты. Воспользовавшись диалоговым окном Шрифт (Font), можно получить дополнительные возможности работы с текстом.

Преимущество работы в FrontPage по сравнению с HTML-кодированием становится очевидным, когда необходимо добавить к Web-страницам специальные символы. Программа FrontPage позволяет добавить абсолютно любой символ и избавит от работы с эквивалентами в HTML-кодировке.

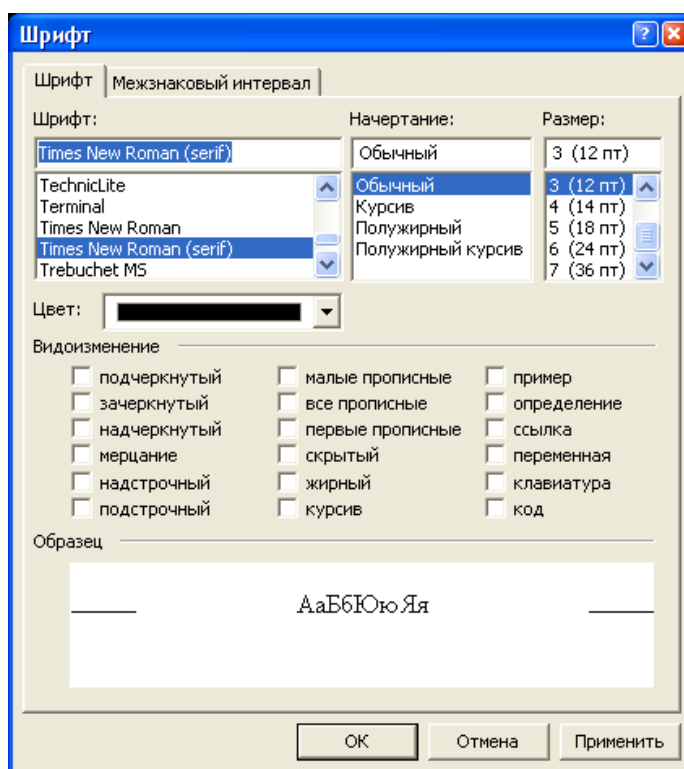


Рисунок 3 – Окно форматирования текста

1.3.6 Форматирование абзацев, списков и заголовков

Хотя рисунки очень украшают Web-сайт, однако, не они имеют решающее значение в структуре страниц. Иерархия определяется, прежде всего, такими элементами, как абзацы, списки и заголовки. В программе FrontPage легко можно конфигурировать эти элементы, в основном с помощью панели инструментов “Форматирование”. Если невозможно с помощью имеющихся настроек добиться нужного результата или нужно создать и использовать новый стиль, то следует работать со списками стилей.

Абзацы. В программе FrontPage можно менять внешний вид абзацев при помощи настроек выравнивания, отступа, интервала до и после абзаца и межстрочного интервала.

Отступы абзаца. Программа FrontPage позволяет выставлять абзацный отступ разными способами. Самый быстрый способ - использовать кнопки на

панели инструментов “Форматирование”. Более точные изменения требуют модификации определенных параметров в диалоговом окне “Абзац” (Paragraph).

Стиль абзаца. Самый простой способ быстро изменить “несколько параметров абзаца - это использование стиля Normal. По умолчанию в стиле Normal используется шрифт Times New Roman с одинарным отступом, выравниванием по левому краю и без отступов абзацев. Можно также применять встроенный стиль Formatted, который использует шрифт Courier и требует ручного ввода новых строк. Этот стиль выглядит не слишком красиво, но до того, как стала использоваться HTML-кодировка, он был единственным средством для выравнивания текста.

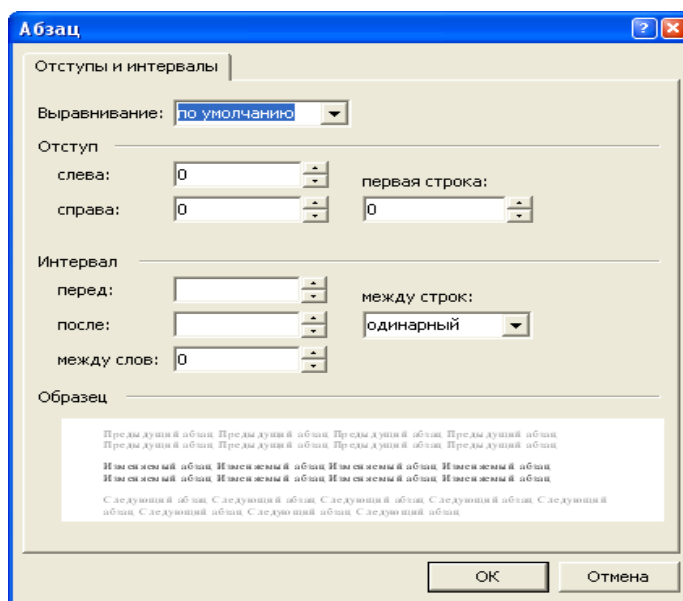


Рисунок 4 – Окно форматирования абзаца

Списки. Программа FrontPage позволяет создавать различные списки, которые можно разделить на два основных типа: нумерованные и нумерованные. Нумерованные списки (обычно их называют маркированными) очень удобны для отображения нескольких объектов, которые не имеют цифровой нумерации, например списка товаров. Если необходимо записать какую-либо последовательность, пользуйтесь нумерованным списком (рисунок 5).

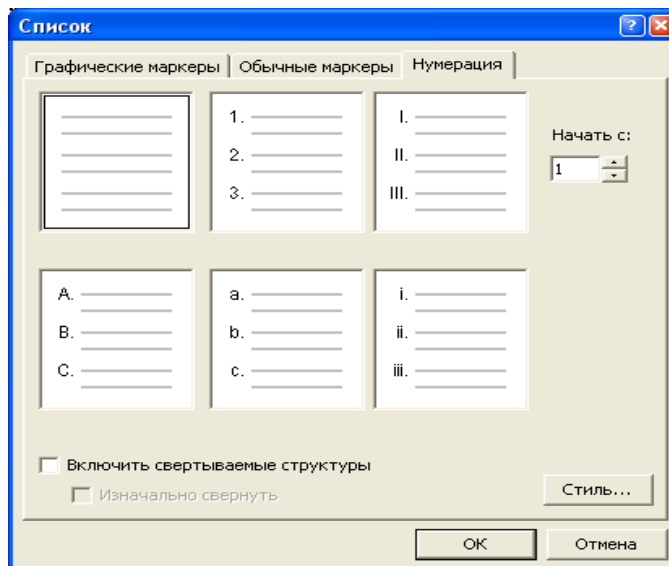


Рисунок 5 – Окно создания списков

Заголовки помогают посетителям сориентироваться на Web-сайте. Как и газетные заголовки, они позволяют читателю быстро находить нужную информацию. Как уже говорилось, следует использовать крупные заголовки для более важной информации, а маленькие - для менее существенной. Это имеет очень большое значение, так как при помощи заголовков приводят страницы сайта в порядок. Хотя в данном разделе рассказывается только об изменении размеров заголовков, таким же образом можно менять шрифт, цвет и стиль.

1.3.7 Гиперссылки

Одним из основных преимуществ при работе в Internet является возможность перехода от одного файла к другому посредством двойного щелчка мышью. Эта возможность реализована при помощи гиперссылок. Гиперссылки (или просто ссылки) могут быть разными: ссылки на другие файлы по всей сети, ссылки на разные части открытого документа, ссылки на рисунки, ссылки, служащие для отправки электронной почты. Программа FrontPage позволяет создавать любые ссылки /3/.

Чтобы успешно работать со ссылками, необходимо различать абсолютные и относительные ссылки. В абсолютной ссылке показан полный адрес (<http://www.peachpit.com/books/K5914.html>), а в относительной - только путь к файлу (</My Webs2/format.html>)

При создании связи с Web-страницей или файлом, которые находятся не на сайте, нужно использовать абсолютную ссылку. Относительная же ссылка служит для связи файлов на другой сайт. Преимущество использования относительных ссылок заключается в том, что можно модифицировать структуру сайта без изменения ссылок. Так как домашняя Web-страница имеет абсолютный адрес, браузер может переходить от одной относительной ссылки к другой.

Параметры вставки гиперссылок. В диалоговом окне “Добавление гиперссылок” (Insert Hyperlink) можно изменять любые параметры гиперссылок. Чтобы открыть диалоговое окно, необходимо выбрать пункт “Вставка” => “Гиперссылка” (Insert => Hyperlink) (Ctrl+K). По умолчанию в диалоговом окне отображается тот Web-сайт, с которым работают в данный момент (рисунок 6).

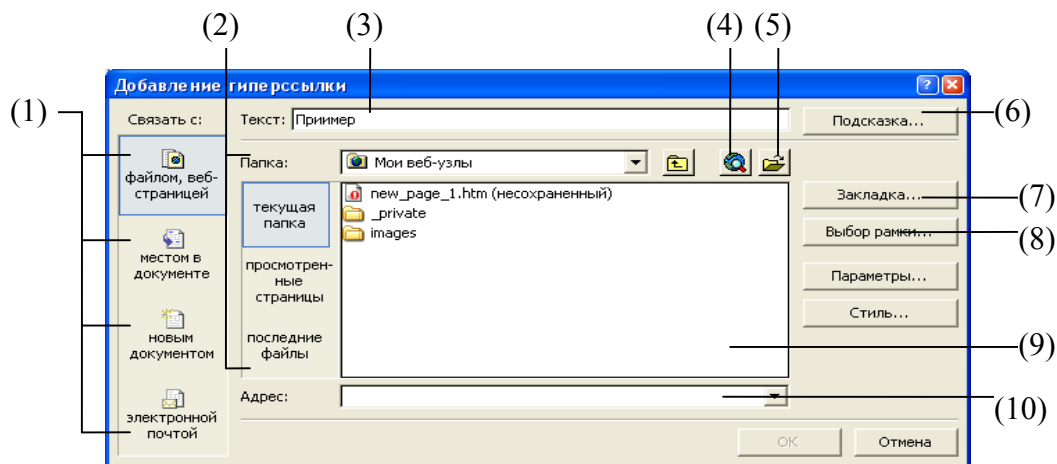


Рисунок 6 – Окно добавления гиперссылки

Добавление гиперссылки включает в себя следующие шаги:

1) Выбор типа гиперссылки. Чтобы выбрать тип гиперссылки, необходимо нажать на одну из следующих четырех кнопок в столбце ”Связать с” (Link to):

а) ссылка на уже существующий файл. Чтобы создать ссылку на файл, находящийся на компьютере, в локальной сети или Internet, необходимо нажать на кнопку “Файлом, Web – страницей” (Existing File or Web Page). При помощи кнопок списка Папка (Look in) найти нужный файл;

б) ссылка на закладку в текущем файле. Чтобы создать ссылку на закладку на активной странице, необходимо нажать на кнопку “Местом в документе” (Place in This Document).

в) ссылка на новый файл. Если требуется поместить ссылку на странице Web-сайта, которую ещё не успели создать, нажать на кнопку “Новый документ” (Create New Document). Появится диалоговое окно “Новый” (New), в котором задают обычную страницу или страницу на основе шаблона программы FrontPage;

г) ссылка на адрес электронной почты. Если требуется создать ссылку на адрес электронной почты необходимо нажать на кнопку “Электронной почтой” (E-mail Address).

2) Поиск файла. Для выбора файла, который будет связан с гиперссылкой, предназначены три кнопки: “Текущая папка” (Current Folder), “Просмотренные страницы” (Browsed Pages) и “Последние файлы” (Recent Files), а также две кнопки навигации; “Поиск в Internet” (Navigate Web using browser) и “Поиск на компьютере/в сети” (Navigate folders on local machine/network). По умолчанию активизирована кнопка “Текущая папка” (Current Folder), а содержимое текущей папки отображается в списке “Текущий

файл” (Current file). После нажатия кнопки “Просмотренные страницы” (Browsed Pages) на экране появится список URL-адресов тех страниц, которые открывались в программе Internet Explorer. Отображенные страницы соответствуют содержанию папки History в Internet Explorer. После нажатия кнопки “Последние файлы” (Recent Files) на экране отобразится список всех файлов, которые недавно открывали в программе FrontPage, независимо от того, на каком Web-сайте они находятся.

3) Текст для отображения в качестве гиперссылки. По умолчанию в поле “Текст” (Text to display) содержится текст, который выбран для гиперссылки. В большинстве случаев не нужно его изменять, но можно ввести в окно новый текст и нажать на кнопку ОК, чтобы заменить существующий.

4) Поиск по Internet с помощью кнопки браузера. Запустить браузер и найти в сети нужную страницу. Затем переключиться в программу FrontPage, и адрес будет автоматически вставлен в поле “Адрес” (Address).

5) Поиск по папкам на компьютере/в локальной сети. Нажать на кнопку “Поиск на компьютере/в сети” (Navigate folders on local machine/network), чтобы открыть любой файл на компьютере или в локальной сети. Адрес файла будет автоматически вставлен в поле “Адрес” (Address).

6) Создание всплывающей подсказки. Кнопка “Подсказка” (ScreenTip) позволяет создать маленькое текстовое сообщение, которое будет появляться в окне браузера, если поместить указатель мыши на ссылку. Однако эту подсказку увидит только посетитель сайта, использующий программу Internet Explorer версии 4 или более поздней.

7) Поиск закладки в другом файле. Если выбрать файл в списке Текущий файл (Current file), то можно при помощи кнопки “Закладка” (Bookmark) создать ссылку на закладку в другом файле, но не в текущем чтобы использовать закладку в текущем файле, нажать на кнопку “Местом в документе” (Place in This Document).

8) Изменение фрейма. Кнопку “Целевой фрейм” (Target Frame) следует применять, если Web-сайт содержит фреймы.

9) Просмотр списка текущих файлов. По умолчанию в этом списке отображаются файлы текущего Web - сайта. Если нажать на кнопку “Просмотренные файлы” (Browsed Pages) или “Последние файлы” (Recent Files), содержание списка изменится. Для просмотра всех файлов в списке необходимо пользоваться полосой прокрутки.

10) Адрес гиперссылки. FrontPage автоматически вводит в это поле адрес гиперссылки в зависимости от того, что выбрано в столбцах “Связь с” (Link to) и (Look in). Также можно ввести эту информацию, если известен URL-адрес файла. FrontPage по умолчанию начинает любой URL-адрес с http://, поэтому если требуется создать ссылку протокола FTP для загрузки файла, необходимо вручную заменить этот адрес на ftp://.

1.3.8 Закладки

Закладки позволяют посетителям сайта быстро переходить в нужные разделы на больших Web-страницах, например, к заголовкам. Создание закладок состоит из двух шагов: сначала нужно создать саму закладку, затем гиперссылки, которые на нее указывают. Закладка может располагаться на текущей или любой другой странице.

При помощи закладок читатели сайта смогут перемещаться по странице без необходимости просматривать весь документ. Закладки необязательно должны относиться к заголовкам. Можно создавать ссылку на фразу, слово, рисунок и даже на один символ.

Если потребуется обновить или удалить закладки на большой Web-странице, не обязательно искать их вручную - программа FrontPage позволяет найти их автоматически.

Создание закладки. Выделить текст, который нужно пометить закладкой, то есть тот текст, к которому при помощи закладки смогут переходить посетители сайта.

1) Выбрать пункты меню “Вставка” => “Закладка” (Insert => Bookmark) (Ctrl+G).

2) В открывшемся диалоговом окне “Закладка” (Bookmark) выделенный текст по умолчанию используется в качестве названия для закладки. При желании можно ввести другое название.

3) После ввода названия закладки нажать на кнопку ОК. Созданная закладка будет выделена подчеркиванием пунктирной линией.

1.3.9 Настройка цвета ссылки

По умолчанию программа FrontPage выделяет те ссылки, которые не посещались, голубым цветом, а посещенные ссылки - темно-синим. При желании можно задать для ссылок любые цвета. FrontPage позволяет задавать цвета для активных ссылок (которые срабатывают при нажатии на них мышью) и неактивных (которые меняют цвет при наведении на них указателя мыши).

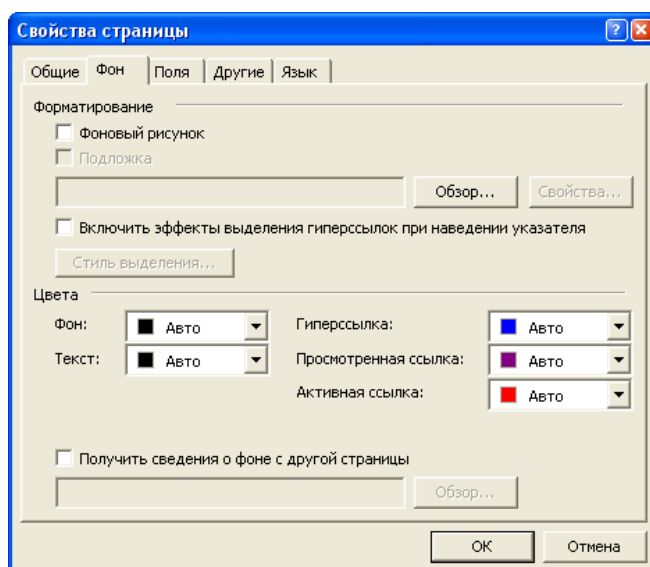


Рисунок 7 – Редактирование фона и цвета гиперссылок

Изменение цвета ссылки. Для изменения цвета ссылки необходимо:

1) Перейти в режим просмотра страницы при помощи кнопки Page на панели видов.

2) Щелкнуть по странице правой клавишей мыши и выбрать пункт “Свойства страницы” (Page Properties).

3) После появления диалогового окна Page Properties перейти на вкладку Фон (Background).

4) Чтобы изменить цвет ссылки необходимо нажать на одну из стрелок справа в разделе “Цвета” (Colors) и выбрать новый цвет в ниспадающем меню. Если необходимо задать произвольный цвет, нужно выбрать в ниспадающем меню пункт “Дополнительные цвета” (More Colors). В появившемся диалоговом окне “Дополнительные цвета” (More Colors) нажать на кнопку “Выбрать” (Select), выбрать цвет и нажать на кнопку ОК.

5) Чтобы использовать цвета ссылок, установленные для другой странице, после повторного появления диалогового окна “Свойства страницы” (Page Properties) необходимо поставить флажок “Получить информацию о фоне от другой страницы” (Get background information from another page), затем нажать на кнопку “Поиск” (Browse) для поиска нужной страницы. Цвета ссылок выбранной страницы будут автоматически применены для ссылок текущей страницы.

6) Закончив изменения, необходимо нажать на кнопку ОК. Цвета ссылок будут изменены только для текущей страницы.

1.3.10 Динамические кнопки

Динамические кнопки позволяют применить ряд интересных эффектов, которые будут активизироваться при наведении указателя мыши посетителями сайта. Например, внешний вид кнопки может изменяться, а перемещение указателя будет сопровождаться особым звуком. Очень часто динамические кнопки активизируют различные ссылки.

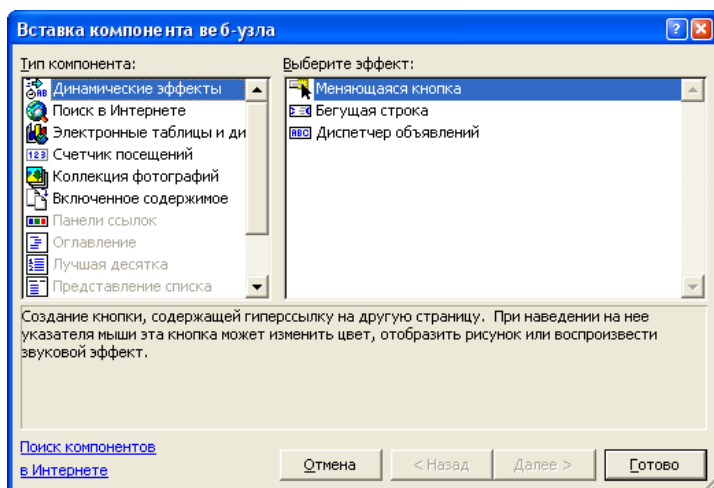


Рисунок 8 – Вставка компонента Web – узла

Вставка динамической кнопки. Для вставки динамической кнопки необходимо:

1) Перейти в режим просмотра страницы при помощи кнопки Page на панели видов. При этом необходимо убедиться в том, что в нижней части окна активизирована кнопка Normal.

2) Щелкнуть мышью в том месте, куда требуется добавить динамическую кнопку, и выбрать пункты “Вставка” => “Веб-компонент” (Insert => Web Component).

3) В появившемся диалоговом окне ”Вставка компонента веб-узла” (Insert Web Component) выбрать вариант “Динамическая кнопка” (Hover Button) в списке “Выбрать эффект” (Choose an effect). По умолчанию в списке “Тип компонента” (Component type) будет активизирован пункт “Динамические эффекты” (Dynamic Effects). Затем нажать на кнопку “Готово” (Finish).

4) В появившемся диалоговом окне “Свойства динамической кнопки” (Hover Button Properties) указать текст, шрифт, ссылку, цвет, тип эффекта и размер будущей кнопки.

5) Нажать на кнопку ОК.

6) Чтобы увидеть, как будет действовать эффект при перемещении указателя мыши, нажать на кнопку Preview в нижней части окна.

1.3.11 Рисунки

Форматы Web-изображений. Существует огромное количество форматов рисунков - BMP, TIFF, TGA, EPS, RAS. Но при работе в Internet используются только два формата - GIF и JPEG. В формате GIF следует сохранять маленькие рисунки, в формате JPEG - средние и большие.

В течение нескольких последних лет все большую известность приобретает формат PNG, который позволяет хорошо сжать рисунки без потери информации, как JPEG, и не связан с разбирательствами по поводу авторских прав, как GIF. Но, хотя браузеры версии 4 и старше поддерживают формат PNG, браузеры более ранних версий не могут с ним работать, что и препятствует его повсеместному распространению. Форматы GIF и JPEG остаются самыми популярными.

Рисунки не должны занимать много места. Перед тем как добавлять к Web-странице рисунки, максимально необходимо уменьшить размер их файлов. Огромные рисунки могут замедлить загрузку страниц до такой степени, что посетители просто уйдут с сайта, не дожидаясь ее завершения.

Обработку рисунков лучше всего выполнять с помощью специальной программы для работы с Web-графикой, например, Photoshop, DeBabelizer или Fireworks. Необходимо начать с удаления с рисунка всей лишней информации /4/.

Самым важным моментом в сжатии файлов является способ, который для этого используется. При сжатии формата GIF уменьшается количество цветов в рисунке. При очень сильном сжатии формата JPEG необходимо следить за сохранением приемлемого внешнего вида рисунка.

На размер файла большое влияние оказывает количество цветов в рисунке (обычно его называют глубиной цвета). Многие пользователи Internet работают с 8-битными мониторами, которые могут воспроизводить не более 256 цветов одновременно. И так как более 40 из этих цветов резервируются для операционной системы компьютера, формат GIF использует палитру из 216 цветов, которую обычно называют Web-палитрой. Процесс конвертирования изображения с высоким разрешением в изображение с Web-палитрой называют индексированием, или отображением рисунка. Такое название появилось потому, что цветовая информация для каждого пиксела изображения сохраняется в отдельной ячейке с индексом.

Преобразование рисунка с несколькими миллионами цветов в рисунок с двумя сотнями цветов может показаться неверным шагом, но следует учесть, что по Internet путешествуют пользователи не только с самыми совершенными компьютерами и последней версией операционной системы. Чтобы найти компромисс между минимально допустимой глубиной битов и приемлемым качеством изображения, необходимо поэкспериментировать с графическими программами и попробовать сохранить рисунок несколько раз с разными глубинами битов.

Не рекомендуется использовать уменьшение глубины битов для фотографий, поскольку в формате GIF фотографическое изображение выглядит очень нечетко. В таких случаях следует использовать формат JPEG, в котором применяется совершенно другой метод сжатия. С помощью программы FrontPage можно увеличивать или уменьшать размеры файлов в формате JPEG. Единственная проблема при работе с форматом JPEG-плохое качество сжатия прозрачных цветов (рисунки выглядят поврежденными).

Если после сжатия размеры рисунка все еще велики, можно попробовать воспользоваться функцией эскиза программы FrontPage.

Вставка рисунков. После того как с помощью графической программы максимально сжаты файлы рисунков, можно без особого труда украсить ими Web-страницу.

Вставка рисунка. Для того чтобы осуществить вставку рисунка необходимо:

- 1) Переключиться в режим просмотра страницы с помощью кнопки “Page“ на панели видов, затем щелкнуть мышью по странице в месте предполагаемой вставки рисунка.

- 2) Выбрать пункт из меню “Вставка” => “Рисунок” (Insert => Picture), а затем - в подменю пункт “Объект clip art” (Clip Art) или “Из файла” (From File).

- 3) Если выбрать пункт “Картинки” (Clip Art), то в правой части главного окна появится область задач “Вставка картинки” (Insert Clip Art). Для поиска рисунка необходимо пользоваться полем “Поиск текста” (Search text) и кнопкой

“Поиск” (Search). Щелкнуть по названию файла в области задач, и рисунок добавится в Web-страницу.

Если выбрать пункт “Из файла” (From File), появится диалоговое окно “Картинки” (Picture). Необходимо пользоваться кнопками “Папка” (Look in) или кнопками навигации, чтобы найти рисунок. Нажать на кнопку “Вставка”(Insert), чтобы поместить его на страницу.

4) Сохранить страницу вместе с новым рисунком (Ctrl+S). Если использовали рисунок, который находился не на текущем Web-сайте, появится диалоговое окно “Сохранить внедренный рисунок” (Save Embedded Files).

5) Чтобы изменить имя файла необходимо нажать на кнопку Rename. Чтобы сохранить файл по другому адресу необходимо щелкнуть по кнопке “Сменить папку” (Change Folder). Затем нажать на кнопку ОК.

6) После сохранения рисунок можно отредактировать.

1.3.12 Мультимедиа

Некоторое количество элементов мультимедиа может украсить Web-страницу и привлечь внимание посетителя. Но следует учесть, что посетители не станут долго ждать окончания загрузки, поэтому всегда необходимо использовать индикатор загрузки FrontPage, чтобы оценить время, необходимое страницам с элементами мультимедиа для появления на экране.

FrontPage включает в себя три основных категории, с помощью которых вполне можно превратить свою Web-страницу в мультимедийную: видеозаписи (или анимации), фоновый звук страницы и Web-компоненты.

Добавление видео и анимации. Хотя технически и допускается поместить часовой учебный видеоролик на Web-сайт, имеется не так-то много достаточно быстрых сетевых каналов, чтобы работать с таким огромным файлом. Для размещения в Internet более подходят видеозаписи и анимации небольшого размера, такие как логотип с бегущей строкой или анимированные стрелки.

Для добавления видео и анимации необходимо:

1) Переключиться в режим просмотра страницы и щелкнуть в том ее месте, где необходимо поместить видео или анимацию.

2) Выбрать пункт из меню “Вставка” => “Рисунок” => “Фильм” (Insert => Picture => Video).

3) В появившемся диалоговом окне “Вставка фильма” (Video) показываются все файлы, находящиеся на Web-сайте. Если нужные видеозаписи или анимации там не отображены, то необходимо воспользоваться ниспадающим меню, значками папок или кнопками, чтобы добраться до соответствующего файла.

4) После обнаружения файла необходимо возвратиться в диалоговое окно Video, нажать “Открыть” (Open), чтобы поместить видео или анимацию на Web-страницу.

5) Сохранить страницу с добавленными объектами (Ctrl+S). Если этот объект находится не на разрабатываемом Web-сайте, то появится диалоговое окно “Сохранить внедренные файлы” (Save Embedded Files).

6) Нажать “Переименовать” (Rename), чтобы изменить имя файла, и “Сменить папку” (Change Folder), чтобы сохранить этот файл в другом месте. Закончив, нажать ОК, и видео или анимация будут сохранены.

7) Как только объект вставлен и сохранен, можно настроить для него частоту повторений и другие параметры.

Добавление на страницу фонового звука. Короткие звуковые циклы могут быть как эффективным средством улучшения восприятия соответствующих страниц, так и доводящим до бешенства раздражителем на строго деловых сайтах. Поэтому необходимо тщательно обдумывать каждый случай использования звукового фона.

1) Переключиться в режим просмотра страницы, щелкнуть правой клавишей мыши на ее свободной части и выбрать пункт “Параметры страницы” (Page Properties) в контекстном меню.

2) В появившемся диалоговом окне “Параметры страницы” (Page Properties) выбрать вкладку “Общие” (General).

3) В разделе “Фоновый звук” (Background sound) нажать на кнопку Browse, чтобы найти нужный звуковой файл. По умолчанию здесь установлен флажок “Непрерывное воспроизведение” (Forever). Если необходимо, чтобы звуковой файл воспроизводился недолгое время можно снять этот флажок и установить число повторений в текстовом поле или с помощью списков.

4) Нажать кнопку ОК, и звук станет элементом Web-страницы. Выбрать кнопку “Предварительный просмотр” (Preview) в нижней части главного окна FrontPage, чтобы воспроизвести звуковой файл, или проверить звучание при помощи Web-браузера.

1.3.13 Web-компоненты

FrontPage включает более двух десятков разных мультимедийных компонентов, которые могут сделать страницы более интерактивными. Число этих Web-компонентов, один из которых был использован вами для создания средств управления воспроизведением видео, существенно пополнилось в FrontPage 2002. Помимо компонентов, включенных в FrontPage 2000 (сменяющие друг друга баннеры, счетчики посещений страницы и прокручиваемые бегущие строки), версия 2002 позволяет создавать оглавления (tables of content), списки наиболее посещаемых разделов сайта, страницы с динамически обновляемой информацией и страницы, чувствительные ко времени (например, автоматическое обновление сводки биржевых котировок через заданные промежутки времени).

Однако некоторые из этих Web-компонентов требуют установки расширений FrontPage 2002 на Web-сервере. Хотя эти компоненты и предоставляют замечательные возможности, в конечном счете, они исключают

из числа посетителей вашего сайта пользователей Internet, не работающих с программными продуктами Microsoft.

Вставка рекламного баннера. Чтобы осуществить вставку рекламного баннера необходимо:

1) Переключиться в окно просмотра страницы, щелкнуть в том месте, где требуется поместить рекламный баннер, и выбрать пункт “Вставка” => “Веб-компонент” (Insert => Web Component).

2) В списке “Тип компонента” (Component type), расположенном в левой части окна “Вставка компонента веб-узла” (Insert Web Component), выбрать пункт “Динамические эффекты” (Dynamic Effects), пункт “Менеджер рекламных баннеров” (Banner Ad Manager) - в списке справа и нажать кнопку “Готово” (Finish).

3) В диалоговом окне “Свойства менеджера рекламных баннеров” (Banner Ad Manager Properties) - установить размеры изображений баннеров.

4) Используя раскрывающееся меню, выбрать Эффект перехода (Transition effect) для перехода от одного изображения к другому.

5) Определить нужную продолжительность показа (в секундах) в текстовом поле “Показывать каждый рисунок в течение...” (Show each picture for (seconds)).

6) При помощи кнопки Browse выбрать Web-страницы, на которые попадут посетители, если щелкнут по этому рекламному баннеру.

7) Используя кнопку “Добавить” (Add), найти все изображения, которые необходимо поместить в рекламный баннер. При помощи кнопок “Переместить вверх” (Move Up) и “Переместить вниз” (Move Down) уточнить порядок показа изображений.

8) Сделав выбор, нажать ОК, затем сохранить страницу (Ctrl+S).

9) FrontPage может показывать только первый кадр баннерной рекламы, поэтому необходимо запустить Web-браузер, чтобы просмотреть воспроизведение всего баннера.

Добавление счетчика посещений страницы. Для того, чтобы добавить счетчик посещений страницы необходимо:

1) Переключиться в режим просмотра страницы и щелкнуть в том ее месте, где требуется поместить счетчик.

2) Выбрать пункты “Вставка” => “Веб-компонент” (Insert => Web Component).

3) В появившемся диалоговом окне “Вставка компонента веб-узла” (Insert Web Component) выделить пункт “Счетчик посещений” (Hit Counter) в расположенном слева списке “Тип компонента” (Component type), выбрать стиль оформления нового объекта в списке справа и нажать на кнопку “Готово” (Finish).

4) В появившемся диалоговом окне “Свойства счетчика посещений” (Hit Counter Properties) установить флажок “Установить счетчик на” (Reset counter to) и ввести в текстовое поле число, если требуется, чтобы счетчик запускался с этим конкретным числом.

5) Установить флажок “Ограниченный размер счетчика” (Fixed number of digits), чтобы счетчик обнулялся при достижении 10, 100, 1000 и т.п.

6) Сделав необходимые настройки, нажать на кнопку ОК. Условный значок счетчика появится на странице, но сам счетчик будет отображаться только после опубликования сайта в Internet.

Вставка бегущей строки. Для того, чтобы осуществить вставку бегущей строки необходимо:

1) Переключиться в режим просмотра страницы и щелкнуть в той ее части, где требуется поместить бегущую строку.

2) Выбрать пункты меню “Вставка” => “Веб-компонент” (Insert => Web Component).

3) В появившемся диалоговом окне “Вставка компонента веб-узла” (Insert Web Component) выбрать пункт “Динамические эффекты” (Dynamic Effects) в расположенном слева списке “Тип компонента” (Component type), затем пункт “Бегущая строка” (Marquee) - в списке справа и нажать на кнопку “Готово” (Finish).

4) В открывшемся диалоговом окне Свойства бегущей строки (Marquee Properties) ввести текст, который необходимо видеть в виде бегущей строки; настроить направление движения текста, скорость, специальные эффекты и выравнивание, размер, частоту повторений и цвет фона.

5) Выполнив необходимые настройки, нажать на кнопку ОК и сохранить страницу (Ctrl+S). Бегущая строка будет помещена на Web-страницу.

6) Чтобы посмотреть, как выглядит бегущая строка, и определить, нужна ли дальнейшая настройка, выбрать в строке состояния FrontPages вкладку Preview.

1.3.14 Таблицы

Благодаря форме, удобной для быстрого просмотра, таблицы значительно облегчают восприятие связного текста или изображений. Кроме того, таблицы можно очень красиво оформить, используя различные рамки и яркие цвета. Главное преимущество использования таблиц состоит в том, что независимо от типа и версии своего браузера посетители увидят все примененные при их создании эффекты. Именно поэтому так много коммерческих Web-сайтов разрабатываются на основе таблиц страничного размера, подобных компоновочным сеткам, используемых при традиционной верстке. Однако гораздо лучше составлять страницу из нескольких таблиц, нежели применять всего одну. Благодаря этому верхняя часть страницы будет показываться на экране быстрее, чем продолжающие загружаться таблицы, расположенные ниже /4/. С помощью панели инструментов “Таблицы” (Tables) можно строить создавать самые разные таблицы от простых до весьма сложных. Необходимо активизировать ее, выбрав “Вид” => “Панели инструментов” => “Таблицы” (View => Toolbars => Tables).

Вычерчивание таблицы. Для того чтобы вычертить таблицу необходимо:

1) Переключиться в окно просмотра страницы и щелкнуть в том ее месте, где необходимо поместить таблицу.

2) Выбрать пункты “Таблица” => “Нарисовать таблицу” (Table=>Draw Table) или, если открыта панель инструментов Tables, нажать на ней кнопку “Нарисовать таблицу” (Draw Table).

3) На странице появится указатель мыши в виде карандаша. Удерживая нажатой левую клавишу мыши, перемещать карандаш, пока выделенный штриховой линией прямоугольник не превратится в таблицу нужного размера. Затем отпустить клавишу мыши, и на странице появится простая таблица.

4) Разбить таблицу на столбцы и строки, «рисую» указателем мыши внутренние границы. Сделать двойной щелчок мышью, чтобы переключить карандаш обратно в вертикальный курсор.

5) Для изменения размера столбца или строки таблицы необходимо поместить указатель на границу ячейки, при этом он примет вид двунаправленной стрелки. Удерживая нажатой левую клавишу мыши, переместить эту границу в нужном направлении. Отпустить клавишу мыши, и таблица будет изменена.

Автоматическая вставка таблицы. Для автоматической вставки таблицы необходимо:

1) Переключиться в окно просмотра страницы и щелкнуть в том ее месте, где необходимо поместить таблицу.

2) Выбрать пункты “Таблица” => “Вставка” => “Таблица” (Table => Insert => Table).

3) В появившемся диалоговом окне “Вставка таблицы” (Insert Table) воспользоваться разделом “Размер” (Size), для установки нужного числа “строк” (Rows) и “столбцов” (Columns) таблицы.

4) В поле “Разметка” (Layout) установить следующие параметры таблицы: “Выравнивание” (Alignment) - по левому краю, по центру или по правому краю, Border size (Ширина границы) - толщина внешней границы, “Отступ” (Cell padding) и “Поля ячеек” (Cell spacing).

5) Установить флажок “Ширина” (Specify width), чтобы задать ширину всей таблицы и используя группу переключателей, чтобы указать, должна ли эта ширина быть абсолютной (в пикселах) или относительной к ширине Web-страницы посетителя (в процентах).

6) Закончив, нажать кнопку ОК, и таблица будет помещена на вашу Web-страницу.

1.3.15 Фреймы

Одно из больших преимуществ создания страниц на основе фреймов состоит в том, что можно формировать навигационные гиперссылки ко всем частям сайта или добавлять кнопки, которые остаются видимыми на одном из фреймов даже тогда, когда посетитель прокручивает содержимое основного фрейма. Благодаря этому посетители легко ориентируются на сайте, без труда переходя к разным его частям.

Тем не менее, у фреймов есть несколько недостатков. Заголовок страницы остается одним и тем же, даже когда посетители перемещаются от фрейма к фрейму, что приводит к разного рода недоразумениям. Посетители не могут сделать закладку на конкретном фрейме в своих Web-браузерах, получая вместо этого ссылку на домашнюю страницу сайта. Один из методов решения этих проблем состоит в использовании навигационных панелей, сохраняющих свободу передвижения по сайту.

Создание фреймовой страницы. Для создания фреймовой страницы необходимо:

1) Перейти в окно просмотра страницы, затем выбрать в меню пункты “Файл” => “Создать” => “Страница или Web” (File => New => Page or Web) (Ctrl+N).

2) В появившейся области задач щелкнуть по ссылке “Шаблоны страниц” (Page Templates).

3) В открывшемся диалоговом окне Page Templates выбрать вкладку “Фреймовые страницы” (Frames Pages). Использовать область “Предварительный просмотр” (Preview) диалогового окна, чтобы выбрать шаблон фреймовой страницы, затем нажать на кнопку ОК. Появится основанная на этом шаблоне пустая фреймовая страница, которая в каждом из своих фреймов предлагает два варианта: “Установить исходную страницу” (Set Initial Page) и “Создать страницу” (New Page).

4) Чтобы связать фрейм с существующей на Web-сайте страницей, необходимо щелкнуть по кнопке Set Initial Page. В появившемся диалоговом окне “Добавление гиперссылки” (Insert Hyperlink) - перейти к той странице, с которой работать, и нажать кнопку ОК. Страница появится во фрейме. Чтобы создать для фрейма новую пустую страницу необходимо нажать “Новая страница” (New Page). Внутри фрейма появится пустая страница, основанная на шаблоне “Обычный” (Normal). Теперь можно разместить в этом фрейме любую информацию.

5) Нажать клавиши Ctrl+S, чтобы сохранить новый фрейм и фреймовую страницу. Если создана новая страница для какого-либо из фреймов, то фрейм этой страницы будет выделен в области предварительного просмотра диалогового окна “Сохранить как” (Save As).

6) Дать новой странице имя и заголовок, отличные от уже существующих на сайте, и затем нажать кнопку “Сохранить” (Save). Новая страница будет сохранена, после чего появится диалоговое окно “Сохранить как” (Save As) для сохранения фреймовой страницы целиком (в окне предварительного просмотра эту страницу будет окружать широкая синяя рамка).

7) Дать фреймовой странице имя и заголовок, отличные от уже существующих на сайте и нажать кнопку “Сохранить” (Save). Фреймовая страница будет сохранена.

1.4 Контрольные вопросы

- 1) Абсолютные и относительные ссылки;
- 2) Гиперссылки: создание, назначение;
- 3) Чем отличается закладка от гиперссылки;
- 4) Для чего нужны динамические кнопки;
- 5) Каким образом можно вставить фоновый звук в Web-документ;
- 6) Форматирование абзацев. Основные принципы;
- 7) Что необходимо сделать для загрузки файла используя FTP;
- 8) Как вставить счетчик посещений страницы.

1.5 Тесты

- 1) Что такое абсолютные ссылки?
 - а) ссылка содержит полный адрес к файлу;
 - б) в ссылке показан только путь к файлу(/My Webs2/format.htm);
 - в) например, адрес (<http://www.peachpit.com/books/ K5914.htm>).
- 2) Что такое относительные ссылки?
 - а) в ссылке показан полный адрес к файлу;
 - б) в ссылке показан только путь к файлу(/My Webs2/format.htm);
 - в) например, адрес (<http://www.peachpit.com/books/ K5914.htm>).
- 3) Гиперссылки: назначение?
 - а) ссылки могут указывать на другой документ, специальное место данного документа;
 - б) выполнять некоторые функции, например запрашивать файл по FTP-протоколу;
 - в) служат для выделения текста;
 - г) нужны для анимации текста при наведении на неё курсора мыши.
- 4) Чем отличаются внутренние гиперссылки от внешних?
 - а) внутренние гиперссылки содержат полный путь внутри сервера;
 - б) внутренние гиперссылки содержат полный путь к документу с указанием конкретного участка текста через знак (#);
 - в) внешние гиперссылки содержат полный путь к документу без указания протокола передачи данных.
- 5) Для чего нужны динамические кнопки?
 - а) динамические кнопки позволяют применить ряд интересных эффектов, которые будут активизироваться при наведении указателя мыши посетителями сайта. Например, внешний вид кнопки может изменяться;
 - б) динамические кнопки позволяют использовать динамическое отображение страниц.

- б) Что необходимо сделать для загрузки файла, используя FTP;
- а) подать заявку системному администратору;
 - б) создать ссылку: <http://www.admin.ru/ask.ftp>
 - в) создать ссылку: <ftp://www.admin.ru/ask.zip>.

2 Лабораторная работа №2

Создание простейших Web-страниц с использованием HTML. Создание ссылок. Форматирование текста

2.1 Цель работы

Изучение основ создания HTML-документов. Создание HTML-документа с использованием HTML-тэгов. Логическое и физическое форматирование текста. Использование ссылок.

2.2 Порядок выполнения работы

1) Создать новый HTML документ с использованием какого-либо обычного или специализированного текстового редактора. В тексте должны содержаться тэги необходимые для любого HTML документа (`<HTML>`, `<BODY>`). Добавить текст по определенной теме. Сохранить данный документ в свою папку вместе с ранее сохраненными документами под названием "index.html".

2) Задать заголовок, отображаемый в строке заголовка окна браузера. Текст строки заголовка - фамилия на русском языке. Использовать при этом тэг `<TITLE>`.

3) Установить необходимые цвета фона и текста документа. Для установки цветов документа используются параметры "BGCOLOR", "TEXT", внутри открывающего тэга `<BODY>`.

4) Написать произвольный абзац текста самым крупным шрифтом. В данном случае следует использовать тэги-контейнеры отвечающие за размер шрифта, например `<H2>`. Осуществить выравнивание различных абзацев (отцентрировать, прижать вправо, влево, выровнять по ширине).

5) Произвести логическое форматирование текста как небольшого фрагмента программного кода с выделением.

6) Произвести физическое форматирование текста, как небольшого фрагмента моноширинным шрифтом, с подчеркиванием и с эффектом мигания.

7) Написать определения трех любых понятий данного курса различными цветами и гарнитурами (Arial, Times, моноширинный шрифт). Сам

термин должен быть выведен полужирным курсивом с подчеркиванием. Добиться данного эффекта можно использованием тэга ** с параметрами.

8) Добавить в Web-документ пять видов ссылок.

9) Провести эксперименты с Escape-последовательностями. При этом не стоит забывать, что данные последовательности чувствительны к регистру.

К сдаче лабораторной предоставляются: работающая страница на сервере с комментариями каждого тэга и его свойств (атрибутов) в исходнике.

2.3 Методические указания

2.3.1 Общие положения языка HTML

Язык HTML (Hypertext Markup Language, язык разметки гипертекста) — это язык структурной разметки, используемый для создания Web-страницы. HTML-документы могут просматриваться различными типами Web-браузеров. Когда документ создан с использованием HTML, Web-браузер может интерпретировать HTML для выделения различных элементов документа и первичной их обработки. Использование HTML позволяет форматировать документы для их представления с использованием шрифтов, линий и других графических элементов на любой системе, их просматривающей.

Web-страницы могут быть созданы с помощью:

- 1) обычного текстового редактора;
- 2) редактора, способного сохранять текст в формате HTML;
- 3) специализированного редактора;
- 4) специализированной системы.

HTML-документы сохраняются на диске как обычные текстовые документы в формате ANSI. Для распознавания Web-страниц по их именам общепринято обозначать такие файлы использованием расширений .HTM (для Windows 3.1) или .HTML (для Windows 95/98/Me/NT/XP, Macintosh и Unix).

Кроме полезного текста в HTML-документах используются специальные управляющие буквенно-цифровые коды (Тэги), помещенных в угловые скобки **, *<HTML>*, **, *<HR>*. Большинство элементов состоят из пар тэгов — открывающего и закрывающего. Открывающий тэг — это просто мнемонический символ элемента, заключенный в угловые скобки **. Закрывающий тэг идентичен открывающему, за тем исключением, что перед символом ставится прямая наклонная черта **. Хотя некоторые элементы не требуют ставить закрывающий тэг, потому что он ничего не окружает *
*.

Структура тэга в общем, виде имеет следующий вид:

<ТЭГ параметр1=значение1...>

текст, на который распространяется действие тэга

</ТЭГ>

или

<ТЭГ параметр1=значение1...>

текст

другие конструкции

При интерпретации тэгов браузер не делает различия между строчными и прописными буквами. Поэтому сами тэги можно набирать на любом регистре. Зачастую параметр (атрибут) является необязательной величиной и его можно пропускать.

2.3.2 Структура HTML-документа

Когда Web-браузер получает документ, он определяет, как документ должен быть интерпретирован. Самый первый тэг, который встречается в документе, должен быть тэгом <HTML>. Данный тэг сообщает Web-браузеру, что документ написан с использованием HTML. Минимальный HTML-документ будет выглядеть так:

<HTML> ...тело документа... </HTML>

В общем виде Web-документ имеет следующую структуру:

<HTML>

<HEAD>

<TITLE>Название документа </TITLE>

</HEAD>

<BODY>

... тело документа

</BODY>

</HTML>

<HTML> и </HTML> заключают внутри себя все элементы HTML-кода, указывая, что используется язык HTML.

<HEAD> и </HEAD> обозначают заголовок Web-документа. Как правило, заголовок содержит название документа, информацию для индексирования и некоторые общие установки для данного документа. Тэг заголовочной части документа должен быть использован сразу после тэга <HTML> и более нигде в теле документа.

<BODY> и </BODY> обрамляют оставшуюся часть документа (тело). Здесь размещается основная смысловая текстовая и графическая информация.

Разделение документа на заголовочную часть и тело имеет лишь смысловую нагрузку. Текст, приведенный в любой из этих частей, на экране выглядит совершенно одинаково.

Внутри контейнера <HEAD> может использоваться тэг <TITLE>, как показано выше. Большинство Web-браузеров отображают содержимое этого тэга в заголовке окна, содержащего документ и в файле закладок, если он поддерживается WEB-браузером. Заголовок документа не появляется при отображении самого документа в окне /5/.

Как любой язык, HTML позволяет вставлять в тело документа пояснительный текст (комментарии), который сохраняется при передаче

документа по сети, но не отображается браузером. Синтаксис комментария следующий:

```
<!-- Это комментарий -->
```

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

2.3.3 Форматирование текста

При выводе на экран текста браузер игнорирует дополнительные пробелы, символы табуляции и возврата каретки, пустые строки. Их можно использовать для того, чтобы сделать текст HTML-документа более легко читаемым, но для правильного его отображения на Web-странице следует использовать специальные тэги:

а) Цвета в HTML-документе.

Язык HTML определяет следующие типы цветов:

- **BGCOLOR** (цвет фона для тела документа),
- **TEXT** (цвет, используемый при выводе на экран текста из данного документа),
- **LINK** (цвет, который будет использоваться при выводе на экран текста из еще не выбранных вами гипертекстовых связей),
- **VLINK** (цвет, который будет использоваться при выводе на экран текста из уже проверенных вами гипертекстовых связей),
- **ALINK** (цвет, которым будут выделяться в тексте гипертекстовые связи в тот момент, когда пользователь щелкает по ним клавишей мыши).

Существует две формы задания цвета: символьная (указывается название одного из predetermined цветов) и цифровая (комбинация RGB: #RRGGBB).

Символьные идентификаторы основных цветов и их RGB-комбинации приведены ниже:

- BLACK (#000000) — черный,
- SILVER (#C0C0C0) — серебряный,
- GRAY (#808080) — серый,
- WHITE (#FFFFFF) — белый,
- MAROON (#800000) — темно-красный,
- RED (#FF0000) — красный,
- PURPLE (#800080) — темно-сиреневый,
- FUCHSIA (#FF00FF) — сиреневый,
- GREEN (#008000) — зеленый,
- LIME (00FF00) — ярко-зеленый,
- OLIVE (#808000) — оливковый,
- YELLOW (#FFFF00) — желтый,
- NAVY (#000080) — темно-синий,
- BLUE (#0000FF) — голубой,
- TEAL (#008080) — сине-зеленый,
- AQUA (#00FFFF) — ярко-голубой.

Атрибут, указывающий на цвет, может использоваться в тэгах <BODY>, , <HR>, <MARQUEE>, <TABLE>.

б) Элементы форматирования на уровне блоков.

<P align=left|center|right|justify nowrap> текст </P> - разделяет два абзаца пустой строкой. Может не иметь пары </P>. Параметр ALIGN задает выравнивание информации. Применение параметра NOWRAP дает возможность писать текст без переноса слов. Для центрирования текста или графики можно использовать также контейнер <CENTER>.

<H1|H2|H3|H4|H5|H6 align=...> текст </H1|H2|H3|H4|H5|H6> - используются для выделения структурных частей текста. Каждый стиль заголовка имеет свой размер. Тэг <H1> имеет наибольший размер.

<HR align=... size=... width=... noshade> - предназначен для вычерчивания горизонтальной линии. Атрибут SIZE задает толщину линии в пикселях, WIDTH — ее ширину в пикселях или процентах от ширины окна браузера. Атрибут NOSHADE позволяет представить линию без тени в виде простой темной полосы.

 - вызывает переход на новую строку без разрыва абзаца.

<WBR></WBR> - определяет место возможного (рекомендуемого) переноса (разрыва) строки.

<NOBR></NOBR> - заключает в себе текст, который не должен разбиваться на строки, даже если она выходит за границы экрана. Вместо этого браузер позволит горизонтально прокручивать текст.

<PRE></PRE> - , выводится браузером на экран как есть — со всеми пробелами, символами табуляции и конца строки. Это очень удобно при создании простых таблиц.

<BLOCKQUOTE></BLOCKQUOTE> - предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный этим тэгом, отступает от левого края документа на 8 пробелов.

Пример:

<p>Шрифт размера 1</p>

<p> Шрифт размера 2</p>

<p> Шрифт размера 3</p>

<p> Шрифт размера 4</p>

<p> Шрифт размера 5</p>

<p> Шрифт размера 6</p>

<p> Шрифт размера 7</p>

В браузере это будет выглядеть следующим образом:

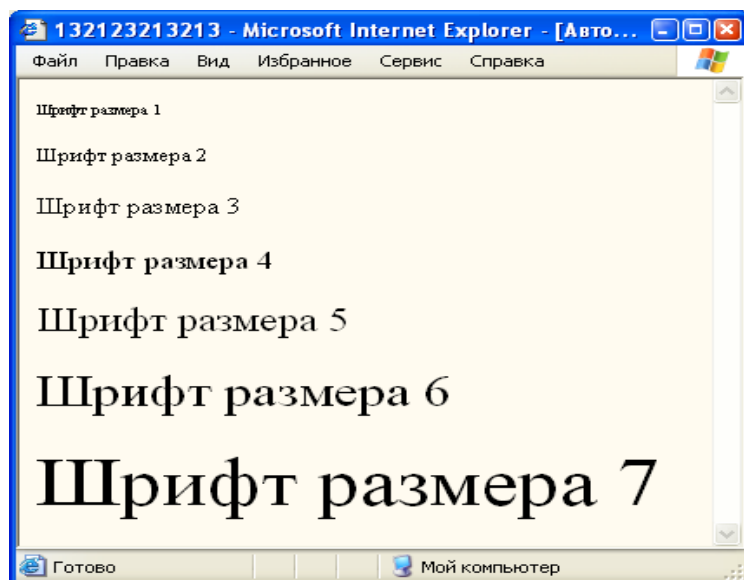


Рисунок 9 – Размеры шрифтов

в) Элементы логического форматирования текста.

<ABBR></ABBR> - Отмечает текст как аббревиатуру.

<ACRONYM></ACRONYM> - Так же, как и тэг **<ABBR>** используется для отметки аббревиатур. Этим тэгом рекомендуется отмечать так называемые акронимы, т.е. произносимые слова, состоящие из аббревиатур.

<CITE></CITE> - Используется для отметки цитат или названий книг и статей, ссылок на другие источники.

<CODE></CODE> - Отмечает свой текст как небольшой фрагмент программного кода.

**** - Отмечает свой текст как удаленный.

<DFN></DFN> - Отмечает свой текстовый фрагмент как определение

<INS></INS> - Отмечает свой текст как вставку.

**** - Используется для выделения важных фрагментов текста.

<KBD></KBD> - Отмечает текст как вводимый пользователем с клавиатуры.

<Q></Q> - Отмечает короткие цитаты в строке текста.

<SAMP></SAMP> - Отмечает текст как образец.

**** - Используется для выделения важных фрагментов текста

<VAR> </VAR> - Отмечает имена переменных программы.

г) Элементы физического форматирования текста

**** - Отображает текст полужирным шрифтом.

<I></I> - Отображает текст курсивом..

<TT></TT> - Отображает текст моноширинным шрифтом.

<U></U> - Отображает текст подчеркнутым.

<STRIKE></ STRIKE> - Отображает текст подчеркнутым.

<S></S> - Отображает текст, перечеркнутым горизонтальной линией.

<BIG></BIG> - Отображает текст большего размера.

`<SMALL></SMALL>` - Отображает текст меньшего размера.

`` - Сдвигает текст ниже уровня строки и выводит его шрифтом меньшего размера.

`` - Сдвигает текст выше уровня строки и выводит его шрифтом меньшего размера.

`<BLINK></BLINK>` - Отображает мигающий текст.

`<BASEFONT>` - Используется для указания размера, типа и цвета шрифта, используемого в документе по умолчанию

` текст ` - Указывает параметры шрифта.

size — устанавливает размер шрифта, который будет использоваться текстом, содержащимся в пределах элемента font. Можно задать абсолютный размер шрифта, указав какое-либо целое число от 1 до 7. Для шрифта можно также указывать относительный размер, сообщая атрибуту целое число со знаком (это может быть size="+1" или size="-2").

color — указывает цвет, которым будет выделен данный кусок текста. Цвета задаются в виде RGB-значения с шестнадцатеричной нотацией, либо выбирается один из стандартных цветов.

face — задает гарнитуру шрифта.

2.3.4 Специальные тэги HTML

Тэг `<ADDRESS>` используется для выделения автора документа и его адреса (e-mail). Синтаксис:

`<ADDRESS> Адрес-автора </ADDRESS>`

Некоторые символы являются управляющими символами в HTML и не могут напрямую использоваться в документе:

- левая угловая скобка "<"
- правая угловая скобка ">"
- амперсанд "&"
- двойные кавычки ""

Чтобы использовать данные символы в документе, необходимо заменить их escape-последовательностями: < — `<`; > — `>`; & — `&`; " — `"`;

Существует большое количество escape-последовательностей для обозначения специальных символов ("`©`" для обозначения знака © и "`®`" для значка ®). Одной из особенностей является замена символов во 2-ой части символьной таблицы (после 127-ого символа) на escape-последовательности для передачи текстовых файлов с национальными языками по 7-битным каналам.

Escape-последовательности чувствительны к регистру: НЕЛЬЗЯ использовать `<`; вместо `<`;

2.3.5 Ссылки

Гипертекстовые связи (ссылки) являются наиболее важным элементом Web-страниц. С их помощью делают документ связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно.

Ссылки могут указывать на другой документ, специальное место данного документа или выполнять другие функции, например, запрашивать файл по FTP-протоколу для отображения его браузером. URL (Universal Resource Locator) - универсальный способ адресации ресурсов в сети может указывать на специальное место по абсолютному пути доступа, или указывать на документ в текущем пути доступа, что часто используется при организации больших структурированных Web-сайтов. В URL после имени файла через # (диез) может указываться специальный маркер. Данный элемент является ссылкой на строку (точку) внутри HTML-документа /5/.

Ссылка состоит из двух частей. Первая из них — это то, что видно на Web-странице; она называется указатель ссылки (anchor). Вторая часть, дающая инструкцию браузеру, называется адресной частью ссылки (URL-адрес). Когда происходит щелчок мышью по указателю ссылки, браузер загружает документ, адрес которого дается URL-адресом. Ниже рассмотрены правила построения отдельных элементов ссылок. Вторая часть ссылки является URL – адрес. Это не что иное, как адрес Web – страницы, которая будет загружена при щелчке мышью на указателе.

Ссылка осуществляется тэгом <A>. Тэг <A> имеет единственный параметр HREF, значением которого является URL-адрес. Этот тэг является контейнером, поэтому необходимо поставить закрывающий тэг .

Текстовый указатель ссылки

Указатель может быть как относительным, так и абсолютным.

Относительные указатели удобны в использовании. Намного проще вставить только имя файла, а не весь длинный URL-адрес. Они также позволяют перемещать файлы в пределах сервера без больших изменений в межстраничной адресации.

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

Кроме ссылок на другие документы, часто бывает полезно включить ссылки на разные части текущего документа. Например, большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы.

Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Например, если необходимо сделать ссылку на текст определенной главы документа, нужно разместить там указатель и дать ему имя при помощи параметра NAME тэга <A>. При этом параметр HREF не используется, и браузер не выделяет содержимое тэга <A>.

Пример:

` `

Необходимо обратить внимание, что в приведенном примере отсутствует содержимое тэга `<A>`. Обычно именно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого, вместо указания в параметре HREF адреса документа, как это делалось ранее, поместим туда имя ссылки с префиксом #, говорящим о том, что это внутренняя ссылка.

Пример:

`Глава 5`

Теперь, если пользователь щелкнет кнопкой мыши на словах "Глава 5", браузер выведет соответствующую часть документа в окне просмотра.

Web-пространство является лишь частью сети Интернет. Другие ресурсы начали свое существование задолго до рождения WWW, поэтому накопили уже много достойной внимания информации и имеют достаточно большую аудиторию. Поэтому, разрабатывая свою персональную страницу или документ можно включить ссылки и на другие ресурсы.

Ресурсы Интернета весьма разнообразны по форме и содержанию. Хотя HTML предполагает возможность создания своих собственных версий этих ресурсов с помощью механизма обработки данных форм, есть более простые пути к взаимодействию с системами UseNet, Telnet, FTP, e-mail и другими. Например, можно создать документ с использованием различных тэгов форм, текстовых элементов и кнопкой для отправки электронного письма на необходимый адрес. Однако будет намного проще для связи указать лишь свой адрес электронной почты. В этом случае упрощается обновление страницы, которое не будет связано с изменениями форм. Кроме того, многие браузеры имеют встроенную поддержку некоторых ресурсов, что дает возможность сократить время на установление связи с ними.

Самой популярной деятельностью в Интернете является обмен электронными письмами. Пользователей этого ресурса намного больше, чем любого другого. Большинство современных программ для обмена электронными сообщениями имеют дружественный интерфейс и просты в использовании.

Создание ссылки на электронную почту так же просто, как и на другую страницу. Для этого вместо URL-адреса следует указать адрес электронной почты, предварив его словом mailto:.

Пример:

`Присылайте ваши отзывы`

Ссылки на другие ресурсы Интернета записываются аналогично.

Web – страница	-	http://sitename
e-mail	-	mailto:address
Newsgroup	-	news:newsgroupname
ftp	-	ftp://sitename
Gopher	-	gopher://sitename

WAIS	-	wais://sitename
Telnet	-	telnet://sitename

2.4 Контрольные вопросы

- 1) Что понимается под терминами закрывающий и открывающий тэги? Всегда ли они оба применяются?
- 2) Влияет ли регистр языка HTML на результат выполнения?
- 3) Каким тэгом можно внести комментарий в HTML?
- 4) Какой тэг позволяет изменить цвет фона HTML-документа?
- 5) Назначение тэга <HR>?
- 6) Назначение тэга <BIG>?
- 7) К каким изменениям приведёт применение тэга <ADDRESS>?
- 8) Как для построения внутренней ссылки создать указатель, определяющий место назначения внутри документа?
- 9) Привести пример ссылки на электронную почту.

2.5 Тесты

- 1) Цвет, который будет использоваться при выводе на экран текста из еще не выбранных гипертекстовых связей?
 - а) ALINK;
 - б) VLINK;
 - в) LINK.

- 2) Тэг, вызывающий переход на новую строку без разрыва абзаца?
 - а) <WBR>;
 - б) <HR>;
 - в) <P>;
 - г)
.

- 3) Тэг, предназначенный для обозначения в документе цитаты из другого источника?
 - а) <ABBR></ABBR>;
 - б) <PRE></PRE>;
 - в) <BLOCKQUOTE></BLOCKQUOTE>;
 - г) <ACRONYM></ACRONYM>.

- 4) Тэг, отображающий текст моноширинным шрифтом?

<STRIKE></STRIKE>;

 - а) <U></U>;
 - б) ;
 - в) <TT></TT>.

5) Сдвигает текст ниже уровня строки и выводит его шрифтом меньшего размера?

- a) <SMALL></SMALL>;
- б) <BLINK></BLINK>;
- в) <STRIKE></STRIKE>;
- г) .

6) Гиперссылки: создание?

- a) Присылайте ваши отзывы;
- б) ;
- в) ;

7) Выберите внутреннюю гиперссылки?

- a) ;
- б) ;
- в) ;

3 Лабораторная работа №3

Метаданные. Фреймы. Работа с FTP сервером

3.1 Цель работы

Познакомиться с назначением метаданных. Научиться использовать фреймы. Освоить работу с FTP-сервером.

3.2 Порядок выполнения работы

1) Открыть ранее созданный Web-документ. Проанализировать его содержимое. Выявить основные моменты, и подготовить информацию для заполнения метаданных. Используя описание, приведённое в методическом материале, заполнить метаданные страницы.

2) Создать Web-документ, состоящий из пяти или более фреймов. Для работы с фреймами необходимо будет создать или использовать не менее пяти страниц, каждая из которых будет в своей области отображения на главной странице.

Создать фреймовую структуру:

Фрейм 1	Фрейм 2
Фрейм 3	

Фрейм 1	Фрейм 2
	Фрейм 3

3) Сделать несколько ссылок, которые будут открывать какой-либо документ, и использовать в качестве области отображения вызванного документа, некоторый центральный (наибольший) фрейм. Провести эксперименты с параметром «TARGET».

4) Отредактировать свой Web-документ таким образом, чтобы у нескольких фреймов можно было поменять размеры при помощи мыши, а размеры остальных оставались фиксированными. Изменить толщину и цвет, рамок фреймов.

5) Подключиться к указанному FTP-серверу. Создать на нём свою папку, и скопировать в него ранее созданный Web-документ. Вывести Web-документ на просмотр через браузер.

К сдаче лабораторной предоставляются: работающая страница на сервере с комментариями каждого тэга и его свойств (атрибутов) в исходнике.

3.3 Методические указания

3.3.1 Метаданные

META - это наиболее популярный элемент разметки заголовка, более распространён только элемент TITLE. Такое положение дел объясняется назначением данного элемента разметки. META содержит управляющую информацию, которую браузер использует для правильного отображения и обработки содержания тела документа.

Впервые контейнер META был задействован при принудительной перезагрузке документа браузером через заголовок HTTP-сообщения. В заголовке HTTP-сообщения можно указать оператор refresh. Время, заданное как параметр этого оператора, определяет интервал в секундах, после которого браузер загружает документ, определённый атрибутом URL данного оператора. Впервые этот механизм был реализован на сервере CERN, но наибольшую популярность приобрёл при использовании сервера WN (Web-сервер, который был разработан для платформы Unix (Linux)).

В контейнере META подобный механизм реализуется следующим образом:

```
<META HTTP-EQUIV="Refresh" CONTENT="1; URL=refresh.html">
```

В данном случае через одну секунду после загрузки документа браузер должен инициализировать загрузку страницы refresh.html.

Используя этот механизм, можно построить автоматически перезагружаемую последовательность страниц. Для этого в заголовке каждой страницы из данной последовательности следует разместить соответствующий контейнер META.

```
<META HTTP-EQUIV="Refresh" CONTENT="1";  
URL="refreshX.html">
```

Заглавная буква "X" в слове "refreshX.html" — это цифра номера кадра. На странице нулевого кадра в этом месте следует указать на первый кадр (refresh1.html), на странице первого кадра — на второй (refresh2.html).

В Windows 95 и Windows NT 4.0 с поддержкой таблиц UNICODE появилась возможность указывать тип кодировки документа — CHARSET. К сожалению, на многих Unix-платформах этот механизм не работает, что часто приводит к ошибкам. Скептическое отношение поклонников Unix к этой возможности ничем не подкреплено, так как основная масса пользователей российской части Internet просматривает документы World Wide Web в Windows. Для перекодировки на стороне в заголовок документ необходимо включить META-тэг следующего вида:

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html;  
CHARSET=windows-1251 ">
```

Приведенный выше пример показывает, как используются операторы заголовка HTTP-сообщения. Однако здесь тоже следует быть осторожным. Большинство российских Web-узлов используют в качестве HTTP-сервера Russian Apache. Эта модификация сервера поддерживает перекодировку документов "на лету" для правильного отображения на стороне клиента. Russian Apache сам вставляет в HTTP-заголовок (не путать с HEAD) директиву Content-type. Если в документе будет META-элемент с указанием типа кодировки, а Apache перекодировал содержание, то возможно несоответствие между указанным в META типом кодировки и реальной кодировкой содержания документа.

Кроме Content-type, можно указать и другие операторы. Например, запретить кэширование документа. Необходимость в этом возникает при частом обновлении документа или наличии в нем изменяющихся SSI-вставок. Для запрета кэширования достаточно вставить в заголовок META - тэг вида:

```
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
```

Pragma — это наследие HTTP 1.0. В новой версии протокола HTTP (HTTP 1.1) управление кэшированием осуществляется через оператор Cache-Control. Для получения такого же эффекта, как в случае с Pragma, и в заголовке HTML-документа достаточно указать:

```
<META HTTP-EQUIV="Cache-Contror CONTENT="no-cache">
```

Новый механизм управления кэшированием и хранением документа на стороне клиента гораздо более гибок, чем в HTTP 1.0. Например, можно запретить хранение документа после пересылки:

```
<META HTTP-EQUIV="Cache-Control" CONTENT="no-store">
```

Точно так же можно задать время последней модификации (Last-Modified) или дату истечения актуальности документа (Expire).

С появлением роботов поисковых машин на META-тэг была возложена еще одна функция — описание поискового образа документа. Наиболее последовательно это было впервые реализовано в Webcrawler. До этого в качестве поискового образа документа использовался либо весь список слов документа, либо слова первого абзаца.

Собственно, для описания документа используется два META-тэга. Один определяет список ключевых слов, а второй — реферат (краткое содержание документа), который отображается в качестве пояснения к ссылке на документ в отчете поисковой машины о выполненном запросе. Контейнер TITLE здесь также используется в качестве названия документа.

Пример:

```
<TITLE>Создание Web – Страниц </TITLE>
```

```
<META NAME="description" HTTP-EQUIV="description"  
CONTENT="Лабораторный практикум Создание Web – Страниц.  
Тема: Заголовок HTML-документа. Элемент разметки META.  
Дается краткое описание основных способов применения  
контейнера META в заголовке HTML-документа. Рассматривается  
управление HTTP-обменом и индексирование документов.">
```

```
<META NAME="keywords" HTTP-EQUIV="keywords"  
CONTENT="учебный курс; Web-технология; web; технология;  
HTML;  
язык гипертекстовой разметки; заголовок HTML-документа;  
заголовок;HTML; документ; контейнер; META; элемент; HEAD;  
разметка; методичка">
```

При индексировании такого документа содержимое контейнера TITLE и атрибутов CONTENT контейнеров META после фильтрации попадет в индекс поисковой машины и может быть использовано для составления запросов. Процесс фильтрации отбракует так называемые stop-слова и общие слова. Они не попадут в индекс поисковой машины. В частности, будут отбракованы предлоги или, если речь идет о тематическом поисковом индексе, например по технологиям World Wide Web, то в него не попадут: web, Web-технология и т.п.

META - тэгом пользуются и программы подготовки документов. Они размещают в нем свой идентификатор. В общем случае контейнер META выглядит следующим образом:

```
<META [name=имя] [HTTP-EQUIV=имя_HTTP-оператора]  
CONTENT=текст>
```

Практика показывает, что при индексировании можно указывать одновременно и атрибут NAME, и атрибут HTTP-EQUIV с одинаковыми значениями. Это связано с тем, что одни роботы индексирования анализируют содержание META - элемента по атрибуту NAME, а другие — по атрибуту HTTP-EQUIV.

3.3.2 Фреймы

Фреймы — достаточно мощный механизм представления информации на Web-страницах. С помощью фреймов экран разделяется на несколько прокручивающихся областей, в каждой из которых отображается содержимое

отдельной страницы и даже отдельного Web-узла. Каждый фрейм, может иметь следующие свойства:

- Каждый фрейм имеет свой URL, что позволяет загружать его независимо от других фреймов.

- Каждый фрейм имеет собственное имя (параметр NAME), позволяющее переходить к нему из другого фрейма.

- Размер фрейма может быть изменен пользователем прямо на экране при помощи мыши (если это не запрещено указанием специального параметра).

Данные свойства фреймов позволяют создавать продвинутое интерфейсные решения, такие как:

- Размещение статической информации, которую автор считает необходимым постоянно показывать пользователю, в одном статическом фрейме. Это может быть графический логотип фирмы, авторское право, набор управляющих кнопок.

- Помещение в статическом фрейме оглавления всех или части Web-документов, содержащихся на Web-сервере, что позволяет пользователю быстро находить интересующую его информацию.

- Создание окна результатов запросов, когда в одном фрейме находится собственно запрос, а в другом результаты запроса.

- Создание формы типа "мастер-деталь" для Web-приложений, обслуживающих базы данных.

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тэга BODY используется контейнер FRAMESET, содержащий описание внутренних HTML-документов, содержащий собственно информацию, размещаемую во фреймах/5/.

```
<HTML>  
<HEAD>...</HEAD>  
<FRAMESET>...</FRAMESET>  
</HTML>
```

Однако, фрейм-документ является специфичным видом HTML-документа, поскольку не содержит элемента BODY и какой-либо информационной нагрузки соответственно. Он описывает только фреймы, которые будут содержать информацию (кроме случая двойного документа, который мы рассмотрим позже).

Общий синтаксис фреймов:

```
<FRAMESET COLS="value" | ROWS="value">  
<FRAME SRC="URL1">  
<FRAME SRC="URL2">  
...  
</FRAMESET>
```

Общий контейнер FRAMESET описывает все фреймы, на которые делится экран. Можно разделить экран на несколько вертикальных или горизонтальных фреймов. Внутри контейнера могут находиться только тэг <FRAME>, вложенные контейнеры <FRAMESET> и <NOFRAME>, который

позволяет строить двойные документы для браузеров, поддерживающих и не поддерживающих фреймы.

Данный тэг имеет два взаимоисключающих параметра: ROWS и COLS.

ROWS="список определений горизонтальных подокон". Данный тэг содержит описания некоторого количества подокон, разделенные запятыми. Каждое описание представляет собой числовое значение размера подокна в пикселях, процентах от всего размера окна или связанное масштабное значение. Количество подокон определяется количеством значений в списке. Общая сумма высот подокон должна составлять высоту всего окна (в любых измеряемых величинах). Отсутствие атрибута ROWS определяет один фрейм, величиной во все окно браузера.

COLS="список определений вертикальных подокон". То же самое, что и ROWS, но делит окно по вертикали, а не по горизонтали.

В качестве списка определений могут выступать:

1) Простое числовое значение определяет фиксированный размер подокна в пикселях. Это не самый лучший способ определения размера подокна, т.к. различные браузеры имеют различный размер рабочего поля, не говоря уже о различных экранных разрешениях у пользователя. Если все же используется данный способ описания размера, то настоятельно рекомендуется сочетать его с каким-либо другим, чтобы в результате точно получили 100 % - ное заполнение окна браузера.

2) Значение величины подокна в процентах от 1 до 100. Если общая сумма процентов описываемых подокон меньше или превышает 100, то размеры всех фреймов пропорционально увеличиваются или уменьшаются до суммы 100 %.

3) Число со звездочкой. Вообще говоря, числовое значение в данном описании является необязательным. Символ "*" указывает на то, что все оставшееся место будет принадлежать данному фрейму. Если указывается два или более фрейма с описанием "*", или "*,*" то оставшееся пространство делится поровну между этими фреймами. Если перед звездочкой стоит цифра, то она указывает пропорцию для данного фрейма (во сколько раз он будет больше аналогично описанного чистой звездочкой). Описание "3*,*," говорит, что будет создано три фрейма с размерами 3/5 свободного пространства для первого фрейма и по 1/5 для двух других.

Примеры:

<FRAMESET COLS="50*,50"> — описывает три фрейма, два по 50 точек справа и слева, и один внутри этих полосок.

<FRAMESET ROWS="20%,3*,*"> — описывает три фрейма, первый из которых занимает 20 % площади сверху экрана, второй 3/4 оставшегося от первого фрейма места (60 % всей площади окна), а последний 1/4 (20 % всей площади окна).

<FRAMESET ROWS="*,60%,*"> — аналогично предыдущему примеру.

Тэги <FRAMESET> могут быть вложенными, т.е. например:

<FRAMESET ROWS="50%,50%">

```
<FRAMESET COLS="*,*"
</FRAMESET>
```

```
</FRAMESET>
```

Тэг FRAME описывает каждый фрейм в отдельности. Рассмотрим более детально каждый компонент.

```
<FRAME SRC="URL" NAME="frame_name" MARGINWIDTH="n1"
MARGINHEIGHT="n2" SCROLLING=yes|no|auto NORESIZE>
```

Где SRC="URL" — описывает URL документа, который будет отображен внутри данного фрейма. Если он отсутствует, то будет отображен пустой фрейм. В качестве URL допустимо использование не только HTML-, но и GIF- или JPEG-файла.

NAME="frame_name" — описывает имя фрейма. Имя фрейма может быть использовано для определения действия с данным фреймом из другого HTML-документа или фрейма (как правило, из соседнего фрейма этого же документа). Имя обязательно должно начинаться с символа. Содержимое поименованных фреймов может быть задействовано из других документов при помощи специального атрибута TARGET, описываемого ниже.

MARGINWIDTH="n1" или MARGINHEIGHT="n2" — используются, если автор документа хочет указать величину вертикальных или горизонтальных разделительных полос между фреймами сбоку. Значения n1 и n2 указываются в пикселах и не могут быть меньше единицы.

SCROLLING="yes | no | auto" — позволяет задавать наличие полос прокрутки у фрейма. Параметр yes указывает на обязательное присутствие у фрейма полос прокрутки, параметр no — на отсутствие, auto — определяет наличие полос прокрутки только при их необходимости (значение по умолчанию).

NORESIZE — отменяет возможность изменить при помощи мыши размер фрейма так же, как и размер окна Windows. Если у одного фрейма установлен атрибут NORESIZE, то у соседних фреймов тоже не может быть изменен размер со стороны данного.

BORDER — управляет толщиной рамки.

BORDERCOLOR — задает цвет рамки.

FRAMEBORDER=no — отключает отображение рамки.

NOFRAMES используется в случае, если создается документ, который может просматриваться браузерами, как поддерживающими, так и не поддерживающими фреймы. Данный тэг помещается внутри контейнера FRAMESET, а все, что находится внутри тэгов <NOFRAMES> и </NOFRAMES> игнорируется браузерами, поддерживающими фреймы.

Пример: Рассмотрим реализацию фреймов для подобного разбиения окна: <FRAMESET ROWS="*,*">

```
<NOFRAMES>
```

```
<H1>Пример использования фреймов!</H1>
```

```
</NOFRAMES>
```

```
<FRAMESET COLS="65%,35%">
```

```
<FRAME SRC="link1.html">
```

```

<FRAME SRC="link2.html">
</FRAMESET>
<FRAMESET COLS="*,40%,*">
<FRAME SRC="link3.html">
<FRAME SRC="link4.html">
<FRAME SRC="link5.html">
</FRAMESET>
</FRAMESET>

```

В браузере это будет выглядеть следующим образом:

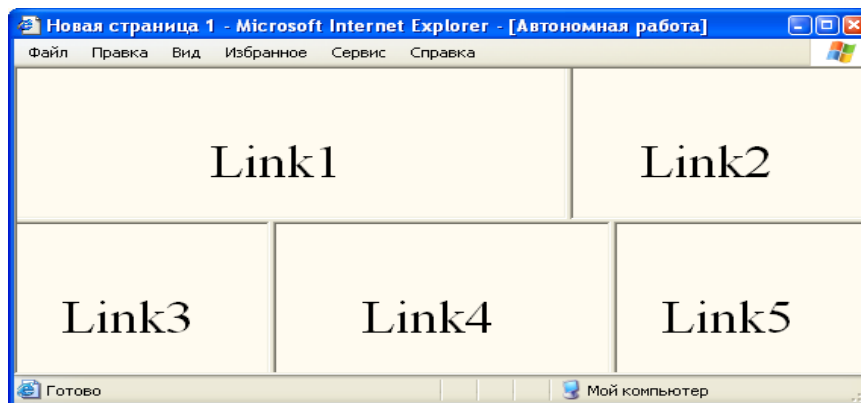


Рисунок 10 – Фреймы

С появлением фреймов сразу возникает вопрос: "А как сделать так, чтобы, нажимая на ссылку в одном фрейме инициировать появление информации в другом?"

Ответом на данный вопрос является планирование взаимодействия фреймов. Каждый фрейм может иметь собственное имя, определяемое параметром NAME при описании данного фрейма. Существует, также, специальный атрибут — TARGET, позволяющий определять, к какому фрейму относится та или иная операция. Формат данного атрибута следующий:

```
TARGET="windows_name"
```

Данный атрибут может встречаться внутри различных тэгов:

TARGET в тэге A. Это самое прямое использование TARGET. Обычно, при активизации пользователем ссылки соответствующий документ появляется в том же окне (или фрейме), что и исходный, в котором была ссылка. Добавление атрибута TARGET позволяет произвести вывод документа в другой фрейм.

Пример:

```
<A HREF="mydoc.html" TARGET="Frame1"> фрейм №1 </A>
```

TARGET в тэге BASE. Размещение TARGET в тэге BASE позволит не указывать при описании каждой ссылки фрейм-приемник документов, вызываемых по ссылкам. Это очень удобно, если в одном фрейме находится меню, а в другой выводится информация.

Пример:

Документ №1.

```
<FRAMESET ROWS="20,*">
```

```
<FRAME SRC="doc2.html" NAME="Frame1">
<FRAME SRC="doc3.html" NAME="Frame2">
</FRAMESET>
```

Документ №2 (doc2.html).

```
<HTML>
<HEAD>
  <BASE TARGET="Frame2">
</HEAD>
<BODY>
  <A HREF="url1"> Первая часть</A> |
  <A HREF="url2"> Вторая часть</A>
</BODY>
</HTML>
```

Также можно включать тэг TARGET в описание ссылки при создании карты изображения.

Пример:

```
<AREA SHAPE="circle" COORDS="100, 100, 50"
  HREF="http://www.softexpress.com" TARGET="Frame1">
```

TARGET в тэге FORM. То же относится и к определению формы.

В данном случае, после обработки переданных параметров формы результирующий документ появится в указанном фрейме.

```
<FORM ACTION="url" TARGET="window_name">
```

Зарезервированные имена фреймов служат для разрешения специальных ситуаций. Все они начинаются со знака подчеркивания. Любые другие имена фреймов, начинающиеся с подчеркивания, будут игнорироваться браузером.

TARGET="_blank". Данное значение определяет, что документ, полученный по ссылке, будет отображаться в новом окне браузера.

TARGET="_self". Данное значение определяет, что документ, полученный по ссылке, будет отображаться в том же фрейме, в котором находится ссылка. Это имя удобно для переопределения окна назначения, указанного ранее в тэге BASE.

TARGET="_parent". Данное значение определяет, что документ, полученный по ссылке, будет отображаться в родительском окне, вне зависимости от параметров FRAMESET. Если родительского окна нет, то данное имя аналогично "_self".

TARGET="_top". Данное значение определяет, что документ, полученный по ссылке будет отображаться на всей поверхности окна, вне зависимости от наличия фреймов. Использование данного параметра удобно в случае вложенных фреймов.

Microsoft Internet Explorer поддерживает и так называемые плавающие фреймы. Они могут появляться в любом месте экрана. Текст, расположенный на главной странице, обтекает этот фрейм.

Пример:

```
<IFRAME ALIGN=RIGHT WIDTH=50% HEIGHT=300  
SRC="FILE.HTMI">  
</IFRAME>
```

Реализация работы этого фрагмента программы в браузере представлена на рисунке 11.

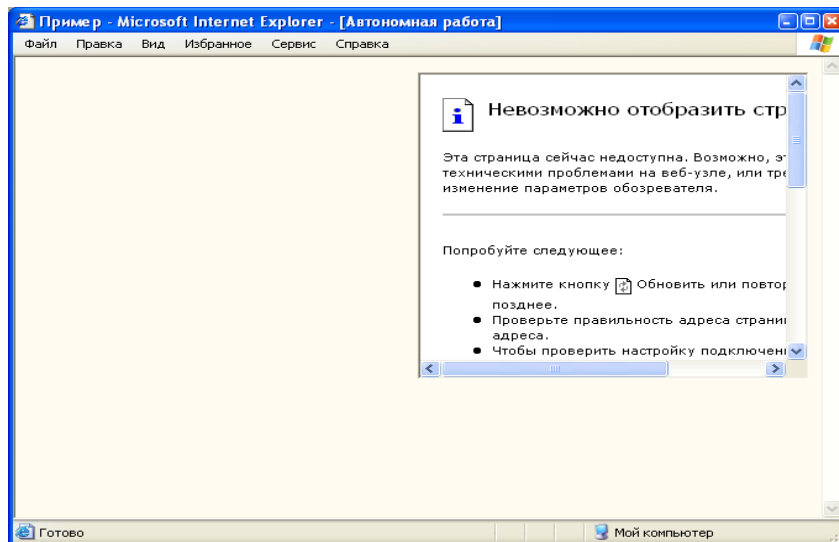


Рисунок 11 – Плавающий фрейм

3.3.3 FTP

Один из протоколов TCP/IP, используемый для копирования файлов с одного компьютера на другой через Интернет. При этом оба компьютера должны поддерживать соответствующие роли FTP: один должен быть клиентом FTP, а другой — сервером FTP.

FTP позволяет передавать файлы с компьютера на компьютер с помощью программ, поддерживающих протокол FTP (File Transfer Protocol). Данная команда может использоваться как в интерактивном, так и в пакетном режимах. В последнем случае она позволяет обрабатывать текстовые файлы в формате ASCII.

Синтаксис команды для подключения к FTP серверу:

ftp://пользователь:пароль@сервер:порт/папка

Где

"сервер" обязателен, префикс "ftp://" также обязателен при подключении из командной строки и может быть пропущен при подключении из панели серверов. Все прочие части имени опциональны. Если имя пользователя не указано, то будет использовано "anonymous". Если пропущен пароль, то на сервер будет послан пароль по умолчанию, заданный в конфигурации FTP-клиента.

FTP-клиент позволяет работать с файлами FTP-сервера аналогично файлам на локальных дисках. Он поддерживает копирование, перенос и

удаление файлов и папок, создание папок, просмотр и редактирование файлов. Кроме того, можно возобновить оборванную пересылку файла, воспользовавшись кнопкой "Возобновить" при запросе перезаписи файла, но эта функция должна поддерживаться и FTP-сервером. Также можно использовать команду "Поиск файла" для поиска требуемых файлов на FTP-сервере.

Примеры:

ftp://POVT-S1

ftp://ivanov.hotmail.ru:1234567@ftp.newmail.ru

3.4 Контрольные вопросы

- 1) Понятие метаданных.
- 2) Варианты использования метаданных.
- 3) Понятие фрейма.
- 4) Синтаксис описания
- 5) Как указать фрейм в качестве области отображения для ссылки на какой либо ресурс.
- 6) Возможные значения параметра «TARGET».
- 7) Как изменить толщину и цвет, рамки фрейма.

3.5 Тесты

- 1) Какие из перечисленных контейнеров размещают в заголовке документа?
 - а) <LINK HREF="..." REL="...">;
 - б) <BODY BGCOLOR="..." TEXT="...">;
 - в) <SCRIPT LANGUAGE="..." SRC="...">;
 - г) ;
 - д) <META NAME="..." CONTENT="...">.
- 2) Есть ли у параметра HEAD дополнительные атрибуты?
 - а) атрибутов нет;
 - б) есть атрибут PROFILE;
 - в) в DTD атрибуты не определены.
- 3) С помощью какого контейнера задается фреймовая структура документа?
 - а) FRAME;
 - б) BODY;
 - в) FRAMESET;
 - г) IFRAME.

4) Какие атрибуты надо указать тэгу <A> чтобы при нажатии на ссылку в одном фрейме, документ загружался в другом фрейме?

- а) надо указать в атрибуте TARGET имя требуемого фрейма;
- б) никаких дополнительных атрибутов не требуется;
- в) необходимо указать TARGET=top_.

5) Как указывается страница для отображения в кадре?

- а) с помощью атрибута HREF тэга FRAMESET;
- б) с помощью атрибута LOCATION тэга BODY;
- в) с помощью атрибута SRC тэга FRAME.

4 Лабораторная работа №4

Мультимедиа и графика. Создание табличных данных. Списки

4.1 Цель работы

Научиться применять мультимедиа, графику, таблицы и списки в HTML-документах, для улучшения их внешнего вида.

4.2 Порядок выполнения работы

1) Открыть созданный ранее Web-документ. Подобрать подходящие мультимедиа-файлы.

2) В данной работе необходимо в Web-документ вставить несколько подходящих по смыслу музыкальных фрагментов в разных форматах (.Wav, .Mid), озвучить фон странички каким-либо файлом в формате Mid. Подключить один AVI-файл, например файл, находящийся в папке с Delphi: "C:\Program Files\Borland\Delphi6\Demos\CoolStuf\cool.avi". Данный файл подходит для наших целей, так как его размер не превышает 35 килобайт.

3) Вставить графические образы в Web - документ. Вначале оформить фон документа таким образом, чтобы графический файл был уложен на фоновой плоскости черепицей и оставался не подвижным при прокручивании документа.

4) Добавить в документ несколько графических файлов в качестве иллюстраций для улучшения внешнего вида. Необходимо помнить, что слишком большие графические файлы будут долго открывать на компьютере пользователя.

5) Создать несколько ссылок на внутренние или внешние ресурсы с использованием графических файлов.

6) Добавить на Web-страницу таблицу, содержащую элементы меню в виде списков, а в качестве элементов списков должны использоваться гиперссылки.

7) Изменить внешний вид таблицы и списков. Изменить рамку таблицы, её цвет толщину видимость и т.д., провести исследование всех основных атрибутов таблицы.

8) Создать таблицу вида:

Файл 1	Файл 3
Файл 2	Файл 4
Изображение	

К сдаче лабораторной предоставляются: работающая страница на сервере с комментариями каждого тэга и его свойств (атрибутов) в исходнике.

4.3 Методические указания

4.3.1 Мультимедиа

Начнём со вставки звука и видео на страницу.

Синтаксис тэга:

```
<EMBED src="File" width="n1" height="n2" controls="smallconsole"
autostart="false" loop="true" play_loop="2" hidden="true"></EMBED>
```

Где

SRC="File" – MIDI (*.mid), WAV (*.wav) или AVI (*.avi - видео) файл.

WIDTH="n1" HEIGHT="n2" - ширину (width) и высоту (height) пульта управления.

CONTROLS="smallconsole" - при использовании этого атрибута пульт становится маленьким.

AUTOSTART="false" - надо ли проигрывать снова видео или музыкальный ролик (false - нет, true - да).

LOOP="true" - повторять ролик с самого начала (false - нет, true - да).

PLAY_LOOP="3" - если да, то сколько раз (вместо 3 поставить кол-во раз).

HIDDEN="true" - спрятать пульт ролика (false - нет, true - да).

Но можно, и даже лучше ставить просто ссылку на видео файл, так как браузер сам запускает плагины для их воспроизведения, и отпадает необходимость во вставке выше перечисленного HTML кода.

Пример:

```
<A HREF="VideoFail.avi">Мой видео файл</A>
```

А если файл находится не на разрабатываемой странице, то нужно прописать весь адрес файла.

Пример:

Мой видео файл

RealAudio и RealVideo - это такая программа, которая может воспроизводить музыку в online или файлы с музыкой или видео в Real формате.

Вначале необходимо открыть программу RealEncoder (до этого надо её установить), заполнить там строки Title (Название), Author (Автор), Copyright (Права на это принадлежат...). Потом выбрать справа качество записи (лучше брать Audio 28.800 Voice для аудио файла и Video 28.800 High Action для видео файла, но можно и другие). Затем открыть в меню File => Open session. Там нужно выбрать File и нажать кнопку Add и выбрать свой аудио или видео файл. Затем нажать кнопку "Save as..." и сохранить его под любым именем. Выполнив всё это, нажать "OK", внизу загораются кнопка Start -необходимо нажать её. Вот и сделали Real файл. Теперь осталось разместить его на своей странице, сделав просто ссылку на файл.

Также можно сделать так, чтобы музыку слушали люди в ONLINE режиме. Нужно взять любой текстовый редактор (Блокнот для Windows, к примеру) и написать адрес Real файла в Интернете, но заменив http на pnm. И получится что-то похожее на это:

Пример:

pnm://www.domain.com/~vasya/arhiv/3pesnya.ra

И сохранить в формате *.ram. Ну и надо теперь просто дать ссылку на ram файл, и он будет брать файл с музыкой или видео и показывать его через RealPlayer.

Простейший способ озвучить фон странички - добавить в тело создаваемого документа тэг вида:

Пример:

<BGSOUND LOOP="1" SRC="file.mid"></BGSOUND>

Тэг имеет всего два атрибута: LOOP задает число повторений проигрывания пьесы, (LOOP=0 позволяет мелодии звучать постоянно).SRC - файл-источник.

Подобные элементы при хорошем подборе музыкального файла могут создавать оригинальную атмосферу сайта.

4.3.2 Графика и анимация

Одним из способов использования графики в HTML является внедрение графических образов в документ, что позволяет пользователю непосредственно увидеть изображения. Такая техника проектирования документов называется "online image".

Синтаксис тэга:

Где

URL — адрес, указывающий браузеру, где находится рисунок. Рисунок должен храниться в графическом формате, поддерживаемом браузером.

ALT="text" задает текст, который будет отображен браузером, не поддерживающим отображение графики или с отключенной подкачкой изображений.

HEIGHT=n1 используется для указания высоты рисунка в пикселях. Если данный параметр не указан, то используется оригинальная высота рисунка. Это параметр позволяет сжимать или растягивать изображения по вертикали, что позволяет более четко определять внешний вид документа.

WIDTH=n2 позволяет задать абсолютную ширину рисунка в пикселях.

ALIGN используется, чтобы сообщить браузеру, куда поместить следующий блок текста. Это позволяет более строго задать расположение элементов на экране. Если данный параметр не используется, то большинство браузеров располагает изображение в левой части экрана, а текст справа от него.

BORDER=n позволяет автору определить ширину рамки вокруг рисунка (n=0..9).

VSPACE=m1 позволяет установить размер в пикселях пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

HSPACE=m2 выполняет то же самое, что и VSPACE, но только по горизонтали.

ISMAP сообщает браузеру, что данное изображение позволяет пользователю выполнять какие-либо действия, щелкая мышью на определенном месте изображения. Данная возможность является расширением HTML и в круг вопросов, освещаемых в данной пособии, не входит.

Большинство браузеров позволяет включать в документ фоновый рисунок. Некоторые пользователи любят фоновую графику, некоторые нет. Ненавязчивый полупрозрачный рисунок (обои) обычно хорошо выглядит в качестве фона для большинства документов.

Описание фонового рисунка включается в тэг BODY, графический файл должен быть формата GIF или JPG. Формат тэга выглядит следующим образом:

<BODY BACKGROUND="picture.gif">

Тэг <EMBED> аналогичен , но позволяет включать в Web-страницу файлы произвольного типа. Для них требуется подключение дополнительных модулей (plug-in).

Кроме обычных GIF-файлов, содержащих статические изображения, язык HTML дает возможность включать в документы динамические GIF-файлы.

Бегущая строка. Синтаксис тэга:

<MARQUEE BEHAVIOR=alternate|slide GCOLOR=...

WIDTH=... HEIGHT=...> текст </MARQUEE>

Подключение видеоизображений (AVI- и MPG-файлы).

Графические изображения могут использоваться не только в качестве иллюстрации, но и выполнять роль указателей гипертекстовых связей для обеспечения работы изображения в качестве ссылки на другие ресурсы достаточно включить изображение внутри тэга-контейнера <A>

Пример:

4.3.3 Таблицы

Таблицы в HTML организуются как набор столбцов и строк. Ячейки таблицы могут содержать любые HTML-элементы, такие, как заголовки, списки, абзацы, фигуры, графику, а также элементы форм.

Синтаксис определения таблицы в общем, виде:

```
<TABLE BORDER=... WIDTH=... >  
<TR>  
<TD параметры =... > 1-я клетка 1-ой строки </TD>  
<TD параметры =... > 2-я клетка 1-ой строки </TD>  
</TR>  
<TR>  
<TD> 1-я клетка 2-ой строки </TD>  
...  
</TR>  
...  
</TABLE>
```

Где таблица: <TABLE>...</TABLE>. Это основные тэги, описывающие таблицу. Все элементы таблицы должны находиться внутри этих двух тэгов. По умолчанию таблица не имеет обрамления и разделителей. Обрамление добавляется атрибутом BORDER.

Строка таблицы: <TR>...</TR>. Количество строк таблицы определяется количеством встречающихся пар тэгов <TR>..</TR>. Строки могут иметь атрибуты ALIGN и VALIGN, которые описывают визуальное положение содержимого строк в таблице.

Ячейка таблицы: <TD>...</TD>. Описывает стандартную ячейку таблицы. Ячейка таблицы может быть описана только внутри строки таблицы. Каждая ячейка должна быть пронумерована номером колонки, для которой она описывается. Если в строке отсутствует одна или несколько ячеек для некоторых колонок, то браузер отображает, пустую ячейку. Расположение данных в ячейке по умолчанию определяется атрибутами ALIGN=left и VALIGN=middle. Данное расположение может быть исправлено как на уровне описания строки, так и на уровне описания ячейки.

Заголовок таблицы: <TH>...</TH>. Ячейка заголовка таблицы имеет ширину всей таблицы; текст в данной ячейке имеет атрибут BOLD и ALIGN=center.

Подпись: <CAPTION>...</CAPTION>. Данный тэг описывает название таблицы (подпись). Тэг <CAPTION> должен присутствовать внутри <TABLE>...</TABLE>, но снаружи описания какой-либо строки или ячейки. По умолчанию <CAPTION> имеет атрибут ALIGN=top, но может быть явно установлен в ALIGN=bottom. ALIGN определяет, где будет поставлена подпись (сверху или снизу таблицы). Подпись всегда центрирована в рамках ширины таблицы. Основные атрибуты таблицы приведены ниже.

BORDER — используется в тэге TABLE. Если данный атрибут присутствует, граница таблицы прорисовывается для всех ячеек и для таблицы в целом. BORDER может принимать числовое значение, определяющее ширину границы, например BORDER=3.

ALIGN. Если атрибут ALIGN присутствует внутри тэгов <CAPTION> и </CAPTION>, то он определяет положение подписи для таблицы (сверху или снизу). По умолчанию ALIGN=top. Если атрибут ALIGN встречается внутри <TR>, <TH> или <TD>, он управляет положением данных в ячейках по горизонтали. Может принимать значения left (слева), right (справа) или center (по центру).

VALIGN — встречается внутри тэгов <TR>, <TH> и <TD>. Он определяет вертикальное размещение данных в ячейках. Может принимать значения top (вверху), bottom (внизу), middle (по середине) и baseline (все ячейки строки прижаты кверху).

NOWRAP — говорит о том, что данные в ячейке не могут логически разбиваться на несколько строк и должны быть представлены одной строкой.

COLSPAN — указывает, какое количество ячеек будет объединено по горизонтали для указанной ячейки. По умолчанию — 1.

ROWSPAN — указывает, какое количество ячеек будет объединено по вертикали для указанной ячейки. По умолчанию — 1.

COLSPEC — позволяет задавать фиксированную ширину колонок либо в символах, либо в процентах, например COLSPEC="20 %".

CELLSPACING — задает расстояние между ячейками таблицы (по умолчанию 2 пиксела).

CELLPADDING — определяет расстояние между рамкой ячейки и ее содержимым (по умолчанию 1 пиксел).

WIDTH — задает ширину таблицы либо в абсолютных единицах, либо в процентах относительно размера экрана. Используя внутри тэга <TD>, можно указывать ширину ячейки.

HEIGHT — то же, что и WIDTH, но определяет высоту таблицы.

FRAME — позволяет описывать внешние рамки таблицы. Может принимать следующие значения:

- VOID — нет рамки;
- ABOVE — отображает внешнюю часть рамки;
- BELOW — отображает нижнюю часть рамки;

- HSIDES — отображает верхнюю и нижнюю части рамки;
- LHS — отображает левую часть рамки;
- RHS — отображает правую часть рамки;
- VSIDES — отображает левую и правую части рамки;
- BOX или BORDER — отображает все части рамки.
- RULES — описывает рамки внутри таблицы. Может принимать следующие значения:
 - NONE — нет рамок;
 - GROUPS — отображает горизонтальные части рамки между группами таблицы;
 - ROWS — отображает горизонтальные части рамки внутри таблицы;
 - COLS — отображает вертикальные части рамки внутри таблицы;
 - ALL — отображает все части рамки внутри таблицы;
 - BGCOLOR — задает цвет фона таблицы;
 - BORDERCOLOR — задает цвет рамки;
 - BORDERCOLORLIGHT — задает цвет светлой части трехмерной рамки;
 - BORDERCOLORDARK — задает цвет темной части трехмерной рамки.

Верхние и нижние колонтитулы таблицы задаются контейнерами <THEAD>, <TFOOT>.

Иногда требуется поместить в таблицу пустые ячейки. Это можно сделать, поместив в ячейку пробел (), неразрывный пробел () или тэг
.

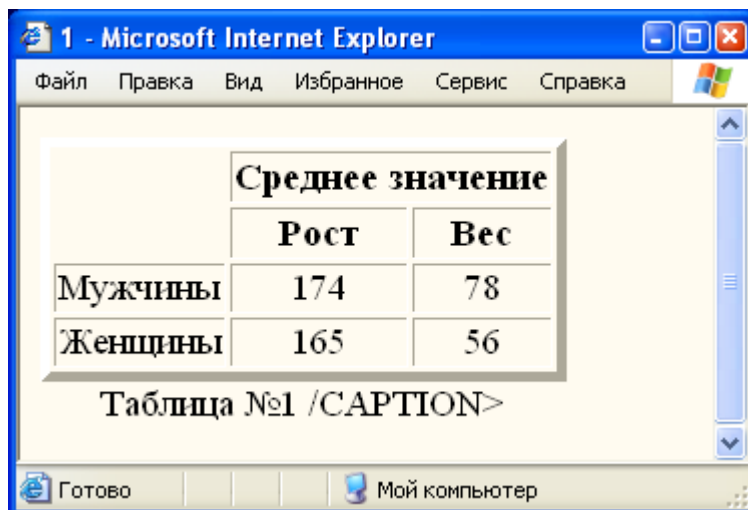
Пример:

```

<TABLE BORDER=5>
<CAPTION ALIGN=bottom> Таблица №1 /CAPTION>
<TR>
<TD ROWSPAN=2></TD>
<TH COLSPAN=2>Среднее значение</TH>
</TR>
<TR>
<TH>Рост</TH>
<TH>Вес</TH>
</TR>
<TR>
<TD>Мужчины</TD>
<TD ALIGN=center>174</TD>
<TD ALIGN=center>78</TD>
</TR>
<TR>
<TD>Женщины</TD>
<TD ALIGN=center>165</TD>
<TD ALIGN=center>56</TD>
</TR>
</TABLE>

```

В браузере это будет выглядеть следующим образом:



The screenshot shows a window titled "1 - Microsoft Internet Explorer" with a menu bar (Файл, Правка, Вид, Избранное, Сервис, Справка) and a taskbar at the bottom. The main content area displays a table with the following data:

Среднее значение		
	Рост	Вес
Мужчины	174	78
Женщины	165	56

Below the table, the text "Таблица №1 /CAPTION>" is visible. The taskbar shows "Готово" and "Мой компьютер".

Рисунок 12 – Таблица

4.3.4 Списки

Существует три основных вида списков в HTML-документе: нумерованный, ненумерованный, список определений.

Можно создавать вложенные списки, используя различные тэги списков или повторяя одни внутри других. Для этого просто необходимо разместить одну пару тэгов (стартовый и завершающий) внутри другой. Будут ли элементы вложенного списка иметь те же маркеры, обозначающие элемент списка — зависит от браузера.

Для создания заголовка списка используется контейнер <LN>.

В нумерованном списке браузер автоматически вставляет номера элементов по порядку. Если удалить произвольное количество элементов нумерованного списка, то остальные номера автоматически будут пересчитаны.

Нумерованный список заключается в контейнер . Каждый элемент списка начинается с тэга .

Пример:

 Первый пункт списка

 Второй пункт списка

 ...

Тэг может иметь параметры:

<OL TYPE=A|a|I|i|1 START=n>

Где

TYPE — вид счетчика: A — большие латинские буквы (A,B,C...); a — маленькие латинские буквы (a,b,c...); I — большие римские цифры (I,II,III...); i — маленькие римские цифры (i,ii,iii...); 1 — обычные цифры (1,2,3...).
Используется по умолчанию.

START=n — число, с которого начинается отсчет.

Для нумерованных списков браузер обычно использует маркеры для пометки элемента списка. Вид маркера, как правило, настраивает пользователь браузера.

Ненумерованный список заключается в контейнер . Как и в случае нумерованного списка, каждый элемент начинается с тэга .

Пример:

```
<UL>
<LI> Первый пункт списка
<LI> Второй пункт списка
<LI> ...
</UL>
```

Тэг может иметь параметр:

```
<UL TYPE=disc|circle|square>
```

Тип тэга определяет внешний вид маркера как вид по умолчанию (disc), круглый (circle) или квадратный (square).

Список определений заключается в контейнер <DL>. Список состоит из двух частей: термина и его описания. Каждый термин начинается тэгом <DT>, а описание — тэгом <DD>. **Пример:**

```
<DL>
<DT> Отдел маркетинга
<DD> Данный отдел занимается продвижением продуктов и услуг
<DT> Финансовый отдел
<DD> Данный отдел занимается всеми финансовыми операциями
<DT> Отдел кадров
<DD> Данный отдел занимается учетом и набором новых
сотрудников фирмы, распределением отпусков, отслеживанием
больничных листов.
```

```
</DL>
```

Пример:

```
<ol>
  <li><p style="margin-top: 0; margin-bottom: 0">132123213213</p></li>
  <li><p style="margin-top: 0; margin-bottom: 0">2132132131</p></li>
</ol><ol type="I">
  <li><p style="margin-top: 0; margin-bottom: 0">132123213213</p></li>
  <li><p style="margin-top: 0; margin-bottom: 0">2132132131</p></li>
</ol><ol type="A">
  <li><p style="margin-top: 0; margin-bottom: 0">132123213213</p></li>
  <li><p style="margin-top: 0; margin-bottom: 0">2132132131</p></li>
</ol><ul>
  <li><p style="margin-top: 0; margin-bottom: 0">23213213123</p></li>
  <li><p style="margin-top: 0; margin-bottom: 0">123213123</p></li>
</ul>
```

В браузере это будет выглядеть следующим образом:

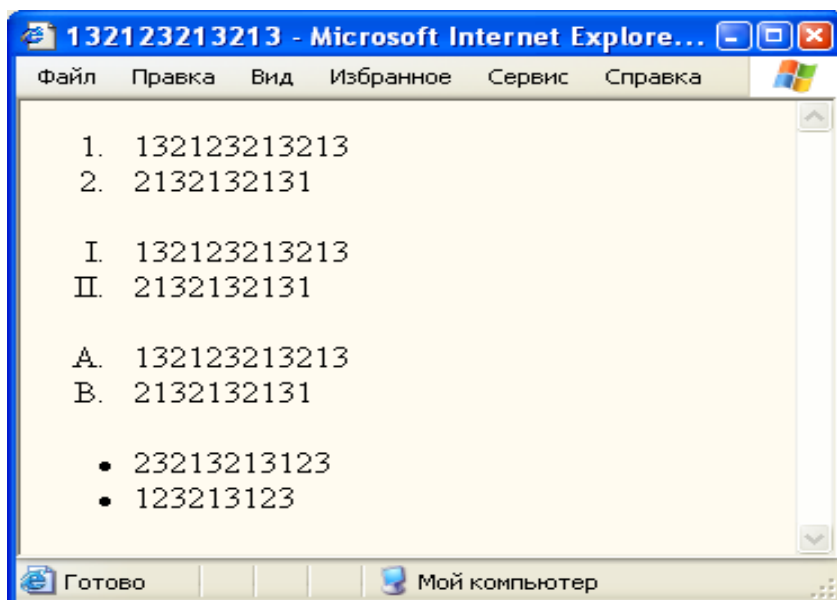


Рисунок 13 – Списки

4.4 Контрольные вопросы

- 1) Каким образом указать конкретное число повторений мультимедиа файлов;
- 2) Как указать программе просмотра проиграть файл автоматически;
- 3) Как можно проигрывать мультимедиа в в ONLINE режиме;
- 4) Как указать текст, который будет отображен браузером, не поддерживающим отображение графики;
- 5) Каким образом запретить разбиение данных в ячейке таблицы, т.е. представить единой строкой
- 6) Как сделать, чтобы в таблице отсутствовали рамки;
- 7) Как указать для списка номер первого пункта;
- 8) Как задать вид счетчика для списков, или указать внешний вид маркера.

4.5 Тесты

- 1) Какие форматы графических файлов наиболее часто применяются в HTML?
 - а) BMP;
 - б) JPG;
 - в) PNG;
 - г) GIF.
- 2) Какие основные значения атрибута ALIGN используются для определения положения изображения относительно окружающего текста?
 - а) Left;

- б) Right;
- в) Top;
- г) Bottom;
- д) Baseline.

3) В каких тэгах возможно применение картинок?

- а) IMG;
- б) INPUT;
- в) TABLE;
- г) BODY.

4) Что будет отображать браузер не поддерживающий изображения, или у которого выключено отображение графики?

- а) ничего;
- б) выделит место под картинку;
- в) будет ALT тэга IMG.

5) Что происходит, если у изображения не указаны атрибуты WIDTH, HEIGHT?

- а) не отображается;
- б) отображается в исходном размере картинки;
- в) показывается в виде иконки.

6) С помощью каких тэгов описывается таблица?

- а) <TABLE>;
- б) <TR>;
- в) ;
- г) <TD>;
- д) <BODY> .

7) Что определяет атрибут BORDER у элемента разметки TABLE?

- а) расстояние между ячейками;
- б) расстояние от содержания до границы;
- в) ширину ячейки;
- г) ширину границы;
- д) вид границы.

5 Лабораторная работа №5

Карты - изображения. Формы. CGI

5.1 Цель работы

Научиться разрабатывать Web-ресурс, содержащий карты изображения и формы.

5.2 Порядок выполнения работы

1) Создать новый Web-документ, который будет картой вашего ранее созданного Web-ресурса. На данном документе разместить какой-либо графический файл, который будет основой для создания будущей карты-изображения.

2) Разбить условное изображение на несколько зон, каждая из которых будет в дальнейшем играть роль гиперссылки.

3) Описать каждую зону карты-изображения через тэги, используя различные формы области изображений.

4) Описать 1-2 области изображения, таким образом, чтобы оставшаяся часть изображения не реагировала на действия пользователя.

5) Создать страницу с элементами формы. При этом использовать по возможности все описанные в методическом указании, элементы управления.

6) Провести исследование элементов управления формы.

К сдаче лабораторной предоставляются: работающая страница на сервере с комментариями каждого тэга и его свойств (атрибутов) в исходнике.

5.3 Методические указания

5.3.1 Карты – изображения

В последнее время многие Web – страницы для организации ссылок используют так называемые карты изображения. Реализация этой возможности предусмотрена языком HTML и позволяет привязать гипертекстовые ссылки к различным областям изображения. Такой подход нагляднее, чем применение обыкновенных текстовых связей, поскольку пользователь может не читать словесное описание связи, а сразу понять её смысл по графическому образу.

Карта- изображение фактически представляет собой обычное встроенное графическое изображение на Web – странице. Эти изображения могут иметь

любой допустимый формат (GIF или JPG). При этом в формате GIF может использоваться прозрачный цвет, а также режим чередования строк. Для того чтобы изображение могло использоваться в качестве опорного для карты – изображения, формально не накладывается никаких дополнительных ограничений.

Концепция карты-изображения на Web-страницах может быть реализована в двух различных вариантах — серверный вариант (server-side imagemap) и клиентский вариант (client-side imagemap). Последнее название часто используют в виде аббревиатуры CSIM. Исторически первым появился и получил распространение серверный вариант карт-изображений, который впервые был реализован в браузере Mosaic. Серверный вариант позволяли использовать первые версии всех трех ведущих браузеров. Серверный вариант может быть реализован в двух различных форматах, которые получили свое наименование по названиям организаций-разработчиков — NCSA и CERN.

В последнее время все большее развитие получает клиентский вариант, который впервые был реализован в браузере Microsoft Internet Explorer. Начиная с версии 2.0, этот вариант также поддерживает браузер Netscape. Данный вариант имеет свои неоспоримые преимущества и становится все более популярным /5/.

Чтобы включить поддержку карты для изображения, необходимо ввести дополнительный параметр в тэг IMG:

```
<IMG SRC="url" USEMAP="url#map_name">
```

Параметр USEMAP указывает, в каком месте находится карта описываемого изображения. Карта изображения определяет, какому участку изображения какой URL соответствует. Карта изображения может находиться в том же документе, что и изображение, или в другом документе. Помещение карты в другой документ позволяет собрать все карты изображений в одном документе (если, их несколько в различных документах), но добавляет еще одну итерацию в сети, когда за первую итерацию выясняется местонахождение карты, а за вторую - выполнение действия, предписанного URL для данного участка изображения. Параметр map_name указывает имя карты для изображения, а предшествующий ему URL определяет местонахождение карты. Если данный URL отсутствует, то карта с указанным именем ищется в текущем документе.

Рассмотрим синтаксис определения карты изображения:

```
<MAP NAME="map_name">
```

```
<AREA [SHAPE=" shape "] COORDS="x,y,..." [HREF=" reference "] [NOHREF]>
```

```
</MAP>
```

где

<MAP NAME="map_name"> - данный тэг определяет начало описания карты с именем map_name.

<AREA...> - описывает участок изображения и ставит ему в соответствие URL. Параметры:

SHAPE - необязательный параметр, указывающий на форму определяемой области изображения. Может принимать значения:

default - по умолчанию (обычно прямоугольник);

rect - прямоугольник;

circle - круг;

poly - многоугольник произвольной формы.

COORDS - координаты в пикселях описываемой области. Для прямоугольника это четыре координаты левого верхнего и правого нижнего углов, для круга - три координаты (две - центр круга, третья - радиус). Для многоугольника это описание каждого угла в двух координатах - соответственно число координат равно удвоенному количеству углов.

Координаты считаются с нуля, поэтому для описания области 100 на 100 используется описание:

<AREA COORDS="0,0,99,99" ...>

HREF="url" - описание ссылки, действия по которой будут выполняться при щелчке мыши в заданной области.

NOHREF - параметр, указывающий, что ссылка отсутствует для данного участка. По умолчанию, если не указан параметр HREF, то считается, что действует параметр NOHREF. Также, для всех неописанных участков изображения считается, что используется параметр NOHREF.

Если две описанные области накладываются друг на друга, то используется ссылка, принадлежащая первой из описанных областей.

</MAP> - данный тэг завершает описание карты изображения.

5.3.2 Формы

Форма — это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана.

Некоторые WWW-browser-ы позволяют пользователю, заполнив специальную форму, возвращающую полученное значение, выполнять некоторые действия на вашем WWW-сервере. Когда форма интерпретируется Web-браузером, создаются специальные экранные элементы, такие, как поля ввода, checkboxes, radiobuttons, выпадающие меню, скроллируемые списки, кнопки. Когда пользователь заполняет форму и нажимает кнопку "Подтверждение" (SUBMIT - специальный тип кнопки, который задается при описании документа), информация, введенная пользователем в форму, посылается HTTP-серверу для обработки и передаче другим программам, работающим под сервером, в соответствии с CGI (Common Gateway Interface) интерфейсом.

Формы передают информацию программам-обработчикам в виде пар [имя переменной]=[значение переменной]. Имена переменных следует задавать латинскими буквами.

Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Форма открывается тэгом <FORM> и заканчивается меткой </FORM>. HTML-документ может содержать в себе несколько форм, однако формы не должны находиться одна внутри другой.

Тэг <FORM> может содержать три атрибута, один из которых является обязательным:

ACTION: Обязательный атрибут. Определяет, где находится обработчик формы.

METHOD: Определяет, каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику. Допустимые значения: METHOD=POST и METHOD=GET. Если значение атрибута не установлено, по умолчанию предполагается METHOD=GET.

GET: методом "get" HTTP браузер берёт значение action, добавляет "?" к нему, затем присоединяет набор данных формы, кодированный с использованием типа содержимого "application/x-www-form-urlencoded". Затем перенаправляет всё по гиперссылке на этот URL. В этом сценарии данные формы ограничены кодами ASCII (нельзя использовать спецсимволы) и имеют весьма жесткие ограничения на объем вводимой информации.

POST: методом "post" HTTP браузер проводит транзакцию HTTP "post" (в теле HTTP-запроса), используя значение атрибута action и сообщение, созданное в соответствии с типом содержимого, определённым атрибутом enctype.

ENCTYPE: Определяет, каким образом данные из формы будут закодированы для передачи обработчику. Если значение атрибута не установлено, по умолчанию предполагается ENCTYPE=application/x-www-form-urlencoded. "Кнопка", чтобы запустить процесс передачи данных из формы на сервер, создается с помощью тэга.

<INPUT TYPE=submit>

Встретив такую строчку внутри формы, браузер нарисует на экране кнопку с надписью Submit, при нажатии на которую все имеющиеся в форме данные будут переданы обработчику, определенному в метке.

Надпись на кнопке можно задать любую путем введения атрибута VALUE="[Надпись]".

Пример:

<INPUT TYPE=submit VALUE="Отправить!">

Надпись, нанесенную на кнопку, можно при необходимости передать обработчику путем введения в определение кнопки атрибута NAME=[имя].

Пример:

<INPUT TYPE=submit NAME=button VALUE="Отправить!">

При нажатии на такую кнопку обработчик вместе со всеми остальными данными получит и переменную button со значением Отправить! (т.е. button=Отправить!, это можно видеть в адресной строке).

В форме может быть несколько кнопок типа submit с различными именами и/или значениями. Обработчик, таким образом, может действовать

по-разному в зависимости от того, какую именно кнопку submit нажал пользователь.

Существуют и другие типы элементов <INPUT>. Каждый элемент <INPUT> должен включать атрибут NAME=[имя], определяющий имя переменной, которая будет передана обработчику. Имя должно задаваться только латинскими буквами. Большинство элементов <INPUT> должны включать атрибут VALUE="[значение]", определяющий значение, которое будет передано обработчику под этим именем.

Основные типы элементов <INPUT>:

TYPE=text

Определяет окно для ввода строки текста. Может содержать дополнительные атрибуты SIZE=[число] (ширина поля для ввода, в символах) и MAXLENGTH=[число] (максимально допустимая длина вводимой строки в символах).

Пример:

```
<INPUT TYPE=text SIZE=30 NAME=student VALUE="Вася">
```

Определяет ширину поля в 30 символов, для ввода текста. По умолчанию в окне находится текст, который пользователь может редактировать. Отредактированный (или неотредактированный) текст передается обработчику в переменной student (student=содержимое_поля). Необходимо попробовать отредактировать поле.

TYPE=password. Определяет окно для ввода пароля. Абсолютно аналогичен типу text, только вместо символов вводимого текста показывает на экране звездочки (*), чтобы посторонний не мог прочесть.

Пример:

```
<INPUT TYPE=password NAME=pswd SIZE=20 MAXLENGTH=10>
```

Определяет окно шириной 20 символов для ввода пароля. Максимально допустимая длина пароля — 10 символов. Введенный пароль передается обработчику в переменной pswd (pswd=содержимое_поля). Необходимо попытаться ввести информацию в поле.

TYPE=radio. Определяет радиокнопку. Может содержать дополнительный атрибут checked (показывает, что кнопка помечена). В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка.

Пример:

```
<INPUT TYPE=radio NAME=modem VALUE="9600" checked> 9600  
бит/с
```

```
<INPUT TYPE=radio NAME=modem VALUE="14400"> 14400 бит/с
```

```
<INPUT TYPE=radio NAME=modem VALUE="28800"> 28800 бит/с
```

Определяет группу из трех радиокнопок, подписанных 9600 бит/с, 14400 бит/с и 28800 бит/с. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку, обработчику будет передана переменная modem со значением 9600 (modem=9600). Если пользователь отметит вторую кнопку, обработчику будет передана переменная modem со значением 14400 (modem=14400).

TYPE=checkbox

Определяет квадрат, в котором можно сделать пометку. Может содержать дополнительный атрибут checked (показывает, что квадрат помечен). В отличие от радиокнопок, в группе квадратов с одинаковыми именами может быть несколько помеченных квадратов.

Пример:

```
<INPUT TYPE=checkbox NAME=comp VALUE="PC">
```

Персональные компьютеры

```
<INPUT TYPE=checkbox NAME=comp VALUE="WS" checked>
```

Рабочие станции

```
<INPUT TYPE=checkbox NAME=comp VALUE="LAN"> Серверы  
локальных сетей
```

```
<INPUT TYPE=checkbox NAME=comp VALUE="IS" checked>
```

Серверы Интернет

Определяет группу из четырех квадратов. Первоначально помечены второй и четвертый квадраты. Если пользователь не произведет изменений, обработчику будут передана одна переменная comp с двумя значениями (comp=WS и comp=IS).

TYPE=hidden - определяет скрытый элемент данных, который не виден пользователю при заполнении формы и передается обработчику без изменений. Такой элемент иногда полезно иметь в форме, в него можно спрятать от пользователя служебные данные.

Пример:

```
<INPUT TYPE=hidden NAME=id VALUE="1">
```

Определяет скрытую переменную индексную id, которая передается обработчику со значением 1.

TYPE=reset - определяет кнопку, при нажатии на которую форма возвращается в исходное состояние (обнуляется). Поскольку при использовании этой кнопки данные обработчику не передаются, кнопка типа reset может и не иметь атрибута name.

Пример:

```
<INPUT TYPE=reset VALUE="Очистить поля формы">
```

Определяет кнопку "Очистить поля" формы, при нажатии на которую форма возвращается в исходное состояние.

Элемент <SELECT>:

Меню <SELECT> из n элементов выглядит примерно так:

```
<SELECT NAME="[имя]">
```

```
<OPTION VALUE="[значение 1]">[текст 1]
```

```
<OPTION VALUE="[значение 2]">[текст 2]
```

...

```
<OPTION VALUE="[значение n]">[текст n]
```

```
</SELECT>
```

Метка <SELECT> содержит обязательный атрибут NAME, определяющий имя переменной.

Метка `<SELECT>` может также содержать атрибут `MULTIPLE`, присутствие которого показывает, что из меню можно выбрать несколько элементов. Большинство браузеров показывают меню `<SELECT MULTIPLE>` в виде окна, в котором находятся элементы меню (высоту окна в строках можно задать атрибутом `SIZE=[число]`). Для выбора нескольких значений одновременно удерживают кнопку "SHIFT" и выбирают значения мышкой.

Пример:

```
<SELECT MULTIPLE SIZE=3 NAME="[имя]">
<OPTION VALUE="[значение 1]">[текст 1]
<OPTION VALUE="[значение 2]">[текст 2]
<OPTION VALUE="...">[...]
<OPTION VALUE="[значение n]">[текст n]
</SELECT>
```

Метка `<OPTION>` определяет элемент меню. Обязательный атрибут `VALUE` устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Метка `<OPTION>` может включать атрибут `selected`, показывающий, что данный элемент отмечен по умолчанию.

Пример:

```
<SELECT NAME="selection">
<OPTION VALUE="option1">Вариант 1
<OPTION VALUE="option2" selected>Вариант 2
<OPTION VALUE="option3">Вариант 3
</SELECT>
```

Такой фрагмент определяет меню из трех элементов: Вариант 1, Вариант 2 и Вариант 3. По умолчанию выбран элемент Вариант 2. Обработчику будет передана переменная `selection` (`selection=...`) значение которой может быть `option1` (по умолчанию), `option2` или `option3`.

Элемент `<TEXTAREA>`:

Пример:

```
<TEXTAREA NAME=address ROWS=5 COLS=50>
```

Поле для ввода большого текста, разбитого на абзацы.
`</TEXTAREA>`

Все атрибуты обязательны. Атрибут `NAME` определяет имя, под которым содержимое окна будет передано обработчику (в примере — `address`). Атрибут `ROWS` устанавливает высоту окна в строках (в примере — 5). Атрибут `COLS` устанавливает ширину окна в символах (в примере — 50). Текст, размещенный между метками `<TEXTAREA>` и `</TEXTAREA>`, представляет собой содержимое окна по умолчанию. Пользователь может его отредактировать или просто стереть.

5.3.3 CGI

CGI - Common Gateway Interface является стандартом интерфейса (связи) внешней прикладной программы с информационным сервером типа HTTP, Web сервер/5/.

Обычно гипертекстовые документы, извлекаемые из WWW серверов, содержат статические данные. С помощью CGI можно создавать CGI-программы, называемые шлюзами, которые во взаимодействии с такими прикладными системами, как система управления базой данных, электронная таблица, деловая графика и др., смогут выдать на экран пользователя динамическую информацию.

Программа-шлюз запускается WWW сервером в реальном масштабе времени. WWW сервер обеспечивает передачу запроса пользователя шлюзу, а она, в свою очередь, используя средства прикладной системы, возвращает результат обработки запроса на экран пользователя. Программа-шлюз может быть закодирована на языках C/C++, Fortran, Perl, TCL, Unix Shell, Visual Basic, Apple Script. Как выполнимый модуль, она записывается в поддиректорий с именем cgi-bin WWW сервера. Работа:

1) Передача данных шлюзам. Для передачи данных об информационном запросе от сервера к шлюзу, сервер использует командную строку и переменные окружения. Эти переменные окружения устанавливаются в тот момент, когда сервер выполняет программу шлюза.

2) Запросы для различных методов. Информация шлюзам передается в следующей форме:

имя=значение&имя1=значение1&...

где имя- имя переменной (из оператора FORM, например), и значение - ее реальное значение. В зависимости от метода, который используется для запроса, эта строка появляется или как часть URL (в случае метода GET), или как содержимое HTTP запроса (метод POST). В последнем случае, эта информация будет послана шлюзу в стандартный поток ввода. На файловый дескриптор стандартного потока ввода посылается CONTENT_LENGTH байт. Так же сервер передает шлюзу CONTENT_TYPE (тип передаваемых данных). Сервер не обязан посылать символ конца файла после отсылки CONTENT_LENGTH байт данных и после того, как шлюз их прочитает.

Пример: Возьмем результат работы формы с методом POST (METHOD="POST") в качестве примера. Пусть получено 7 байт, закодированных примерно так: a=b&b=c.

В этом случае, сервер установит значение CONTENT_LENGTH равным 7 и CONTENT_TYPE в application/x-www-form-urlencoded. Первым символом в стандартном потоке ввода для шлюза будет "a", за которым будет следовать остаток закодированной строки.

3) Аргументы командной строки. Шлюз в командной строке от сервера получает:

- остаток URL после имени шлюза в качестве первого параметра (первый параметр будет пуст, если присутствовало только имя шлюза), и
- список ключевых слов в качестве остатка командной строки для скрипта поиска, или

- чередующиеся имена полей формы с добавленным знаком равенства (на четных позициях) и соответствующих значений переменных (на нечетных позициях).

Ключевые слова, имена полей формы и значения передаются раскодированными (из HTTP URL формата кодирования) и перекодированными в соответствии с правилами кодирования Bourne shell, так что шлюз в командной строке получит информацию в том виде, как она есть, без необходимости осуществлять дополнительные преобразования.

Запросы оператора FORM. Запросы оператора FORM обрабатываются таким образом, что каждый параметр, отвечающий за имя поля, оканчивается знаком равенства, а остаток представляет собой значение этого параметра. Если присутствует что-либо после имени скрипта (шлюза), то эта информация передается в качестве первого параметра. Иначе первый параметр будет пуст.

Примеры:

`/htbin/foo/x/y/z?name1=value1&name2=value2`

вызывается как:

`./.../foo /x/y/z name1= value1 name2= value2`

а

`/htbin/foo?name1=value1&name2=value2`

вызывается как:

`./.../foo " name1= value1 name2= value2`

4) CGI переменные окружения. Следующие переменные окружения не являются специфичными по типу запросов и устанавливаются для всех запросов.

SERVER_SOFTWARE - Название и версия информационного сервера, который отвечает на запрос (и запускает шлюз). Формат: имя/версия .

SERVER_NAME - Имя хоста, на котором запущен сервер, DNS имя, или IP адрес в том виде, в котором он представлен в URL.

GATEWAY_INTERFACE - Версия CGI спецификации на тот момент, когда компилировался сервер. Формат: CGI/версия.

Следующие переменные окружения являются специфичными для разных запросов, и заполняются перед вызовом шлюза:

SERVER_PROTOCOL - Имя и версия информационного протокола, в котором пришел запрос. Формат: протокол/версия.

SERVER_PORT - Номер порта, на который был послан запрос.

REQUEST_METHOD - Метод, который был использован для запроса. Для HTTP, это "GET", "HEAD", "POST".

PATH_INFO - Дополнительная информация о пути, которую передал клиент. Другими словами, доступ к шлюзу может быть осуществлен по виртуальному пути, за которым следует некоторая дополнительная информация. Эта информация передается в **PATH_INFO**.

PATH_TRANSLATED - Сервер передает преобразованную версию **PATH_INFO**, которая включает в себя путь, преобразованный из виртуального в физический.

SCRIPT_NAME - Виртуальный путь к шлюзу, который должен выполняться, используемый для получения URL.

QUERY_STRING - Информация, следующая за ? в URL, к которому относится данный шлюз. Это информация представляет собой строку запроса. Она не должна быть декодирована никоим образом. Вне зависимости от командной строки эта переменная всегда должна быть установлена при наличии такой информации.

REMOTE_HOST - Имя хоста, производящего запрос. Если сервер не имеет такой информации, он должен установить REMOTE_ADDR, а это поле оставить не установленным.

REMOTE_ADDR - IP адрес хоста, производящего запрос.

AUTH_TYPE - Если сервер поддерживает идентификацию пользователя, и шлюз является защищенным от постороннего доступа, этот специфичный для протокола метод идентификации используется для проверки пользователя.

REMOTE_USER - Используется в ситуациях, аналогичных предыдущему случаю, для хранения имени пользователя.

REMOTE_IDENT - Если HTTP сервер поддерживает идентификацию пользователя согласно RFC 931, то эта переменная будет содержать имя пользователя, полученное от сервера.

CONTENT_TYPE - Для запросов, которые содержат дополнительную добавочную информацию, такие как HTTP POST и PUT, здесь содержится тип данных этой информации.

CONTENT_LENGTH - Длина данных, которую передает клиент.

В дополнение к этим, если запрос содержит дополнительные поля заголовка запроса, они помещаются в переменные окружения с префиксом HTTP_, за которым следует имя заголовка. Любые символы '-' в заголовке меняются на символы подчеркивания '_'. Сервер может исключить любые заголовки, которые он уже обработал, такие как Authorization, Content-type, и Content-length. Если необходимо, сервер может исключить любые (или вообще все) дополнительные поля заголовка в случае, когда их включение может привести к превышению предела размера переменных окружения. Примером такой переменной может служить переменная HTTP_ACCEPT, которая была определена в спецификации CGI/1.0. Другим примером может служить заголовок User-Agent.

HTTP_ACCEPT - Список MIME типов, которые клиент может обработать, как задано в HTTP заголовках. Другие протоколы должны получить эту информацию из других мест (если она им необходима). Каждый тип в этом списке должен быть отделен запятой согласно HTTP спецификации. Формат: тип/подтип, тип/подтип.

HTTP_USER_AGENT - Просмотрщик, который использует клиент для отправки запроса. Общий формат: программа/версия библиотека/версия.

5) Вывод информации шлюзом. Шлюз осуществляет свой вывод в стандартный поток вывода. Этот вывод может представлять собой или документ, сгенерированный шлюзом, или инструкции серверу, где получить необходимый документ.

Как правило, шлюз производит свой вывод, который интерпретируется и посылается обратно клиенту. Преимущество этого подхода состоит в том, что шлюз не должен посылать полный HTTP/1.0 заголовок на каждый запрос.

Заголовок выходного потока. Для некоторых шлюзов может быть, необходимо избегать обработки сервером их вывода, и общаться с клиентом непосредственно. Для того, чтобы отличить такие шлюзы от остальных, CGI требует, чтобы их имена начинались с префикса `prh-`. В этом случае, на шлюзе лежит ответственность за возвращение клиенту синтаксически правильного ответа.

Заголовки с синтаксическим разбором. Вывод шлюза начинается с маленького заголовка. Он содержит текстовые строки, в том же формате, как и в HTTP заголовке и завершается пустой строкой (содержащей только символ перевода строки или CR/LF). Любые строки заголовка, не являющиеся директивами сервера, посылаются непосредственно клиенту. В настоящий момент, CGI спецификация определяет три директивы сервера:

Content-type

MIME тип возвращаемого документа

Location

Это поле используется в случае, когда необходимо указать серверу, что возвращается не сам документ, а ссылка на него.

Если аргументом является URL, то сервер передаст клиенту указание на перенаправление запроса. Если аргумент представляет собой виртуальный путь, сервер вернет клиенту заданный этим путем документ, как если бы клиент запрашивал его непосредственно. Status - директива используется для задания серверу HTTP/1.0 строки-статус, которая будет послана клиенту. Формат: `nnn xxxxx`, где `nnn` - 3-х цифровой статус-код, и `xxxxx` строка причины, такая, как "Forbidden" (Запрещено).

Примеры: Предположим, имеется некоторый текстовый конвертер в HTML. Когда он оканчивает свою работу, он должен произвести следующий вывод в стандартный выходной поток:

--- начало вывода ---

Content-type: text/html

--- конец вывода ---

Теперь рассмотрим шлюз, который, в некоторых случаях, должен выдать документ `/path/doc.txt` с данного сервера, как если бы он был непосредственно востребован клиентом через `http://server:port/path/doc.txt`. В это случае вывод шлюза будет таков:

--- начало вывода ---

Location: /path/doc.txt

--- конец вывода ---

Наконец, предположим, что шлюз возвращает ссылки на gopher сервер, например на `gopher://gopher.ncsa.uiuc.edu/`. Вывод шлюза будет следующий:

--- начало вывода ---

Location: gopher://gopher.ncsa.uiuc.edu/

--- конец вывода ---

Non-parsed headers. Допустим теперь, что у нас имеется шлюз, который общается с клиентом непосредственно. Как уже отмечалось, его имя должно начинаться с префикса `ph-` и он должен возвращать допустимый HTTP заголовок. В этом случае, если доступ к шлюзу был осуществлен со значением `SERVER_PROTOCOL` равным `HTTP/1.0`, его вывод должен удовлетворять `HTTP/1.0`:

```
--- начало вывода ---  
HTTP/1.0 200 OK  
Server: NCSA/1.0a6  
Content-type: text/plain  
--- конец вывода ---
```

5.4 Контрольные вопросы

- 1) Какой тэг описывает участок изображения и ставит ему в соответствие URL?
- 2) Что определяет карта изображения?
- 3) Для чего нужны формы?
- 4) Какие значения может принимать атрибут "METHOD" в тэге `<FORM>`?
- 5) Какой тэг используется для ввода одной строки текста?
- 6) Какой тэг используется для ввода более одной строки информации?
- 7) Назначение атрибута "SELECT MULTIPLE"?
- 8) Для чего используется скрытый атрибут "HIDDEN"?

5.5 Тесты

- 1) Какой тэг позволяет вставить графическое изображение в HTML-документ?
 - а) `<INPUT TYPE=image>`;
 - б) ``;
 - в) ``;
 - г) `<BODY BACKGROUND="1.gif">`.
- 2) Возможно ли использовать изображение в качестве гипертекстовой ссылки?
 - а) да;
 - б) нет;
 - в) возможно только определённого формата.
- 3) Чтобы включить поддержку карты для изображения, необходимо ввести тэг:
 - а) ``;
 - б) `<BODY BACKGROUND="picture.gif">`;
 - в) ``.

4) Какой из атрибутов тэга AREA может указать форму определяемой области изображения?

- а) AREA;
- б) SHAPE;
- в) COORDS.

5) Какой из атрибутов тэга AREA может указать координаты в пикселях описываемой области?

- а) AREA;
- б) SHAPE;
- в) COORDS.

6) Что такое форма?

а) это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира;

- б) вид форматирования текста;
- в) способ отображения таблиц.

7) С помощью какого контейнера задается форма?

- а) FORM;
- б) INPUT;
- в) SELECT.

8) Какие контейнеры используются для создания элементов формы?

- а) INPUT;
- б) SELECT;
- в) TEXTAREA;
- г) SUBMIT;
- д) FORM.

Список использованных источников

- 1 Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации: Учебник/ Под редакцией А.П. Пятибратова. – М.: Финансы и статистика, 2001. – 512 с.
- 2 Майкл А. Ларсон Создание Web-страниц с помощью MS Office 97. — М.: БИНОМ, 1998. — 480 с.
- 3 Н. Рэндел, Д. Джонс Microsoft FrontPage в подлиннике. — СПб.: ВHV-Санкт-Петербург, 1997. — 432 с.
- 4 Хестер Н. Front Page 2002 для Windows: Пер. с англ. – М.: ДМК Пресс, 2002. – 448 с.
- 5 Пауэлл Т.А. Полное руководство по HTML / Пер. с англ. А.В. Качанов. – Мн.: ООО “Попурри”, 2001. – 912 с.
- 6 Основы Web-технологий / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин /Под. редакцией П.Б. Храмцова. – М.: ИНТУИТ.РУ ”Интернет-Университет Информационных Технологий”, 2003. – 512 с.

Приложение А
(справочное)

Ключ к тестам

- 1) 1(А,В); 2(Б); 3(А,Б); 4(Б); 5(А); 6(В);
- 2) 1(В); 2(Г); 3(В); 4(Г); 5(Г); 6(Б); 7(Б);
- 3) 1(А,Д); 2(Б); 3(А,В,Г); 4(А); 5(В);
- 4) 1(Г); 2(А); 3(А,Г); 4(Б); 5(Б); 6(А,Б,Г); 7(Г,Д);
- 5) 1(А,Г); 2(А); 3(А); 4(Б); 5(В); 6(А); 7(А); 8(А,Б,Г);
- 6) 1(В); 2(А); 3(А); 4(В); 5(А);

Приложение Б **(*справочное*)**

Структура отчета

- 1 Титульный лист.
- 2 Постановка задачи.
- 3 Содержание.
- 4 Теоретические предпосылки.
- 5 Краткие ответы на контрольные вопросы.
- 6 Список использованных источников.
- 7 Описание хода выполнения работы (последовательность выполняемых действий, команд) или программный ход.
- 8 Выводы.