**В.П. Поляков**

#### **Учебно-методический комплекс дисциплины «WEB – ПРОЕКТИРОВАНИЕ»**

(ДЛЯ СТУДЕНТОВ ЭКОНОМИЧЕСКИХ СПЕЦИАЛЬНОСТЕЙ)

Москва, 2006

Организационно-методический раздел

## Цель учебной дисциплины

Целью изучения дисциплины «WEB-проектирование» является формирование знаний, умений и навыков по созданию как в период обучения, так и в дальнейшей профессиональной деятельности сайтов различного назначения.

Дисциплина относится к области научных знаний, рассматривающей применение информационных систем и технологий в сфере экономики. Объектом изучения дисциплины является деятельность специалиста экономического профиля по использованию современных телекоммуникационных систем и сетевых технологий в профессиональной деятельности. Предмет изучения – компьютерные технологии проектирования сайтов.

## Задачи учебной дисциплины

Дать студентам общие сведения по технологиям проектирования сайтов, инструментальных средствах для создания и редактирования HTML-документов.

Студенты должны освоить применение редактора сценариев Microsoft Script Editor, дескрипторов языка HTML, элементов языка описания сценариев VB Script, а также создание форм для интерактивных HTML - страниц.

## Место учебной дисциплины в профессиональной подготовке дипломированного специалиста

Дисциплина «WEB-проектирование» относится к циклу специальных дисциплин, рекомендуемых для подготовки специалистов в области финансов, кредита, страхования, налогообложения, учета, анализа и аудита, международных экономических отношений, налоговых расследований, антикризисного управления, математических методов в экономике, государственного и муниципального управления. Освоение учебной дисциплины базируется на знаниях основ информатики, информационных систем и компьютерных технологий.

## Требования к уровню освоения содержания учебной дисциплины

В результате изучения дисциплины студенты должны

**ЗНАТЬ:**

1. Принципы использования глобальных вычислительных сетей в осуществлении профессиональной деятельности в сфере экономики и финансово–кредитных отношений.
2. Основные свойства и возможности языка гипертекстовой разметки текста HTML.
3. Способы применения инструментальных средств для создания и редактирования HTML – документов.

**УМЕТЬ:**

1. Уметь создавать сайты с использованием Microsoft Script Editor и VB Script.
2. Создавать формы для HTML – документов.

**ИМЕТЬ ПРЕДСТАВЛЕНИЕ** о тенденциях развития инструментальных средст создания сайтов.

# Объем дисциплины и виды учебной работы

Учебная дисциплина по выбору «WEB-проектирование» изучается в течение одного семестра. Итоговой формой контроля по изучаемой дисциплине является зачет без оценки.

Теоретическая часть курса дает представление об основных технологиях проектирования сайтов.

Практическая часть представляет собой курс обучения работе с использованием языков HTML, VB Script, а также редактора сценариев Microsoft Script Editor. Изучение этой части курса проводится в форме практических занятий в компьютерном классе. От студентов требуется наличие хороших навыков работы на компьютере (работа в среде Windows и с приложениями MS Office).

Трудоёмкость и виды учебной работы приведены в табл.1.

Таблица 1

|  |  |
| --- | --- |
| Виды занятий | Всего часов |
|
| 1. **Общая трудоемкость** | 28 |
| 1. **Аудиторные занятия** | 17 |
| 1. **Лекции** | 6 |
| 1. **Семинарские (практические) занятия** | 11 |
| 1. **Индивидуальная (самостоятельная) работа** | 11 |
| 1. **Формы итогового контроля** | Зачёт (без оценки) |

# СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

## ОСНОВНЫЕ СВЕДЕНИЯ О ТЕХНОЛОГИЯХ WEB – ПРОЕКТИРОВАНИЯ

Информационные технологии, использующие возможности Интернет, проникают в большинство сфер человеческой деятельности. Умение использовать эти возможности в экономической деятельности обеспечивают её успешность и способствует повышению эффективности. Электронная коммерция и банки, консалтинговые услуги, маркетинг с использованием корпоративных сайтов, предложение рекламной продукции посредством Интернет и другие услуги прочно входят в повседневную деловую жизнь. Создание персональных или корпоративных веб-страниц и размещение их в Интернет обеспечивает практически неограниченные возможности рекламы продукции и услуг, решение задач дистанционного образования. В настоящее время веб-пространство стало такой же платформой для создания приложений, как и персональный компьютер. Поэтому овладение технологиями создания веб-приложений является обязательным элементом информационной подготовки современного специалиста.

В ряду современных технологий веб-проектирования заметное место занимает CGI (Common Gateway Interface)-программирование, к достоинствам которого можно отнести:

* CGI-технология не зависит от платформы, CGI-программы применимы в операционных системах UNIX, Windows NT и почти для любого веб-сервера;
* создание CGI-программы не требует использования какого-то конкретного языка программирования.
* удобство интерфейса, обеспечивающее взаимодействие CGI-программы с веб-сервером на обычных UNIX-концепциях стандартного ввода и вывода и переменных среды *(*stdin*,* stdoutи environment variables).

Альтернативные технологии создания веб-приложений можно разделить на три категории:

* технологии, позволяющие вносить команды в текст HTML-страниц (например, Active Server Pages производства Microsoft, ColdFusion производства компании Allaire или PHP);
* технологии, вызывающие из страниц внешние программы (например, серверы приложений WebObjects);
* технологии, создающие программный интерфейс к веб-серверу (например, технология mod\_perl, NSAPI или ISAPI).

Технология Active Server Pages (ASP) обеспечивает создание веб-приложений с помощью HTML, программ и компонентов ActiveX. Страница ASP создается внедрением программы в текст HTML. При обращении пользователь к ASP-странице, веб-сервер выполняет программу, встроенную в эту страницу, и вставляет выходные данные этой программы в страницу как составную часть. ASP поддерживает по умолчанию языки VBScript и JavaScript, но возможна также поддержка других языков, в том числе Perl. В процессе первой загрузки страницы, программа компилируется в этой странице. Результат компиляции сохраняется на сервере до тех пор, пока исходный текст не изменится, затем программа перекомпилируется и записывается на сервере. Метод кэширования позволяет ASP избежать проблем связанной с низкой производительности (не нужно перекомпилировать программу всякий раз, когда пользователь обращается к ней).

Важнейшей частью технологии ASP являются компоненты ActiveX, или программные компоненты. Большинство пользователей компьютера знакомы с компонентами ActiveX по работе в Интернет, где их можно загрузить и использовать с браузером Internet Explorer. Компоненты ActiveX также могут использоваться для расширения свойств веб-сервера. Один из наиболее часто используемых компонентов ActiveX, известный как Active Data Objects (ADO) разработан компание Microsoft. Компонент ADO работает как посредник между страницей ASP и реляционной базой данных.

В разработке компонентов ASP можно использовать большинство современных языков программирования, включая C++, Java и Visual Basic.

ASP-программы вставляются в HTML-страницы и используются как интерфейс с ASP-компонентами. Программы компилируются Web-сервером и выполняются им же при обращении к Web-странице. Вывод программы вставляется непосредственно в страницу на место самой программы. В написании программы возможно использование различных языков, включая Visual Basic и JScript. Кроме того, Microsoft добавила поддержку дополнительных языков, таких как ActiveWare PerlScript, что позволяет вставлять в страницу программы на языке Perl.

Важным в технологии ASP является создание страниц HTML. Результаты работы компонентов и программ ASP, попадая на браузер, выглядят как обычные HTML-страницы. Преимуществом технологии ASP является то, что большая часть работы делаетсяе в виде HTML, дополняя ее некоторым количеством команд ASP.

Технология ColdFusion компании Allaire — это сервер приложений, поддерживающий множество программных интерфейсов веб-серверов (NSAPI производства Netscape, ISAPI производства Microsoft, CGI и др.). Технология ColdFusion по своему действию похожа на ASP. Для создания интерактивных веб-страниц в HTML-текст вставляются специальные знаки разметки, которые интерпретируются сервером, и браузер получает текст Web-страницы на стандартном языке HTML.

Главное отличие между ColdFusion и ASP состоит в том, что в ColdFusion каждая команда заключается в собственные знаки разметки, в то время как в ASP, создаются традиционные блоки команд. Это позволяет писать структурированные программы на ASP, но, с другой стороны, для тех, кто имеет опыт работы с HTML, но мало знаком с программированием, удобнее иметь дело с технологией ColdFusion.

Технология mod\_perl веб-сервера Apache выполнен в виде нескольких программ, называемых модулями. Получая запрос, веб-сервер передает его нескольким модулям, после чего каждый из них проверяет, должен ли он выполнять этот запрос. Технология mod\_perl объединяет интерпретатор языка Perl с веб-сервером Apache, что позволяет запускать CGI-программы, написанные на языке Perl, не требуя обработки этих программ интерпретатором всякий раз, когда на них поступает запрос. Запуск программ через встроенный интерпретатор вместо стандартного значительно улучшает производительность системы, и многие программы на Perl могут работать без дополнительных изменений.

Модуль mod\_perl создает для программ на языке Perl программный интерфейс с веб-сервером, позволяющий писать расширения для сервера Apache на языке Perl. Этот программный интерфейс позволяет дополнять или модифицировать любую функцию, встроенную в веб-сервер Apache, наиболее популярный в сети Интернет. Перенос CGI-программ в технологию mod\_perl, можно осуществить, настроив сервер так, чтобы программы с определенным расширением выполнялись модулем mod\_perl или указать определенный каталог, в которм все программы будут выполняться модулем mod\_perl. Но следует констатировать, что написание mod\_perl-модулей требует от специалиста очень высокой квалификации и прямого доступа к ресурсам сервер, что бывает невозможно в случае аренды ресурсов веб-сервера у хостинговых компаний.

Технология Java Servlet Development Kit компании Sun. Язык Java приобрел популярность как средство для создания аплетов, небольших программ, которые выполняются в браузере и дополняют веб-странице интерактивностью. Но по мере развития популярности этого языка оказалось, что он очень удобен для создания самых разных видов приложений, а не только аплетов. Одним из примеров применения языка Java является технология сервлетов, или программ, работающих на сервере. Главное достоинство среды Java — это переносимость. Компания Sun создала виртуальную машину Java, которая выполняет приложения Java для многих платформ. Если на компьютере установлена виртуальная машина Java, то на нем может выполняться любое приложения Java, независимо от того, на каком компьютере оно было скомпилировано. Пакет JSDK позволяет писать классы Java, которые работают как веб-приложения.

Для того чтобы создавать веб-страницы, необходимо знать язык гипертекстовой разметки и технологию их создания с использованием этого языка. Существуют три языка гипертекстовой разметки - SGML (Standard Generalized Markup Language - стандартный язык общей разметки), НТМL (Нурег Техt Markup Language - язык разметки гипертекста) и ХМL (Ехtensible Markup Language - расширяемый язык разметки). Язык SGML является прародителем НТМL. Он допускает создание дескрипторов с указанием, какие типы данных эти дескрипторы могут содержать, и определением состояний, которые могут поддерживаться их атрибутами. Язык НТМL является спецификацией, предусматривающей использование фиксированного набора дескрипторов. Эта спецификация определяет замкнутый класс документов - множество документов, отвечающих одному и тому же набору правил и имеющих схожие свойства. В отличие от НТМL язык ХМL может использовать возможности SGML, что позволяет определять собственные пользовательские классы документов. Язык НТМL является наиболее простым и понятным для изучения, поэтому освоение технологии создания веб-страниц целесообразно начинать с изучения НТМL.

Одним из базовых понятий веб–проектирования является понятие *веб-узла*, под которым понимается специальная папка, в которой размещены файлы, содержащие текстовую информацию по какой-либо теме, а также информацию в виде рисунков, графиков, фотографий, анимационных изображений и звуковых эффектов. В этих файлах содержатся описания веб-страниц на одном из языков разметки гипертекста – HTML или XML. Они имеют одно из следующих расширений: html, htm, xml. Все, что содержит веб-узел, далее будем называть веб-проектом. Существует три типа веб-узлов:

* создаваемые на веб-сервере поставщика услуг Internet;
* создаваемые в интрасети как веб-узлы группы;
* виртуальный веб-узел, который может быть создан на жестком диске автономного компьютера, не подключенного к какой-либо сети.

Информация на веб-узле размещается в виде отдельных страниц. Каждая веб-страница имеет вполне определенное содержательное и функционально законченное назначение. Поэтому такие страницы называют информационными статьями. Все страницы (статьи) определенным образом связаны друг с другом так, чтобы обеспечить пользователю удобный переход от страницы к странице и быстрый поиск интересующей его информации. Как правило, эта связь организуется по принципу родовых (иерархических) (рис.1) или сетевых отношений.

Родовые отношения обеспечивают удобство при просмотре содержимого веб-узла от общего к частному. Сетевые отношения создаются в тех случаях, когда целесообразно иметь возможность переходить с одних на другие страницы, для получения справочной либо уточняющей информации.

Каждая веб-страница хранится в отдельном файле. Связь между веб-страницами (файлами), обеспечивающая быстрый переход с одной страницы на другую и эффективный поиск нужной информации, устанавливается с помощью гиперссылок.

Одна из страниц выполняет роль главной страницы. В ней должна содержаться информация о тематической направленности проекта, а также элементы, обеспечивающие навигацию по страницам и поиск нужной информации. Именно эта страница будет отображаться первой на экране посетителя. Так, если в адресную строку браузера ввести, например DNS-адрес http://www.fa.ru, то на самом деле будет сформирован URL-адрес HTTP://www.fa.ru/index.htm и будет выполнена попытка найти и загрузить веб-страницу именно с таким URL-адресом.

Поэтому файл, в котором хранится первая веб-страница, и с которой посетитель начнет движение по страницам узла, используя гиперссылки, должен иметь имя "index.htm".

**Главная страница (общие сведения, навигатор)**

**(index.htm**)

**Сведения по первому разделу (теме)**

**Сведения по второму разделу (теме)**

**Уточняющая информация по разделу 1**

**Уточняющая информация по разделу 1**

**Уточняющая информация по разделу 2**

**Уточняющая информация по разделу 2**

Рисунок 1

Папка веб-узла внутри себя должна содержать еще одну папку. Эта папка служит для хранения файлов, содержащих какие-либо графические изображения, которые предполагается отображать на веб-страницах.

Веб-страница представляет собой документ, содержащий описание ее структуры и содержания, создаваемого посредством команд, сформированных на языке HTML. Эти команды выполняются программой - браузером, таким, например, как Microsoft Internet Explorer. Интерпретируя команды HTML, браузер создает визуальное изображение документа, собирая его из отдельных объектов. Таким образом, веб-документ - это изображение в окне браузера, которое он создает, выполняя команды языка HTML. Фактически веб-мастер создает не сам документ, а лишь описывает его структуру на языке гипертекстовой разметки. Сам документ создается браузером, интерпретирующим команды языка HTML. Таким образом, для каждой веб-страницы на узле должен быть помещен файл, содержащий документ HTML с ее описанием.

Возможности языка HTML таковы, что он обеспечивает только описание структуры HTML документа. Он, собственно, не является языком программирования. Для создания интерактивных web-страниц кроме языка HTML служат так называемые сценарии, представляющие собой программы, которые создаются на языках программирования, обеспечивающих их интерпретацию и выполнение браузером. Существует две разновидности таких языков – JavaScript и VBScript. Поэтому для того, чтобы создавать интерактивные web-страницы необходимо использовать язык HTML и один из указанных языков программирования.

## ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА И СТРУКТУРА HTML-ДОКУМЕНТА

Язык разметки гипертекста HTML служит для описания структуры веб-документа, в соответствии с которым веб-браузер формирует документ, и содержит набор команд, интерпретируя которые браузер создает документ.

Команды в HTML называются *дескрипторами* (tag), иногда их называют тегами. Дескриптор - это основной элемент кодирования, принятый в стандарте HTML. Дескрипторы помогают браузеру распознать, что нужно выделить, где отобразить графику, где находятся гиперссылки. Дескрипторы представляют собой ключевые слова, состоящие из последовательности букв, которые заключены в угловые скобки (< >). Так как веб-документ представляет собой упорядоченный набор объектов, то дескрипторы задают установки объектов, их свойств и методов. Дескрипторы имеют следующий синтаксис:

<ИМЯ\_HTML-ДЕСКРИПТОРА [атрибуты]>,

где ИМЯ\_ДЕСКРИПТОРА – определяет тип объекта, а необязательные атрибуты устанавливают его свойства и методы. Например:

<FONT size=12 color=blue>

Здесь FONT – имя дескриптора, а size и color- атрибуты, с помощью которых устанавливаются параметры шрифта: размер 12 пт., цвет шрифта голубой.

Дескриптор

<Input type=”Button” value=”OK” ID=”Button1” onclick=ИМЯ\_Сценария>

предписывает поместить в документ объект Кнопка с надписью ОК, которая имеет имя Button1, кнопка реагирует на событие click и выполняет сценарий с указанным именем.

Действие дескрипторов может распространяться на весь документ, на какую-то его часть документа или действовать только в месте его размещения (локально). В соответствии с этим дескрипторы бывают двух типов: контейнеры и одиночные.

*Контейнер* – это дескрипторная пара, состоящая из начального (или открывающего) и конечного (или закрывающего) дескриптора. *Начальный* *дескриптор* имеет вид

< ИМЯ\_HTML-ДЕСКРИПТОРА>

*Конечный* (*закрывающий*) *дескриптор* контейнера имеет вид

</ ИМЯ\_HTML-ДЕСКРИПТОРА >.

Закрывающий дескриптор аналогичен открывающему с той лишь разницей, что имени дескриптора предшествует символ слэш (/).

Контейнеры служат для хранения какой-либо информации, например текста или других дескрипторов HTML, в том числе и других контейнеров. Содержимое контейнера размещается между начальным и конечным дескрипторами. Конструкция, включающая начальный и конечный дескрипторы вместе с заключенным между ними содержимым, называется элементом HTML-документа. Например, отформатированный текст и дескрипторная пара <PRE> и </PRE>, в которую он заключен, представляет собой элемент:

<PRE> Это содержимое контейнера </PRE>

*Одиночный* *дескриптор* иногда называют пустым дескриптором, так как он не содержит никакой информации. Он имеет только начальный дескриптор и выполняет самостоятельную задачу, не связанную с конкретным текстом.

Примером может являться дескриптор < BR > - этот одиночный дескриптор предписывает переход на новую строку. Он действует локально именно в том месте, где поставлен и не требует закрытия.

Например:

<P> Особенности синтаксиса языка HTML <BR> и его применение </P>

Браузер после интерпретации этого HTML кода выведет на экран две строки:

**Особенности синтаксиса языка HTML**

**и его применение**

Документы HTML могут иметь различную структуру, включающую множество элементов. Но все они должны содержать два таких элемента, как:

* раздел заголовка страницы - HEAD;
* тело документа (страницы) – BODY.

*Раздел* *заголовка* служит для описания общих свойств страницы, таких как заголовок (имя) страницы, который будет отображаться в строке имени окна браузера, МЕТА указаний и описания таблиц стилей. МЕТА указания служат для задания параметров, которые необходимы для поисковых систем. Этот раздел формируется с помощью парного дескриптора <HEAD>. Внутри контейнера <HEAD> могут помещаться дескрипторная пара <TITLE>, содержащая внутри себя информацию, которая должна быть помещена в строку заголовка окна браузера и одиночный дескриптор <META>, который предназначен для записи информации, необходимой для поисковых систем.

*Тело* *документа* служит для описания содержимого HTML страницы - текстовой информации, таблиц, интерактивных элементов управления, графики и т.п. Тело документа заключается в дескрипторную пару <BODY> </ BODY>. Весь документ заключается в дескрипторную пару <HTML> </HTML>.

## ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ ВВОДА И РЕДАКТИРОВАНИЯ HTML-ДОКУМЕНТОВ

HTML- документ можно создать с помощью любого текстового редактора, например с помощью встроенного в операционную систему редактора БЛОКНОТ (NotePad). Так, например, для того, чтобы создать простой HTML-документ необходимо запустить встроенный редактор БЛОКНОТ, создать документ и поместить в него текст, приведенный в примере 1, сохранить документ, присвоив ему имя и указав расширение .htm

**Пример** 1:

<HTML>

<HEAD>

<TITLE> Exercise 1 </TITLE>

</HEAD>

<BODY bgcolor="#cccc99">

<FONT size=14 color=”blue”> Tag "Font" </FONT>

</BODY>

</HTML>

Далее следует открыть программу ПРОВОДНИК, найти в файловой системе сохраненный файл и открыть его. После выполнения этой операции инициализируется программа-браузер Internet Explorer, в окне которой отображается веб-документ (рис.2). В строке заголовка браузера помещен текст Exercise 1, то есть такой, который указан в дескрипторе <TITLE>, а текст Tag "Font" , помещенный внутри дескриптора <BODY>, отображается в окне браузера.

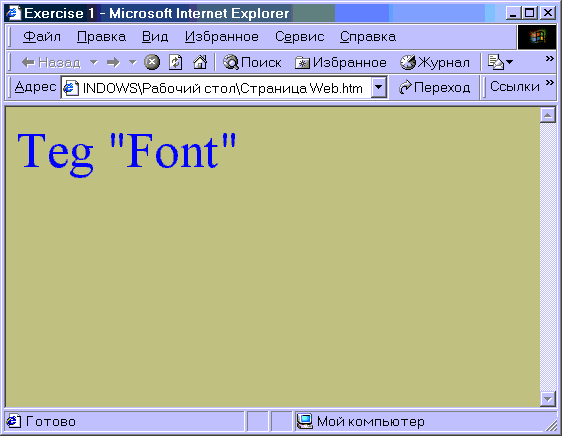


Рисунок 2

Код HTML в примере 1 содержит следующие дескрипторы:

* весь документ заключён в парный дескриптор < HTML > здесь содержание страницы < /HTML > , который указывает, что всё, что внутри этого дескриптора, браузеру необходимо интерпретировать. Внутри существуют ещё два больших элемента HEAD и BODY, тоже парные, которые требуют закрытия.
* HEAD - это раздел заголовка, в нём нужно описывать свойства документа в целом, такие как общий заголовок документа TITLE и некоторые другие.
* BODY - это так называемое тело документа, где описывается всё содержимое страницы.

У дескриптора BODY есть несколько атрибутов, с помощью которых можно определить какие-либо установки. Можно задать цвет страницы атрибутом bgcolor="X". Параметр X можно задать двумя способами: указанием названия цвета, например bgcolor="red" или шестнадцатеричной константой, например bgcolor="#017D75", (кавычки при указании значения атрибута обязательны). Можно установить цвет основного текста на страничке атрибутом TEXT="X". Параметр X - это значение цвета, устанавливаемое по любому из двух описанных выше способов. В качестве фона можно использовать рисунок, для этого в дескрипторе BODY следует задать следующий атрибут: background="Y". Здесь Y - это путь к файлу фона, например fon.gif, но для этого этот файл должен находиться в той же папке, что и файл странички, на которую он загружается. Можно зафиксировать фон на одном месте, добавив атрибут bgproperties="fixed". Так же можно указать цвета не просмотренных ссылок, атрибут LINK= , цвет активной ссылки (на которую наведён курсор) устанавливается атрибутом ALINK= , цвет просмотренных ссылок - атрибутом VLINK=.

Создание веб-страниц с помощью редактора БЛОКНОТ предполагает знание и умение использовать все возможные дескрипторы HTML. В этом редакторе нет средств, помогающих пользователю, поэтому использовать его для создания HTML документов достаточно сложно и неэффективно. Для этой цели разработан ряд специальных инструментальных средств, предназначенных для создания HTML документов. Среди них Front Page, HTML Editor и некоторые другие. В MS Office 2000 встроены специальные инструментальные средства для создания веб-страниц, такие как “Мастер Web-страниц” и MS VB Script Editor. С помощью Мастера Web-страниц можно создавать веб-страницы без интерактивных элементов, практически не зная и не используя язык HTML. Редактор MS VB Script Editor обеспечивает создание интерактивных веб-страниц, использующих элементы форм.

## СОЗДАНИЕ WEB-ПРОЕКТА С ПОМОЩЬЮ МАСТЕРА WEB СТРАНИЦ И MS WORD

Начиная с Word 2000 создание веб-страниц перестало существенно отличаться от создания стандартных документов. Команды Word и интерфейс программы практически не зависят от того, с каким документом ведётся работа: над обычным текстом или веб-страницей в формате HTML. Технологию создания Web-проекта с помощью инструмента ***Мастер Web-страниц*** рассмотрим на следующем примере. Пусть требуется разработать простой учебный веб-узел, содержащий Web-проект по теме:” Основы Web-проектирования” с помощью инструмента MS Office ***Мастер Web страниц***. С этой целью необходимо выполнить следующие действия:

1. Определите статьи проекта и их содержание. Пусть на страницах Web-проекта в соответствии с выбранной темой раскрываются следующие вопросы:

* Основные понятия Web.
* Технология создания узла с помощью Мастера Web-страниц.
* Технология применения активных элементов и применение редактора MS VB Script Editor.

Таким образом, веб-документ должен содержать три страницы (статьи): главную, содержащую информацию об основных понятиях Web, и две страницы, отражающие содержание второго и третьего вопроса.

2. Создайте виртуальный Web-узел, где будет храниться содержимое разрабатываемого Web-проекта. Это будет обычная папка на жестком диске с именем, например **Web\_design.** Для запуска программы ***“Мастер Web страниц”*** можно использовать два способа:

* выполните команду главного меню **Создать документ MS Office**. В раскрывшемся диалоговом окне **Создание** **документа** выберите вкладку **Web-страницы** и щелкните на **ОК** – раскроется диалоговое окно ***Мастер Web—страниц***.
* запустите приложение MS Word. В меню **Файл** выберите команду **Создать**. В области задач **Создание документа** в группе **Создать с помощью шаблона** выберите **Общие шаблоны**. Откройте вкладку **Web-страницы**. Дважды щелкните на значке ***Мастер Web-страниц***.

***Мастер Web-страниц*** предлагает пошаговую технологию создания Web-проекта. Таких шагов пять. Каждый шаг предполагает выполнение определенных операций. Переход к следующему шагу производится щелчком на кнопке **Далее** или на соответствующем квадратике в схеме последовательности ***Мастера***, возврат к предыдущим этапам – щелчком на кнопке **Назад**.

* На первом шаге нужно указать адрес веб-узла, в котором он будет храниться – URL адрес ( в нашем случае это путь к созданной ранее папке Web\_Design) и наименование веб-узла. Используя кнопку <Обзор>, установите адрес узла и в поле *Заголовок веб-узла* введите его название (рис. 3).

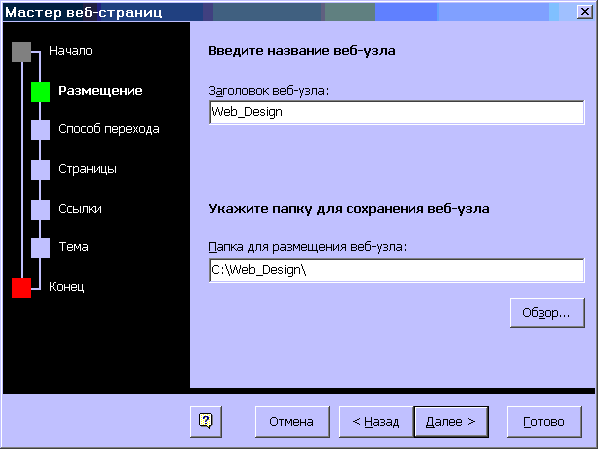


Рисунок 3.

* На втором шаге Мастер предлагает выбрать способ перехода от страницы к странице веб-документа, иными словами, предлагает выбрать тип навигатора. Если обозреватель не поддерживает такие объекты, как фреймы, то навигатор следует размещать на отдельной странице. Положим, в нашем проекте ссылки навигатора следует разместить вертикально в левой части окна, для этого установите соответствующий переключатель (рис. 4).

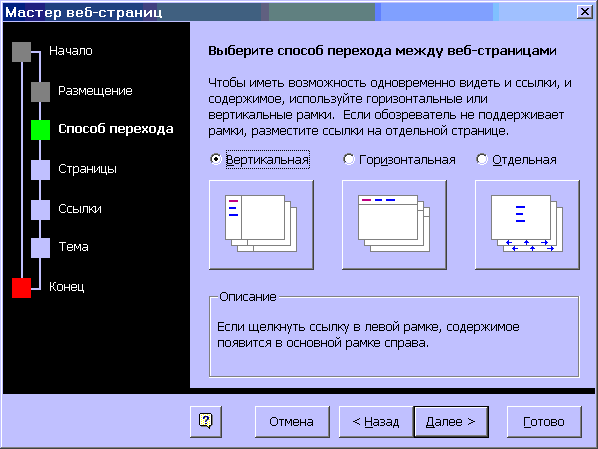


Рисунок 4.

* На третьем шаге можно определить схему и тип проекта – либо по шаблону, имеющемуся в библиотеке, из ранее созданного файла, либо создать новый проект (рис. 5). В нашем случае создается новый проект, состоящий из трех страниц. Используя кнопки Новая пустая страница и Удалить страницу, удалите ненужные страницы, например страницы, вставленные из шаблона, и добавьте новые.

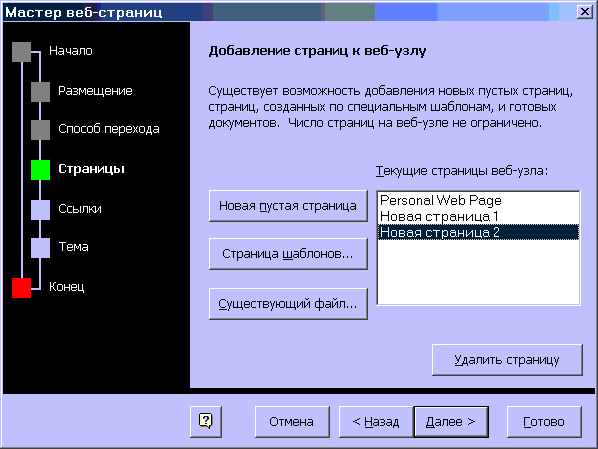


Рисунок 5.

* На четвертом шаге Мастер предлагает создать переходные ссылки между веб-страницами, а также указать названия этих страниц. Эти названия будут использованы в переходных ссылках, поэтому в них как можно точнее должна отражаться тематическая направленность каждой из страниц. Первой странице установите название “ Общие сведения о Web-проектировании”, второй – “Создание Web-проекта с помощью Мастера”, а третьей “Создание страниц с активными элементами” (рис. 6).

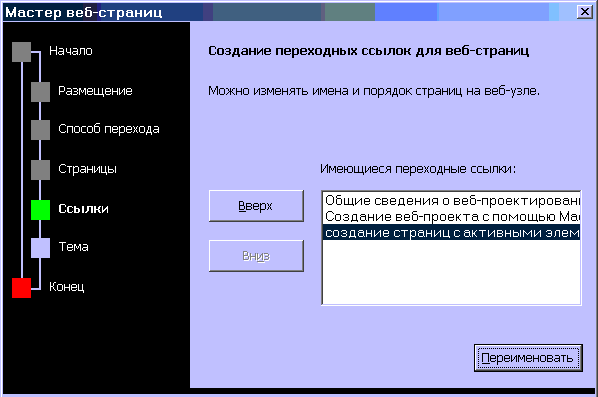


Рисунок 6.

* На пятом шаге Мастер предлагает установить Тему для оформления веб-узла – стилевое оформление. Если такое оформление будет применяться, то нужно установить соответствующий переключатель в окне Мастера и щелкнуть на кнопке Просмотр тем. В раскрывшемся окне Темы, выберите нужное стилевое оформление, например Перетекание

После выполнения операций пятого шага щелкните на кнопке **Готово** в результате на экране отобразится первая страница создаваемого веб-проекта (рис. 7) .

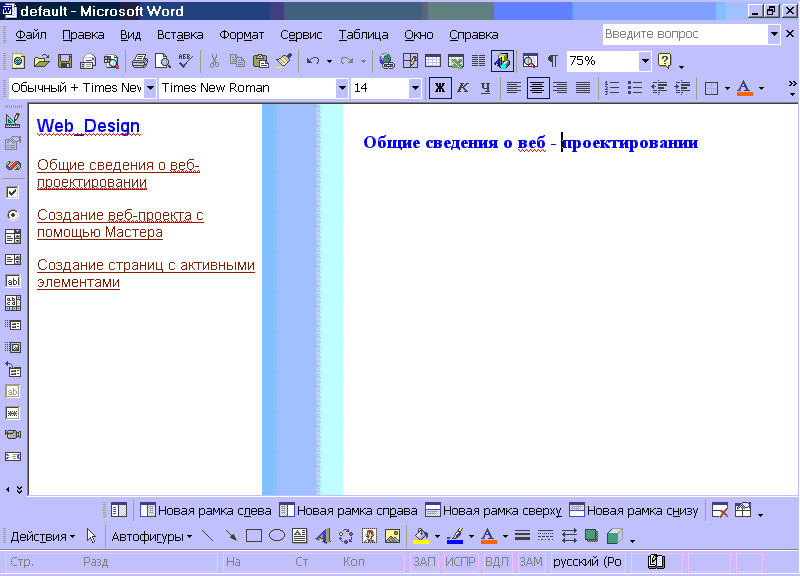


Рисунок 7.

В левой части экрана в виде столбца размещен навигатор, содержащий ссылки на все страницы проекта. В верхней его части помещено название веб-узла. В правой части – содержимое первой веб-страницы, которой было присвоено наименование “Общие сведения о Web-проектировании”. Используя навигатор, можно переключаться на каждую из страниц и редактировать (вводить и редактировать текст) их содержимое также, как в текстовом процессоре MS Word.

Для просмотра содержимого созданного веб-узла в Internet-обозревателе выполните следующие операции:

* закройте приложение MS Word;
* запустите программу Internet Explorer;
* в строке адреса обозревателя укажите путь к папке Web\_design и в ней выберите файл с именем default.htm.

**Редактирование веб-страниц в MS WORD**

Редактирование веб-страниц с использованием текстового процессора Word рассмотрим на следующем примере, поясняющем, как можно вводить текстовую информацию в веб-страницу, создавать внутренние ссылки в пределах одной страницы, помещать на страницу иллюстрации. Для этого продолжим работу со страницами созданного ранее веб-узла.

Пусть первая страница должна содержать информацию, включающую текст и подзаголовки, например такую:

**Пример 2:**

***Что такое WEB узел***

* + *Web- узел – это специальная папка, в которой размещены файлы, содержащие текстовую информацию по какой-либо теме в формате HTML, а также информацию в виде рисунков, графиков, фотографий, анимационных изображений и звуковых эффектов. Существует два вида Web- узлов:*
* *Создаваемые на Web- сервере поставщика услуг Internet*
* *Создаваемые в интрасети как Web- узлы группы*

***Последовательность создания проекта***

* + *Процесс создания проекта состоит из нескольких этапов.*
  + *На первом этапе, прежде чем приступать к созданию проекта, следует определиться с его функциональным назначением и тематической направленностью информационного материала.*
  + *На следующем этапе обычно проводится декомпозиция общей темы на тематические подразделы и отдельные статьи с определением связей между ними. Результатом такой декомпозиции является создание структуры проекта.*

***Структура HTML документа***

*Документы web могут иметь различную структуру, включающую различные элементы. Но все они должны содержать два таких элемента, как:*

* *Раздел заголовка страницы –HEAD;*
* *Тело документа (страницы) – BODY.*

*Раздел заголовка служит для описания общих свойств страницы, таких как заголовок (имя) страницы, который будет отображаться в строке имени окна браузера, МЕТА указаний и описания таблиц стилей. МЕТА указания служат для задания параметров, которые необходимы для поисковых систем.*

Для примера 2 предложены лишь фрагменты текста. Для удобства его просмотра на странице желательно поместить ее содержание и предусмотреть возможность перехода на нужный заголовок с помощью внутренних ссылок. Далее необходимо выполнить следующие действия:

* запустить приложение MS Word и открыть в нем документ, содержащий первую страницу Web- проекта с именем “Общие сведения о Web-проектировании” – отобразится первая страница;
* ввести на страницу приведенный выше текст примера 2;
* для всех заголовков установить стили заголовков, например “Заголовок 1”;
* после названия страницы вставить оглавление без использования номеров страниц и заполнителя. Оно будет выполнять роль навигатора по разделам этой страницы;
* сохранить документ с тем же именем “Общие сведения о Web-проектировании”.

**Добавление рисунков и графических изображений на страницу**

Веб-страница может содержать растровые рисунки и другие графические изображения, текст в виде бегущей строки, а также фильмы.

Поскольку рисунки должны будут передаваться по каналам Internet к компьютеру пользователя сети, и желательно, чтобы их передача осуществлялась как можно быстрее, рекомендуется их записывать в сжатые файлы в формате GIF или JPEG. Для подготовки новых рисунков или преобразования существующих к таким форматам можно использовать редакторы Corel Draw, Adobe Fotoshop или встроенный в операционную систему редактор Paint. Анимированные изображения и фильмы могут создаваться в форматах GIF и MPEG.

Технологию добавления рисунков и графических изображений на веб-страницу рассмотрим на примере 3, для чего создадим файл, содержащий рисунок в формате GIF, а затем поместим его на Web-страницу, созданную выше по примеру 2. Следуя теме нашего Web-проекта, положим, что рисунок должен изображать окно **Мастера Web-страниц**.

**Пример 3:**

* запустить описанным выше способом **Мастер** **Web-страниц** - появится диалоговое окно ***Мастер Web-страниц***;
* записать в буфер обмена изображение экрана монитора, используя клавишу **Print Screen**;
* запустить редактор Paint;
* в окно документа Paint поместить содержимое буфера обмена;
* используя соответствующие инструменты редактора, вырезать нужный фрагмент рисунка;
* создать новый документ и поместить в него содержащийся в буфере обмена фрагмент рисунка;
* сохранить рисунок в папке ранее созданного по примеру 2 веб-узла в формате .gif или .jpeg;
* запустить приложение MS Word и открыть файл, содержащий веб-страницу с заголовком “Создание Web-проекта с помощью Мастера”;
* используя команду меню **Вставка ⇒ Рисунок ⇒ Из файла** вставить созданный рисунок;
* сохранить документ и закрыть его;
* открыть измененный документ из программы Проводник или из программы Internet Explorer, при этом в окне браузера Internet Explorer отобразится рисунок.

**Создание полей форм**

Размещение на странице элементов формы, таких как “Бегущая строка”, “Фильм”, “Переключатель” и других производится с помощью панели инструментов “***Web***-***компоненты***”. Она включается командой меню **Вид ⇒ Панели инструментов**.

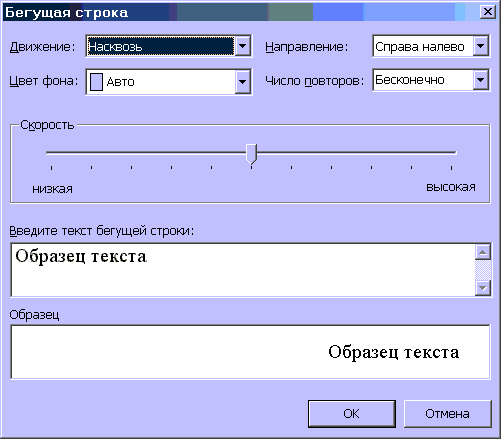
Каждое из полей форм после его размещения в документе Word становится объектом, который имеет имя (Name), какие-либо присущие ему свойства (Properties) и способен выполнять определенные действия при появлении событий.

Для размещения поля формы в поле документа нужно установить курсор в строку, где должен быть помещен элемент, и щелкнуть на выбранном элементе панели ***Web-компоненты***. Чтобы просмотреть или изменить свойства помещенного в документ объекта, следует перейти в режим **Конструктора**, щелкнув для этого на соответствующем значке панели ***Web-компоненты***, маркировать нужный объект, щелкнуть на кнопке <**Свойства**> и в раскрывшемся окне свойств установить нужные значения.

Технологию размещения поля формы в поле документа рассмотрим на примере 4, в соответствии с которым на ранее созданную веб-страницу требуется поместить поле формы “***Бегущая* *строка*”** и установить ее свойства.

Пример 4:

* открыть в приложении Word веб-страницу создаваемого веб-проекта с именем “Общие сведения о Web-проектировании” ;
* установить курсор в левую рамку страницы над навигатором;
* поместить на экран панель ***Web-компоненты***;
* выбрать на панели ***Web-компоненты*** поле формы ***Бегущая*** ***строка*** и щелкнуть на нем - в месте размещения курсора появится объект ***Бегущая*** ***строка*** и раскроется диалоговое окно ***Бегущая*** ***строка*** (рис. 8).



**Рисунок** 8**.**

* в диалоговом окне установить нужные свойства объекта и ввести текст, который должен отображаться в строке. Щелкнуть на кнопке **ОК**;
* перейти в режим конструктора и маркировать внедренный объект;
* используя команду меню **Формат** **⇒ Шрифт**, установить параметры шрифта (тип, размер, начертание, цвет);
* для изменения высоты и ширины объекта щелкнуть на элементе **Свойства** - раскроется окно свойств маркированного объекта;
* установить нужные значения свойств Height (высота) и MarqueWidth (ширина). Выйти из режима конструктора;
* сохранить документ, а затем открыть его в Internet Explorer.

## РЕДАКТОР СЦЕНАРИЕВ

В предыдущем разделе была рассмотрена технология создания веб-проекта, содержащего несколько страниц с использованием инструмента Мастер Web-страниц и текстового процессора MS Word. Она достаточно проста и доступна. Следует сказать, что использование MS Word для создания страниц Web является не лучшим стилем хотя бы по той причине, что HTML код такой страницы имеет существенную избыточность, требует для своего размещения неоправданно много памяти и, следовательно, потребуется больше времени для передачи кода по каналу связи. Это не всегда совпадает с интересами пользователя. Поэтому наилучшим стилем является описание веб-страницы на языке HTML, используя в качестве инструментальной среды какой-либо предназначенный для этой цели редактор. Существует целое семейство HTML-редакторов, например MS FrontPage, HTML Editor и других. Самым доступным и достаточно простым редактором является Редактор сценариев (Microsoft Script Editor), встроенный в MS Office (2000 и ХР).

**Запуск редактора сценариев**

Для запуска редактора сценариев нужно выполнить следующие действия:

* запустить одно из приложений MS Office, например MS Word или MS Excel;
* выполнить команду меню **Сервис ⇒ Макрос ⇒ Редактор сценариев**, или нажать комбинацию клавиш **Alt + Shift + F11**.

**Элементы интерфейса редактора сценариев**

После активизации инструмента ***Редактор*** ***сценариев*** на экране отобразится его окно, которое содержит ряд объектов, образующих интерфейс. Ограничимся кратким рассмотрением основных элементов интерфейса.

Окно документа HTML предназначено для ввода и редактирования документа. При создании нового документа ***Редактор*** ***сценариев***, используя шаблон, формирует и помещает в создаваемый документ основные дескрипторы.

Панель элементов полей форм содержит инструменты, с помощью которых в документ HTML можно помещать поля формы.

Окно проекта предназначено для отображения структуры Web-проекта. Используя это окно, можно быстро перейти на нужную HTML страницу, выполнив двойной щелчок на ее пиктограмме.

В окне свойств отображаются допустимые атрибуты для дескрипторов HTML. Оно состоит из трех частей. В верхней его части расположен раскрывающийся список, в котором отображается название того дескриптора, внутри которого помещен мерцающий курсор. Ниже раскрывающегося списка расположено окно атрибутов. Оно поделено на две части. В левой части в текущий момент времени отображаются возможные атрибуты того дескриптора, внутри которого помещен мерцающий курсор. В правой части отображаются параметры атрибутов. Используя окно свойств, для каждого дескриптора можно указать нужные атрибуты и задать для них необходимые параметры. Это создает определенные удобства, так как пользователю нет необходимости задумываться о том, какие атрибуты может иметь тот или иной дескриптор и какие параметры необходимо устанавливать.

Установку атрибутов и их значений можно выполнить двумя различными способами.

Способ первый:

* установить курсор после ключевого слова дескриптора и сделать пробел - раскроется список атрибутов (рис. 9);

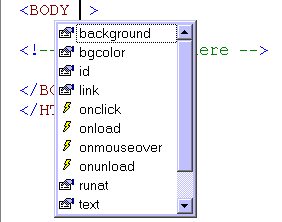


Рисунок 9

* в списке выбрать нужный атрибут и дважды щелкнуть на нем или нажать клавишу **TAB** - в тексте документа появится название атрибута;
* Ввести знак “=” – раскроется список возможных значений атрибута или кнопка указателя (рис. 10);
* щелкнуть на кнопке указателя – раскроется окно для установки значения (например, окно селектора цвета или окно установки URL);
* установить нужное значение параметра и нажать клавишу **Enter**.

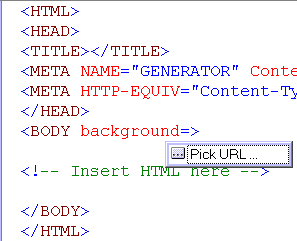


Рисунок 10

Способ второй:

* установить курсор после ключевого слова дескриптора;
* в левой части окна свойств (рис. 11) выбрать нужный атрибут и щелкнуть на нем. Используя кнопку или список в правой части окна свойств, установить нужное значение параметра.

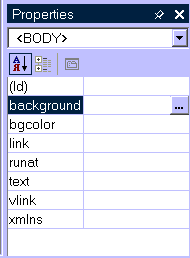


Рисунок 11

**Создание HTML документа в среде редактора сценариев**

Для создания веб-страницы одним из описанных выше способов активизируйте редактор сценариев и выполните команду меню File⇒NewFile – в результате откроется диалоговое окно *New File*.

Выберите шаблон *HTML Page*, раскройте список, расположенный правее кнопки Open*,* и выберите в списке элемент *Open With* - раскроется диалоговое окно *Open With* (рис. 12). Если создаваемая страница будет главной в создаваемом проекте, то в списке этого диалогового окна выберите *HTML/XML Editor (Defoult).* Если создаваемая страница будет одной из подчиненных, то выберите в списке *HTML/XML Editor with Encoding* или *Source Code (text) Editor*, затем щелкните на кнопке [Open].

После выполнения этой операции откроется окно редактирования HTML-кода для создаваемого документа, в котором по умолчанию будет создана заготовка HTML-документа в виде базового шаблона, представленного в примере 5.

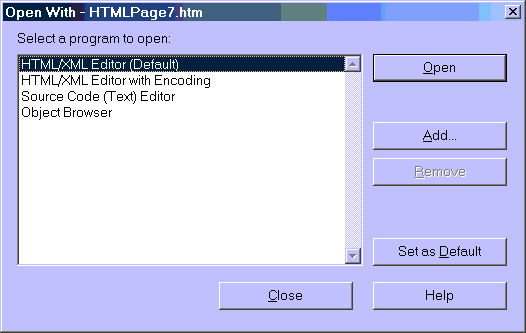


Рисунок 12

**Пример 5:**

<HTML>

<HEAD>

<TITLE></TITLE>

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<BODY>

<!-- Insert HTML here -->

</BODY>

</HTML>

Сохраните документ, присвоив файлу имя. Приведенный в примере 5 шаблон можно использовать как основу для создания веб-страницы. Рассмотрим элементы, которые он включает. Весь документ заключён в парный дескриптор< HTML >< /HTML > который указывает, что всё, что размещено внутри этого дескриптора, браузеру необходимо интерпретировать. Внутри помещены ещё два парных дескриптора HEAD и BODY.

Дескриптор HEAD - это раздел заголовка. Он устанавливает свойства документа в целом. В нем может находиться ряд других элементов, включая:

* TITLE - название документа, которое будет отображаться в строке имени браузера;
* META – сообщает браузеру дополнительные сведения о документе;
* BASE – задает базовый адрес документа;
* STYLE – определяет внешний вид документа;
* ISINDEX – указывает, что документ будет использоваться в поисковых системах;
* SCRIPT – определяет операторы интерпретируемых языков программирования таких, как JavaScript и VBScript.

Из перечисленных элементов дескриптор TITLE является обязательным. Он должен содержать простое, короткое и содержательное название, наиболее полно отражающее функциональное назначение веб-страницы. В названии следует избегать общих фраз. Например, из двух названий: *<TITLE> Проектирование веб-страниц </TITLE>* и *<TITLE> Проектирование Web-страниц, использование HTML </TITLE>* второе является более удачным, поскольку оно конкретизирует содержание Web-страницы.

Элемент <META> предназначен для предоставления поисковым веб-серверам информации для каталогизации и индексации страниц. Он включает атрибуты:

* NAME или HTTP-EQUIV - указывает, какая информация содержится в атрибуте CONTENT;
* CONTENT – служит для краткого описания странички с помощью ключевых слов, которые позволяют ориентироваться при поиске информации на заданную тему, также он может содержать имя автора и название приложения, в котором страничка создана.

Контейнер HEAD может содержать несколько элементов META, так как, например, в коде HTML-документа, представленного примером 6:

**Пример 6:**

<HEAD>

<TITLE> Проектирование Web-страниц-использование HTML </TITLE>

<META NAME="KeyWords" Content=" дескрипторы, контейнеры, элементы, разметка страницы,графика ">

<META NAME="Author" Content="Kafedra IT">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

## ПРИМЕНЕНИЕ HTML ДЛЯ ОПИСАНИЯ ВЕБ-ДОКУМЕНТА

**Ввод и форматирование текста.** **Создание структуры документа с помощью заголовков**

При разработке HTML документов целесообразно в первую очередь создать структуру, используя заголовки. В HTML существует шесть уровней заголовков, каждому из которых соответствует свой размер и стилевое оформление шрифта. Для создания заголовков служат контейнерные дескрипторы <H1>, <H2>, <H3>, <H4>, <H5>, <H6>. Синтаксис этих дескрипторов имеет вид:

<Hn> Содержимое дескриптора (текст) </Hn>

**Форматирование текстовых абзацев**

В документе HTML текстовая строка может быть любой длины. Internet-браузер самостоятельно выполняет перенос строки, если она не умещается по ширине экрана. Но в ряде случаев требуется специальное оформление текста с разбивкой на абзацы и на строки. Это можно сделать с помощью специальных одиночных дескрипторов <P> и <BR>. Дескриптор <P> служит для вставки строки текста, для разрыва абзаца и вставки пустой строки. Например:

<P> Выводимый текст

Дескриптор <BR> выполняет разрыв строки. Он может иметь атрибут CLEAR="X". Обычно этот атрибут применяют, при вставке изображений, если необходимо установить обтекание его текстом. Эти дескрипторы в документе HTML записываются так, чтобы они предшествовали абзацу или новой строке. Дескрипторы разрыва строк удобно использовать для создания списков.

**Выравнивание и форматирование текстовых фрагментов**

Стандартной установкой выравнивания текста является выравнивание по левому краю. Другие виды выравнивания заголовков и текста - по центру или по правому краю окна браузера - служат атрибуты, применяемые в дескрипторах в соответствии с приведенным ниже синтаксисом:

<Hn ALIGN= вид выравнивания>

<P ALIGN= вид выравнивания>

Ключевое слово ALIGN является наименованием атрибута, а “вид выравнивания” – устанавливаемое значение (RIGHT, LEFT или CENTER).

Если на Web-страницу нужно поместить уже отформатированный текстовый фрагмент из другого документа, например документа, подготовленного в редакторе БЛОКНОТ, и сохранить это форматирование, то для сохранения форматирования в HTML коде можно использовать контейнерный дескриптор PRE.

**Форматирование текста с помощью стилей**

HTML позволяет форматировать текст несколькими различными способами: с помощью физических стилей, с помощью логических стилей и устанавливать размер шрифта. Текст, отформатированный физическим стилем, во всех браузерах отображается одинаково, а отформатированный логическим стилем в различных браузерах может отображаться по-разному. Для форматирования текста с помощью стилей существуют специальные дескрипторы. В таблице 1 приведены дескрипторы для создания физических стилей.

**Таблица** 1**. Дескрипторы для создания физических стилей**

|  |  |
| --- | --- |
| Дескриптор | Назначение |
| < B> | Устанавливает полужирное начертание текста |
| < I > | Устанавливает курсив |
| < U > | Устанавливает подчеркивание текста |
| <TT> | Устанавливает моноширинный текст как у пишущей машинки |
| < SUP > | Устанавливает надстрочный текст |
| <SUB > | Устанавливает подстрочный текст |
| <STRIKE> или <S> | Устанавливает зачеркнутый текст |

Дескрипторы логических стилей указывают на характер форматирования, а не на точный способ отображения текста в окне браузера. Все логические стили устанавливаются с помощью специальных контейнеров (см. табл.2).

**Таблица** 2**. Дескрипторы для создания логических стилей**

| Дескриптор | Назначение |
| --- | --- |
| < EM > | Выделяет текст курсивом |
| <STRONG> | Сильно выделенный текст |
| <CITE> | Текст в виде цитаты |
| < CODE | Указывает, что введенный текст является кодом HTML |
| <DFN> | Текст, представляющий собой определение |
| <SAMP> | Указывает, что введенный текст является кодом HTML |
| <KBD> | Текст, представляющий название клавиши клавиатуры |
| <VAR> | Текст, определяющий переменную программы или значение |
| < BIG > | Увеличивает размер шрифта |
| < SMALL > | Уменьшает размер шрифта |
| < SAMP> | Описывает вывод компьютерной программы |

**Изменение размера и цвета шрифта**

Для изменения размера шрифта в HTML предусмотрен одиночный дескриптор <FONT>, который имеет атрибут SIZE. Используя этот атрибут, можно установить абсолютный размер шрифта (от 1 до 7) или относительный, по отношению к основному шрифту страницы (-4 до +4). Синтаксис определения размера шрифта следующий:

<FONT SIZE= значение>

Размер основного шрифта документа устанавливается одиночным дескриптором <BASEFONT>, который имеет следующий синтаксис:

<BASEFONT SIZE=значение>

Для изменения цвета шрифта служит атрибут COLOR:

<FONT COLOR= Х>

COLOR= Х - устанавливает цвет текста. Параметр Х задается шестнадцатиричным кодом или словом, указывающим цвет на английском языке. Для указания типа шрифта служит атрибут FACE.

**Оформление HTML-документа** **с помощью линий**

В веб-страницу можно помещать горизонтальные линии. Они могут быть использованы для разделения текста на части. Где и какую линию необходимо поместить указывается с помощью одиночного дескриптора <HR>. Этот дескриптор имеет параметры, с помощью которых можно изменять внешний вид линий или их положение в окне браузера. В таблице 3 приведены атрибуты дескриптора <HR>.

Таблица 3 Атрибуты дескриптора <HR>

|  |  |
| --- | --- |
| Атрибут | Назначение |
| SIZE | Определяет толщину линии |
| WIDTH | Определяет длину линии |
| ALIGN | Определяет вид выравнивания линии (LEFT, CENTER, RIGHT) |

Дескриптор <HR> c атрибутами имеет синтаксис:

<HR SIZE=значение> - значение указывается в количестве пикселей.

< HR WIDTH=”значение в процентах от ширины экрана”>

< HR WIDTH=”значение в пикселях”>

< HR ALIGN= вид выравнивания>

Пример кода, определяющего линию толщиной 4 пикселя, длиной 70% ширины экрана, выровненной по центру:

<HR SIZE=4 WIDTH=70% ALIGN=CENTER>

**Вставка комментариев**

Комментарии служат для того, чтобы можно было проще разобраться в HTML коде документа. Они служат пояснениями к отдельным фрагментам кода. Для вставки комментариев используется синтаксис:

<! --Текстовая часть комментария -->

Комментарии могут занимать несколько строк:

<! – Этот комментарий

занимает две строки

-->

**Вставка гиперссылок**

Гиперссылки служат для обеспечения перехода с одного раздела текста на другой или для переключения с одной Web-страницы на другую. В HTML есть три типа гиперссылок – внутренние, внешние и относительные.

*Внутренние ссылки* – это ссылки на объекты, которые находятся в том же документе. С их помощью можно перемещаться по одной Web-странице от раздела к разделу.

*Внешние ссылки* – это ссылки на другие Web-серверы.

*Относительные* или *локальные ссылки* – это ссылки на другие Web-страницы, размещенные на том же сервере, что и страница, содержащая ссылки.

В основе ссылок лежит URL (Uniform Resource Locator) - унифицированный локатор ресурсов. URL – это адрес, который отображается в строке адреса браузера и включает две части – тип протокола и собственно адрес. Наиболее распространенным протоколом является http:// (HyperText Transfer Protocol) – протокол передачи гипертекста, предназначенный для передачи HTML-документов.

Собственно адрес – это список элементов (папок, каталогов, файлов), указывающих путь к файлу, иными словами – это путь к файлу, например: http://www.huracan-oreiro.narod.ru/read.htm. В этом примере префикс http:// указывает тип используемой службы Internet, www.huracan-oreiro – представляет имя сервера. Компонент narod.ru определяет домен, в котором находится сервер.

**Создание внутренних ссылок**

Если Web-страница по содержанию объемная и содержит несколько разделов, то при создании ее HTML кода следует предусмотреть возможность переключения с одного раздела текста на другой. Эту возможность можно обеспечить с помощью внутренних ссылок. Внутренние ссылки создаются посредством объектных привязок. Объектом в данном случае является часть текста, которой присваивается имя. Определение раздела текста именованным объектом выполняется с помощью дескриптора <A> и имеет следующий синтаксис:

<A NAME=”ИМЯ\_РАЗДЕЛА”> Раздел текста </A>

На именованный объект можно ссылаться с помощью конструкций вида:

<A HREF=”ИМЯ\_РАЗДЕЛА” > Текст ссылки </A>

Для иллюстрации использования внутренних ссылок рассмотрим пример 7. Положим, что веб-страница содержит несколько больших разделов, которые включают общие сведения об услугах банка, а также сведения об условиях предоставления каждой из услуг. Нужно обеспечить пользователю удобство работы, создав возможность переключения с одного раздела на другой и возврат в начало документа. Код HTML документа, который реализует такую возможность, приведен в листинге, а сама веб-страница изображена на рис. 13.

**Пример 7:**

<HTML>

<HEAD>

<TITLE>Услуги сбербанка </TITLE>

<META NAME="Author" Content="Kafedra IT">

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<BODY>

<!—Этот контейнер содержит первый раздел и контейнеры ссылок -->

<a name="#начало"><h4 align=center>Услуги сбербанка по вкладам граждан</h4> В настоящее время сбербанк России предоставляет следующие услуги по вкладам:<BR>

<!—Эти контейнеры содержат внутренние ссылки -->

<a href="#ДВ "> Депозитные вклады</a><BR>

<a href="#ПВ "> Пенсионные вклады</a><BR>

<a href="#СВ "> Срочные вклады</a><BR>

<a href="#ВВ"> Валютные вклады</a><BR>

</a>

<!— присвоение имен разделам текста с условиями для различных вкладов. Обратите внимание на то, что каждый именованный фрагмент заключается в контейнер дескриптора А -->

<a name="ДВ "><h4 align=center>Депозитные вклады </h3>( Текст с условиями депозитных вкладов)</A>

<a name="ПВ "><h4 align=center>"> Пенсионные вклады </h3>( Текст с условиями пенсионных вкладов)</A>

<a name="СВ "><h4 align=center>"> Срочные вклады</h3> (Текст с условиями срочных вкладов)</A>

<a name="ВВ "><h4 align=center>"> Валютные вклады </h3> (Текст с условиями валютных вкладов)</A><BR>

<!—Ссылка на начало документа -->

</a><a href="#начало" >В начало документа</a>

</BODY>

</HTML>

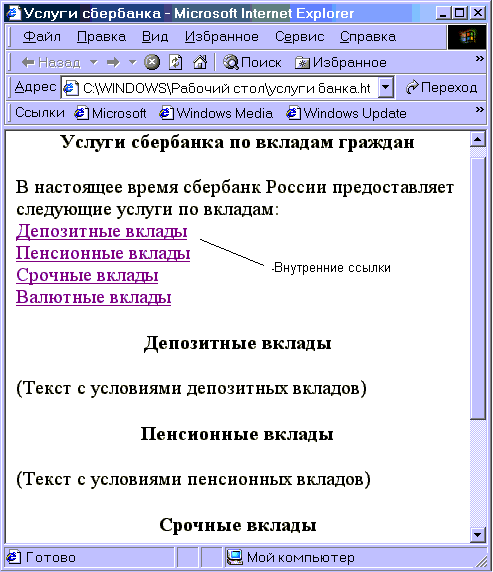


Рисунок 13

**Создание относительных гиперссылок**

Относительные гиперссылки – это ссылки на другие веб-страницы, расположенные на том же веб-сервере. Ссылки создаются, также как и в предыдущем случае, с помощью контейнерного дескриптора <A>. Этот дескриптор имеет атрибут HREF, значением которого является путь к файлу.

Для создания гиперссылки используется формат:

<A HREF=”URL”> ТЕКСТ ССЫЛКИ </A>

При создании относительной ссылки нет необходимости включать в URL доменное имя. Используя относительные ссылки, можно связывать веб-страницы не только с другими веб-страницами, но и с другими файлами (документами MS Word, текстовыми файлами, файлами мультимедиа и другими).

**Создание ссылок на адреса электронной почты**

Если тема веб-страницы такова, что желательно получить на нее отзыв или другую информацию по электронной почте, то следует предусмотреть в ней ссылки на адрес электронной почты. Используя эту ссылку, пользователь имеет возможность отправить сообщение непосредственно из браузера.

Синтаксис ссылки на адрес электронной почты имеет формат:

<A HREF=MAILTO:ЭЛЕКТРОННЫЙ\_АДРЕС > Текст адреса </A>

Для правильной ссылки на электронный адрес перед ним следует установить слово “MAILTO:”.

**Структурирование документа с помощью списков**

Списки являются хорошим средством структурирования документа. Это известно хотя бы из опыта использования текстовых процессоров. HTML имеет специальные дескрипторы, которые позволяют создавать пять типов списков. Наиболее часто используемыми типами списков являются списки, представленные в таблице 4.

Таблица 4. Дескрипторы списков

|  |  |
| --- | --- |
| Тип списка | Дескриптор |
| Нумерованный список | <OL> |
| Маркированный список | <UL> |
| Список определений | <DL> |

*Нумерованные* *списки* часто называют упорядоченными. Они создаются с помощью контейнера <OL> </OL>. Каждый элемент списка внутри этого контейнера определяется с помощью дескриптора <LI>. Синтаксис определения нумерованного списка имеет вид:

<OL TYPE=тип > <!—Определяется нумерованный список -->

<LI> 1 – ый элемент списка

<LI> 2 – й элемент списка

. . .

<LI> Последний элемент списка

</OL>

Атрибут TYPE устанавливает стиль форматирования списка - арабскими или римскими цифрами, а также буквами алфавита (как строчными, так и прописными). Параметр тип может принимать значения, приведенные в таблице 5 .

Для изменения вида нумерации элемента внутри списка следует установить значение параметра для атрибута TYPE в соответствующем этому элементу дескрипторе <LI>, используя конструкцию:

<LI TYPE=тип>

Последующие элементы списка будут иметь нумерацию, указанную в атрибуте TYPE дескриптора <OL>.

Таблица 5.

|  |  |
| --- | --- |
| **Значение параметра**  **«тип»** | **Стиль отображения** |
| 1 | Нумерация арабскими цифрами (применяется по умолчанию) |
| a | Строчные буквы латинского алфавита |
| A | Прописные буквы латинского алфавита |
| i | Нумерация строчными римскими цифрами |
| I | Нумерация прописными римскими цифрами |

В примере 8 приведен HTML – кода нумерованного списка, а на рисунке 14 – его отображение в окне браузера.

Пример 8:

<HTML>

<HEAD>

<TITLE></TITLE>

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<BODY background="file:///C:\Program Files\Microsoft Office\Office10\Bitmaps\Styles\ACSUMIPT.GIF">

<h4 align=center>Нумерованный список </h4>

<p align="left">Для запуска редактора сценариев выполните действия:</p>

<ol>

<li>Запустите приложение MS Word</li>

<li>Закройте открытый документ</li>

<li>Выполните команду Сервис/Макрос/Редактор сценариев </li>

</ol>

</BODY>

</HTML>

Весь список находится внутри контейнера <OL> </OL>. Каждый элемент списка представляет текстовую строку, но элементами списка могут быть гипертекстовые ссылки, изображения и любые другие элементы HTML.

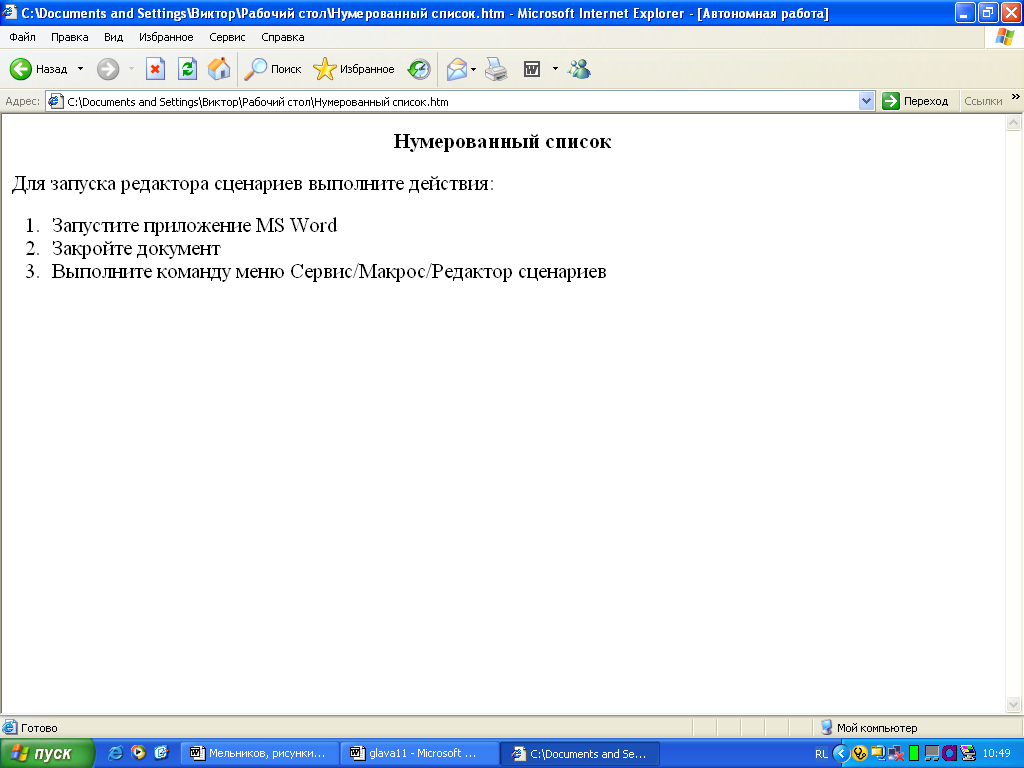


Рисунок 14

В отличие от нумерованных в *маркированных* (*неупорядоченных*) *списках* перед каждым элементом отображается маркер. Создание маркированных списков производится с помощью HTML -конструкции вида:

<UL TYPE=ТИП\_МАРКЕРА>

<LI TYPE=ТИП\_МАРКЕРА> элемент списка

<LI > элемент списка

…

<LI TYPE=ТИП\_МАРКЕРА> элемент списка

</UL>

Типы маркеров для маркированных списков приведены в таблице 6.

Таблица 6. Типы маркеров

|  |  |
| --- | --- |
| Тип маркера | Изображение в окне браузера |
| Disk | Заполненный кружок |
| Square | Заполненный квадратик |
| Circle | Незаполненный кружок |

Списки определений служат для создания абзацев, определяющих термин, указанный в элементе списка. Элементы списков определений состоят из названия элемента и его описания, поэтому такие списки в какой-то мере напоминают толковые словари. Название элемента устанавливается с помощью дескриптора <DT>, а его определение – с помощью дескриптора <DD>. Для создания списка определений применяется следующая синтаксическая конструкция:

<DL>

<DT> Элемент списка (термин)

<DD> Определение элемента списка

<DT> Следующий элемент списка (термин)

<DD> Определение элемента списка

…

<DT> Последний элемент списка (термин)

<DD> Определение элемента списка

</DL>

Используя приведенные выше дескрипторы, в документе можно создавать вложенные и комбинированные списки.

**Вставка таблиц и их редактирование**

Таблицы в HTML-документе могут создаваться в тех случаях, когда необходимо представить информацию в табличной форме, как это делается в обычных документах. Но кроме этого таблицы в HTML-документе могут выполнять и другие функции. В ячейках таблицы кроме тестовой или числовой информации могут размещаться графические изображения, аудио- и видеозаписи, а также гиперссылки. Используя возможности HTML-таблиц, можно создавать карту сайта.

Основным элементом таблицы является ячейка. Ячейки располагаются в строках. Строки размещаются горизонтально. Ячейки в HTML-документе определяются одна за другой построчно, начиная с левой позиции каждой строки.

Прежде чем создавать таблицу, нужно определить необходимое количество строк и столбцов.

**Создание таблицы**

Для размещения кода таблицы HTML служит контейнер TABLE. При описании строк и столбцов нужно указать количество строк и столбцов, а также их расположение. Строки определяются с помощью дескрипторной пары <TR> и </TR> (Table Rows).

Каждый контейнер TR содержит одну строку таблицы. Внутри этого контейнера помещаются описания ячеек. Для этой цели служит дескрипторная пара <TD> и </TD> (Table Data), а также дескрипторы <TH> и </TH> (Table Head) и атрибут COLSPAN. Внутри контейнера TD помещаются данные, записываемые в ячейки, а внутри контейнера TH – заголовки столбцов или названия строк таблицы. Контейнер TD размещается внутри контейнера TH. Если за дескрипторами <TD> и <TH> следуют другие дескрипторы <TD> , <TH> или <TR>, то они считаются закрытыми и закрывать их необязательно.

При создании таблицы, кроме определения количества строк и столбцов, следует также определить такие общие характеристики таблицы, как надпись, рамка, ширина таблицы, количество столбцов, интервалы между ячейками. Эти характеристики устанавливаются с помощью атрибутов дескриптора <TABLE>.

Для определения надписи (названия таблицы) служит контейнер <CAPTION>.

Для определения рамки предназначен атрибут BORDER=n, где n - толщина линии рамки, выраженная в количестве пикселей (по умолчанию принимает значение равное 1).

Для определения ширины таблицы служит атрибут WIDTH=ЗНАЧЕНИЕ, где ЗНАЧЕНИЕ указывает абсолютное значение ширины в пикселях или относительное в процентах от ширины экрана, например:

<TABLE BORDER=2 WIDTH= 600>

<TABLE BORDER=3 WIDTH= “60%”>

Определение количества столбцов производится посредством атрибута COLS= n, где n – целое число.

Для определения интервалов в таблице служат атрибуты CELLSPASING и CELLPANDING. Атрибут CELLSPASING определяет расстояние между ячейками, выраженное в пикселях, а атрибут CELLPANDING устанавливает расстояние между границей ячейки и ее содержимым, также в пикселях, например:

<TABLE BORDER=2 WIDTH= 600 CELLSPASING =4 CELLPANDING =5>

Последовательность создания HTML таблицы рассмотрим на следующем примере.

Пусть нужно создать таблицу, подобную рис. 15.

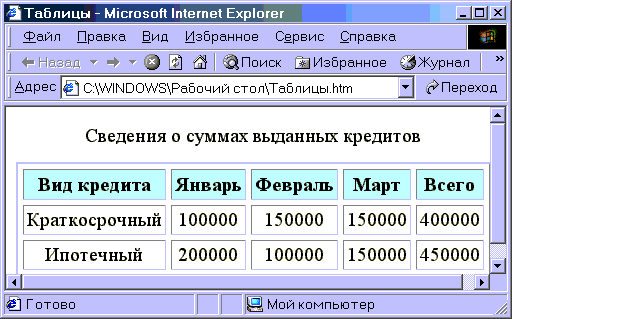


Рисунок 15

Таблица состоит из трех строк и пяти колонок. Первая строка таблицы является заголовками колонок. Содержимое ячеек выравнивается по центру. Между ячейками установлен интервал 5 пикселей, расстояние содержимого до границы ячейки составляет 3 пикселя. Таблица имеет границу шириной 2 пикселя. Цвет фона строки заголовка голубой, а цвет фона двух других строк – бледно-желтый.

Код HTML-документа, содержащий описание такой таблицы, приведен в примере 9.

Пример 9:

<HTML>

<HEAD>

<TITLE>Таблицы</TITLE>

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<BODY>

<!—Начало таблицы -->

<table width="80%" border=2 cols=5 cellspacing =5 cellpadding=3

align=center >

<!—Название таблицы -->

<caption align=top> Сведения о суммах выданных кредитов</caption>

<!—Строка заголовка -->

<tr align=center bgcolor="#ccffff"> <!—Начало строки-->

<th>Вид кредита</th> <!—первая ячейка-->

<th> Январь</th> <!—вторая ячейка-->

<th> Февраль</th> <!—третья ячейка-->

<th> Март</th> <!—четвертая ячейка-->

<th> Всего</th> <!—пятая ячейка-->

</tr> <!—окончание строки-->

<!—Вторая строка -->

<tr align=center bgcolor="#ffffcc"> <!—Начало строки-->

<td> Краткосрочный</td> <!—первая ячейка-->

<td>100000</td> <!—вторая ячейка-->

<td>150000</td> <!—третья ячейка-->

<td>150000</td> <!—четвертая ячейка-->

<td>400000</td> <!—пятая ячейка-->

</tr> <!—окончание строки-->

<!—Третья строка -->

<tr align=center bgcolor="#ffffcc"><!—Начало строки-->

<td> Ипотечный</td> <!—первая ячейка-->

<td>200000</td> <!—вторая ячейка-->

<td>100000</td> <!—третья ячейка-->

<td>150000</td> <!—четвертая ячейка-->

<td>450000</td> <!—четвертая ячейка-->

</tr> <!—окончание строки-->

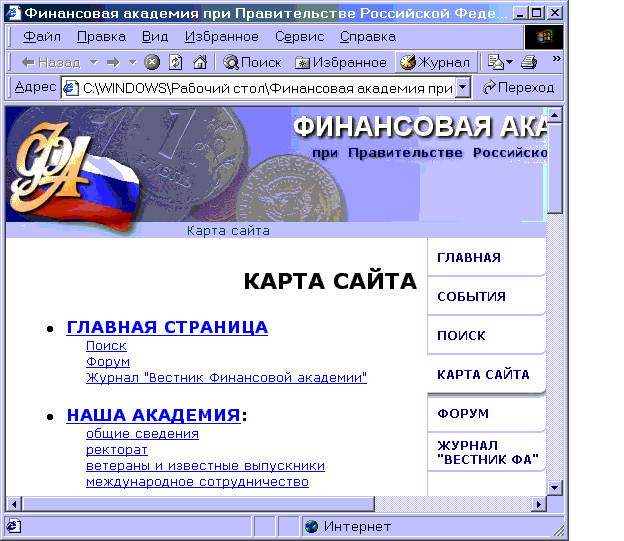
</table> <!—окончание таблицы -->

</BODY>

</HTML>

**Макетирование страниц с помощью таблиц**

Таблицы можно вкладывать одну в другую. Возможность создавать вложенные таблицы предоставляет хорошие средства для макетирования страниц. Это используется в тех случаях, когда на странице должно быть помещено много различных элементов, таких, например, как графические изображения, текст и др., расположенных в различных частях экрана. Эта возможность является удобной и при создании карты ссылок. На рис. 16 показана веб-страница Финансовой академии при Правительстве РФ. В ней использована табличная структура.



**Рисунок** **16**

Для создания табличной структуры страницы ее нужно определить, т.е. определить строки и ячейки, но без рамки и установить значение атрибута WIDTH=100%. Затем нужно определить вложенную таблицу с рамками. Вкладываемая таблица размещается в ячейке внешней таблицы, поэтому для того, чтобы вложить одну таблицу в другую, нужно поместить определение вкладываемой таблицы в контейнер TD. В примере 10 приведен листинг, демонстрирующий код, описывающий вложение таблиц, на рис. 17 показано изображение в окне веб-браузера.

**Пример 10:**

<BODY>

<TABLE border=1>

<TR>

<TD> Внешняя ячейка

<!—Вложение таблицы из двух ячеек -->

<TABLE border=2>

<TR> <TD> Вложенная ячейка 1

<TD> Вложенная ячейка 2

</TABLE>

</TABLE>

</BODY>

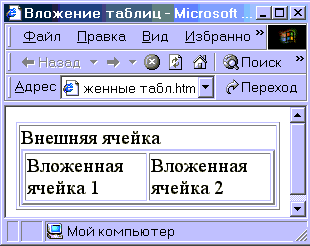


Рисунок 17

**Создание фреймов**

Последовательность создания веб-страницы с фреймовой структурой (т.е. состоящей из совокупности объектов) включает следующие этапы. На первом этапе следует создать документы, которые будут помещаться во фреймах – рисунки, формы, навигаторы, тексты, таблицы и др. Технология создания таких документов рассмотрена выше. На втором этапе создается документ, определяющий фреймовую структуру.

Для создания веб-страницы с фреймовой структурой служит контейнер FRAMESET. В HTML –документе дескрипторная пара <FRAMESET> </FRAMESET> помещается вместо дескрипторной пары <BODY> </BODY>.

Дескриптор <FRAMESET> имеет два атрибута - ROWS и COLS. С помощью атрибута ROWS указывается, что страница должна быть разделена на части, размещенные по вертикали, а атрибут COLS – по горизонтали. В каждом дескрипторе <FRAMESET> можно использовать только один из этих атрибутов. Устанавливаемое атрибутам значение определяет ширину или высоту фреймов и указывается в относительных величинах (процентах) или в абсолютных (пикселях). Синтаксис для этих атрибутов имеет следующий формат:

<FRAMESET ROWS=”высота\_окна1, высота\_окна2, …, \*”

<FRAMESET COLS=”ширина\_окна1, ширина\_окна2, …, \*”

Количество фреймов определяется количеством указанных значений в списке каждого атрибута. Значение, указанное символом “\*”, устанавливает, что последний фрейм занимает оставшуюся часть экрана.

Например, HTML-код:

<FRAMESET ROWS=70, \*>

указывает, что страница разбивается на два фрейма по вертикали. Верхний фрейм имеет высоту 70 пикселей, а нижний занимает оставшуюся часть экрана.

Для создания фреймовой структуры внутри фрейма нужно применять вложенные дескрипторы <FRAMESET> - так же как это делается в случае вложенных таблиц.

Контейнер FRAMESET выполняет примерно такое же назначение, как и контейнер TABLE при создании таблиц, т.е. первоначальную разметку. Для указания, что должно быть помещено внутрь фрейма, служит дескриптор <FRAME>, который выполняет роль подобную дескриптору <TD> в таблицах. Каждый дескриптор <FRAME> создает отдельное окно фрейма. С помощью этого дескриптора можно указать URL или имя, связанное с фреймом. Дескриптор <FRAME> имеет следующие атрибуты:

SRC – для указания URL;

NAME – устанавливает имя окна фрейма;

SCROLLING – устанавливает в окне фрейма линейку прокрутки;

NORESIZE – запрещает изменение размера окна фрейма;

ID – устанавливает имя фрейма;

TARGET – устанавливает имя фрейма, в котором нужно отобразить страницу, рисунок или содержимое файла.

Для установки значений атрибута TARGET существуют специально зарезервированные имена, при указании которых ссылка поведёт себя следующим образом:

\_blank -документ откроется в новом окне браузера

\_self -ссылка откроется в пределах текущей панели

\_parent -документ откроется в начальном наборе фреймов

\_top -документ откроется в обычном режиме окна, а фреймы исчезнут.

Создание веб-страницы с фреймовой структурой может быть проиллюстрировано примерами 11, 12. В верхнем фрейме размещается страница с текстом “Работа с фреймами”, в другом фрейме – навигатор, созданный в виде таблицы, в третьем фрейме помещаются веб-документы, вызываемые с помощью гиперссылок. Изображение в окне браузера приведено на рис. 18.

HTML – код первой страницы приведен в примере 11, а текст кода страницы-навигатора – в Примере 12.

**Пример 11**

<HTML>

<HEAD>

<TITLE>Фреймы</TITLE>

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<frameset rows=20%,\*>

<frame frameborder=1 name=first src="file:///C:\Страница1.htm" >

<frameset COLS=20%,\*>

<frame name=Two frameborder=1 src="file:///C:\Навигатор.htm">

<frame frameborder=1 name=three>

</frameset>

<frame frameborder=1 >

</frameset>

</HTML>

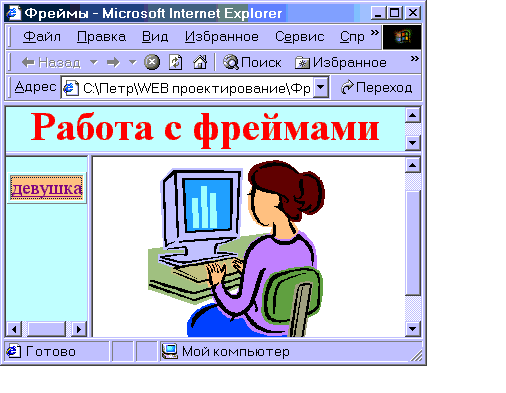


Рисунок 18

**Пример 12:**

<HTML>

<HEAD>

<TITLE>Навигатор</TITLE>

<META NAME="GENERATOR" Content="Microsoft Visual Studio">

<META HTTP-EQUIV="Content-Type" content="text/html; charset=UTF-8">

</HEAD>

<BODY bgcolor="#ccffff">

<table width=15% height=10% border=1>

<tr> <td bgcolor="#ffcc99" width=15% height=10%> <a href=file:///C:\Girl.gif

target=three> Девушка</a>

</table>

</HTML>

## СОЗДАНИЕ ИНТЕРАКТИВНЫХ HTML-ДОКУМЕНТОВ

**Создание HTML-формы**

Формы в HTML документе предназначены как для отображения информации, так и для ее сбора (ввода). Отображение и ввод информации в формах производится в специальных полях. Формы в сочетании со встроенными или CGI сценариями позволяют сделать HTML документ интерактивным. Пользователь веб-страницы, в которую помещена форма, получает возможность взаимодействовать с формой, изменять вид и содержание веб-документов, а также использовать сразу несколько окон.

Для создания HTML-форм предназначена дескрипторная пара <FORM> </FORM>. Все, что расположено внутри этой дескрипторной пары, является либо описанием полей формы, либо текстовыми или графическими элементами. Текстовые и графические элементы применяются для сообщения пользователю, информация какого характера отображается (вводится) в то или иное поле.

Синтаксис открывающего дескриптора <FORM>:

<FORM NAME=”имя\_формы” METHOD=”метод” ACTION=”имя\_сценария”>

Атрибут NAME служит для определения имени формы, которое служит для ее идентификации.

Атрибут METHOD определяет метод, посредством которого информация, полученная от формы, передается CGI сценарию. Атрибут может принимать значения GET или POST.

Атрибут ACTION определяет, с каким сценарием должна быть связана форма. Параметр “*имя\_сценария”* представляет собой URL CGI сценария.

Если форма использует только внутренний сценарий, созданный на одном из языков сценариев, то атрибуты в дескрипторе <FORM> можно не указывать.

**Поля форм**

Установка полей формы внутри контейнера <FORM> осуществляется посредством одиночного дескриптора <INPUT>, который имеет следующий синтаксис:

<INPUT TYPE=”тип\_поля” NAME=”имя” VALUE=”значение\_по\_умолчанию”>

Атрибут TYPE указывает, поле какого типа помещается в форму. Атрибут NAME устанавливает имя поля (идентификатор). Атрибут VALUE устанавливает значение поля по умолчанию.

В HTML-форму можно помещать следующие типы полей:

*Текстовое поле* (TEXT) – предназначенное для ввода любой текстовой информации, которая может быть размещена в одной строке. Для текстового поля можно использовать два дополнительных атрибута: SIZE – устанавливает размер поля в количестве символов, если этот атрибут опущен, то браузер создаст поле длиной в 20 символов; MAXLENGHT – устанавливает максимальное количество символов, которое может быть введено в это поле.

*Скрытое текстовое поле* (HIDDEN)- применяется в случаях, когда необходимо передать CGI сценарию какую-либо информацию, но, при этом, нужно исключить для пользователя возможность изменять ее.

*Поле пароля* (PASSWORD) – специальный тип текстового поля. Он обрабатывается таким же образом, что и текстовое поле, но при этом браузер не отображает на экране вводимую в него информацию.

Поле *“Флажок”* (CHECKBOX)- отображается в виде небольшого квадрата, который может быть пуст (поле не установлено), либо перечеркнут (поле установлено). Пользователь может устанавливать это поле щелчком мыши.

Поле *“Переключатель”* (RADIO) – подобно полю CHECKBOX, но с тем отличием, что если в форме присутствует несколько переключателей с одним и тем же именем, то включенным может быть только одно из них.

Поле *“Кнопка”* (SUBMIT) – служит для подачи браузеру команды на передачу всей собранной в форме информации на сервер для передачи ее CGI сценарию.

Поле *“Кнопка”* (RESET) – служит для подачи браузеру команды на аннулирование всех введенных данных и установку формы в исходное состояние.

Поле *“Кнопка”* (BUTTON)- служит для подачи браузеру команды на вызов сценария, подготовленного на языке сценария.

Кроме дескриптора <INPUT> есть еще два парных дескриптора, с помощью которых в форму помещаются поля, это дескриптор <TEXTAREA> и <SELECT>.

Дескриптор <TEXTAREA> определяет поле для ввода нескольких строк символьной информации. Его синтаксис:

<TEXTAREA NAME=имя\_поля ROWS=rr COLS=cc>

Текст, выводимый по умолчанию

</TEXTAREA>

Атрибут ROWS устанавливает количество выводимых (вводимых строк), атрибут COLS указывает количество символов в каждой строке.

Дескриптор <SELECT> помещает в форму раскрывающийся список. Он имеет синтаксис:

<SELECT NAME=имя\_поля SIZE==n MULTIPLE>

<OPTION> элемент списка 1

<OPTION> элемент списка 2

. . .

<OPTION> элемент списка n

</SELECT>

Атрибут SIZE указывает количество элементов списка (строк). Атрибут MULTIPLE устанавливает режим, при котором пользователь может выбрать сразу несколько элементов списка. Дескриптор <OPTION> служит для задания одного элемента списка.

Все поля формы представляют собой объекты, которые могут реагировать на различные события. Обработка события осуществляется сценарием – специально созданной для этой цели программой.

**Разработка сценариев с использованием языка сценария**

Сама по себе форма, технология создания которой в HTML-документе рассмотрена выше, еще не обеспечивает режим интерактивной работы с веб-страницей и соответствующую обработку информации, введенной в поля форм. Для обеспечения такого режима необходимо создать сценарии (scripts). Различают два вида сценариев в зависимости от того, где он выполняется: CGI сценарии (Common Gateway Interface –стандартный шлюзовой интерфейс) и сценарии, помещаемые в HTML-документ. Их отличие заключается в том, что CGI сценарии выполняются на веб-сервере. Для их вызова в HTML-документе помещается ссылка в виде URL. При вызове такого сценария браузер передает на сервер содержимое полей форм, которые программа сценария использует в качестве исходных данных и обрабатывает их. Результат обработки сервер отправляет браузеру компьютера, вызвавшему сценарий. После получения результата браузер помещает его в соответствующее поле формы. Такие сценарии разрабатываются на каких- либо языках программирования, например С++ или других, и хранятся на сервере в специально отведенных для этой цели каталогах.

Сценарии, помещаемые в HTML-документ, в отличие от CGI сценариев выполняются непосредственно в браузере. Таким образом, отсутствует процесс передачи информации, содержащейся в полях форм, для ее обработки на Web-сервер и передача результата от сервера обратно.

Сценарии, помещаемые в HTML-документ представляют собой небольшие программы, разработанные на языках сценариев. Языки сценариев обеспечивают такое управление браузером, которое позволяет ему выполнять обработку информации из полей форм не обращаясь к веб-серверу. Такие сценарии будем называть в дальнейшем просто сценариями.

Для подготовки сценариев в настоящее время применяются два языка программирования – JavaScript и VBScript, которые называют языками сценариев. Технология разработки сценариев далее будет рассмотрена применительно к VBScript.

Положим, что на веб-странице нужно создать пользователю возможность для пересчета курса валют. HTML -документ в окне браузера должен выглядеть так, как изображено на рис. 19. Для этого необходимо создать HTML-документ, содержащий код, представленный в Примере 13 и сохранить его в формате htm. Затем открыть документ в окне браузера и проверить корректность выполнения вычислений.

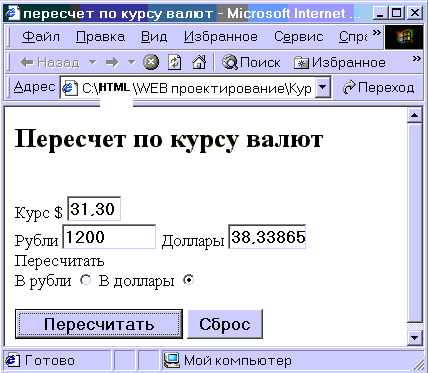


Рисунок 19

**Пример 13:**

<HTML>

<HEAD>

<TITLE>пересчет по курсу валют </TITLE>

</HEAD>

<BODY>

<!-- дескриптор, определяющий язык сценария-- >

<script language=vbscript>

<!— процедура сценария -- >

sub KURS()

dim a

a=cbool(kursval.shet(1).checked)

If a=true then

kursval.dollar.value =kursval.rub.value/kursval.kurs.value

else

kursval.rub.value=kursval.kurs.value\*kursval.dollar.value

end if

end sub

</script>

<h2>Пересчет по курсу валют</h2> <br>

<!—Создаем форму -- >

<form name=kursval>

<!—поле для ввода курса -- >

Курс $ <INPUT type=text name=kurs size=5> <BR>

<!—поле для ввода (вывода) количества рублей -- >

Рубли <INPUT type="text" Name=rub size=10>

<!—поле для ввода (вывода) количества долларов -- >

Доллары <INPUT type="text" Name=dollar size=8> <br>

Пересчитать <BR>

<!—переключатели для указания направления пересчета –

рубли в доллары или наоборот -- >

В рубли <INPUT type="radio" Name=shet>

В доллары <INPUT type="radio" Name=shet checked > <br><BR>

<!—кнопка для запуска вычислений (запуск сценария) -- >

<INPUT type="button" value="Пересчитать" ID=Button1 onclick=KURS()>

<!—кнопка сброса -- >

<INPUT type="reset" value="Сброс" ID=Reset1>

</form>

</BODY>

</HTML>

Сценарий в этом примере выполняется процедурой, которая имеет имя KURS. Сценарий вызывается, если происходит событие Click для поля формы типа BUTTON, которому присвоено имя BUTTON1. Он выполняет следующие операции:

проверяет, какой переключатель включен – пересчитать в рубли или пересчитать в доллары.

в зависимости от состояния переключателей выполняет пересчет.

Данные для расчета выбираются из полей формы типа TEXT, результат вычисления также помещается в поле формы.

Из приведенного текста сценария видно, что для подготовки даже простого сценария, нужно иметь некоторые представления об языке программирования сценариев.

## ОБЗОР ПРИЛОЖЕНИЯ FRONT PAGE

Несмотря на то, что большинство приложений Мicrosoft Office XP имеют собственные возможности для создания веб-страниц и публикации документов в Интернете, конечным инструментом Office для веб-издательства является приложе­ние Front Page 2002. Данное приложение предлагает широкий набор средств и инструментов для создания, управления и публикации веб-сайтов. В него также интегрирован редактор веб-страниц с широким диапазоном возможностей.

Приложение Front Page позволяет:

1. Быстро создавать веб-сайты с помощью мастера или шаблона. Каждый мастер или шаблон создает совокупность веб-страниц, выполняющих опреде­ленную функцию, например, совместное использование веб-сайта, проведение дискуссий в режиме оn-line, предоставление личной информации, управле­ние групповым проектом. Мастер или шаблон последовательно форматирует страницы, позволяя формировать доступ, добавлять переходы, графические изображения, текст, а также другие элементы веб-страниц.
2. Работать со знакомым интерфейсом. Для работы с папками и файлами веб­-сайта используется интерфейс, схожий с интерфейсом браузера Мicrosoft Explorer.
3. Просматривать информацию о компонентах веб-сайта. О статусе файлов, гиперссылок и другие элементов веб-страниц можно узнавать непосредствен­но в приложении Front Page. Статистические сведения о совместной работе группы лиц с сайтом можно найти в представлении Reports (Отчеты).
4. Использовать графические представления веб-страниц. Для просмотра, из­менения и проверки гиперссылок, связывающих веб-страницы в единое це­лое, навигации по веб-сайту и переходам к другим веб-сайтам можно использовать специальные графические представления веб-страниц.
5. Применять при создании веб-сайта стилистические темы. Каждая тема фор­мирует определенный стиль фона, текста, списков, баннеров, меню навигации и других элементов для каждой страницы веб-сайта.
6. Управлять проектами. Вы можете управлять совместными проектами, со­ставлять список заданий для каждого члена команды, назначать, сортировать, управлять доступом и выполнять другие операции по администрированию.
7. Открывать и редактировать веб-сайт непосредственно на сервере. Кроме того, приложение Front Page позволяет поддерживать работу с копией веб-сай­та на локальном компьютере или сетевом диске, с последующей публикацией этого веб-сайта на сервере.

Для создания и редактирования веб-страниц можно использовать следующие воз­можности Front Page 2002:

1. Вносить изменения на страницы веб-сайта или создавать новые страницы.
2. Использовать шаблоны или мастера, главное преимущество которых — это создание страницы с уже готовым базовым набором элементов и возможность последовательного форматирования этих элементов. Некоторые мастера или шаблоны создают специальные страницы для библиографических списков, таблиц или форм, другие генерируют страницы с определенным порядком расположения столбцов и т. д.
3. Одновременно вносить изменения в совместный веб-сайт. При совместной работе над проектом веб-сайта каждый участник команды может дополнять веб-сайт библиотекой документов, списками и другой необходимой информацией.
4. Форматировать элементы веб-страниц. Приложение Front Page позволяет форматировать текст, символы, графические изображения, видеоклипы, границы, гиперссылки, таблицы, формы, фреймы и другие стандартные элементы веб-страниц.
5. Форматировать веб-сайт при помощи каскадных таблиц стилей. Вы можете изменять, создавать и использовать СSS-стили для форматирования веб-страниц.
6. Добавлять на веб-страницы новые элементы. Приложение Front Page позволяет добавлять на веб-страницы такие элементы, как поля дат и комментарии, динамические эффекты, анимационные эффекты, например, динамические кнопки (hover buttons), активизирующиеся при наведении на них указателя мыши), поисковые формы, таблицы, счетчики посещений, фотогалереи, переходы, статистику совместной работы и т. п.
7. Добавлять управляющие элементы. Вы можете добавлять на веб-страницу апплеты Java, управляющие элементы АсtiveX и др.
8. Проверять орфографию. С помощью приложения Front Page 2002 можно провести проверку правописания содержимого веб-страниц, внести грамматические и стилистические изменения в текст.
9. Осуществлять поиск и замену текста. Вы можете проводить поиск заданных текстовых вхождений на веб-страницах, заменять найденные вхождения другими.
10. Просматривать веб-страницы и выводить их на печать. Используя возможности Front Page, вы легко можете просматривать веб-страницы и выводить на печать их отдельные части.

## ПУБЛИКАЦИЯ ВЕБ-СТРАНИЦ

Для того чтобы информация, помещенная на ваших веб-страницах, стала доступной для широкого круга пользователей, следует поместить их на веб-сервер и выполнить процедуру регистрации. Сервер обеспечит сохранность страниц и доступ к ним любого пользователя сети Интернет.

Первым шагом в решении задачи по размещению веб-страниц на веб-сервере является поиск провайдера, который может хранить ваши файлы на своем компьютере. Исследуйте в этом направлении следующие варианты.

Если провайдер Интернета, к которому подключен ваш компьютер, или организация, в которой вы работаете или учитесь, имеет веб-сервер, попытайтесь выяснить, не согласится ли он разместить созданные вами веб-страницы.

Существуют так называемые провайдеры присутствия, которые специализируются на публикации информации различных пользователей на своем веб-сервере, выделяя для этой цели сдой ресурс за арендную плату.

Можно разместить свой веб-узел в одной из коммерческих диалоговых служб, например Россия-Он-Лайн или Microsoft Network. Для этого подключитесь по сети, используя свою учетную запись, и ознакомьтесь с информацией о предоставляемых услугах и условиях.

Если ваш компьютер подключен к Интернету с использованием протоколов SLIP или РРР, то его можно сделать персональным веб-сервером. Тогда компьютер должен иметь постоянный IP-адрес и быть круглосуточно подключенным к Интернету. В этом случае вам придется платить провайдеру за круглосуточное подключение.

Самым доступным и дешевым способом получения веб-пространства и публикации веб-страниц является размещение их на веб-сервере провайдера. Если ваша информация не является коммерческой, то скорее всего плата за использование ресурса будет символической.

После выбора провайдера Интернета нужно переместить HTML-код созданных веб-страниц на его сервер. Эта операция потребует использования программы FTP (протокола передачи файлов). Программа FTP позволяет скопировать нужные файлы с компьютера разработчика на веб-сервер. С помощью FTP-программы нужно создать соответствующую запись в файле конфигурации для подключения к серверу провайдера. При выполнении этой операции следует учитывать приведенные ниже моменты.

* Если адрес сервера вашего провайдера имеет вид, Например http://www.fa.ru, то его FTP-адрес будет ftp.fa.ru.
* Запустив программу FTP, введите свой Login и пародь.

Подробности о расположении каталогов веб-сервера следует уточнить у службы технической поддержки провайдера.

После запуска FTP-программы отобразится окно регистрации, в которое необходимо ввести требуемую информацию. После регистрации на экране отобразится список каталогов. Его содержимое зависит от конкретных файлов и папок, размещенных на сервере.

Для копирования файлов в окне Local System (локальная система) следует выбрать файлы, которые нужно скопировать на сервер провайдера, а в окне Remote System (удаленная система) выбрать каталог, в который нужно поместить копируемые файлы. При копировании на сервер содержимое файлов локального компьютера не должно изменяться, поэтому в FTP-программе следует использовать режим передачи двоичных (бинарных) файлов.После копирования файлов на сервер провайдера их можно просмотреть, если известен URL. Его можно узнать у службы технической поддержки провайдера.

World Wide Web не имеет четкой структуры. Это обстоятельство, с одной стброны, является недостатком, а с другой - существенным достоинством, которое отражает открытость Интернета. Именно по этой причине после размещения ваших веб-страниц на сервере провайдера они не сразу становятся доступными для других пользователей, поскольку поисковые системы еще не знают, где они размещены и каковы их поисковые признаки. Чтобы веб-страницы стали доступны для других пользователей Интернета, их нужно зарегистрировать в Специальных каталогах вебстраниц и системах поиска. Эти каталоги содержат перечни вебстраниц, в которых может выполняться поиск по какой- либо теме. Ниже приведены сведения о некоторых каталогах и поисковых системах, которые пользуются у пользователей наибольшей популярностью.

Таблица 7

**Сведения о некоторых каталогах и поисковых системах**

| Каталог, поисковая система | URL |
| --- | --- |
| Yaho! | www.yaho.com |
| Web Crawler | www.webcrawler.com |
| Lucos | www.hicos.com |
| HotBot | www.hotbot |
| AltaVista | www.altavista4igital.com |
| InfoSeek | www.infoseek.com |
| Yandex | www.yandex.ru |
| Rambler | www.rambler.ru |

Операцию регистрации можно выполнить посредством какй-либо из поисковых систем. Каждая из них использует свою процедуру регистрации, во время которой нужно заполнить специальную форму. Так, например, можно использовать для регистрации поисковую систему Yandex.

Для регистрации в Yandex нужно вызвать поисковую систему по адресу http://www.yandex.ru. После загрузки первой страницы найти ссылку <Зарегистрироваться> и щелкнуть по ней - загрузится страница "Паспорт" (рис.20). Выполните предлагаемые для регистрации шаги.

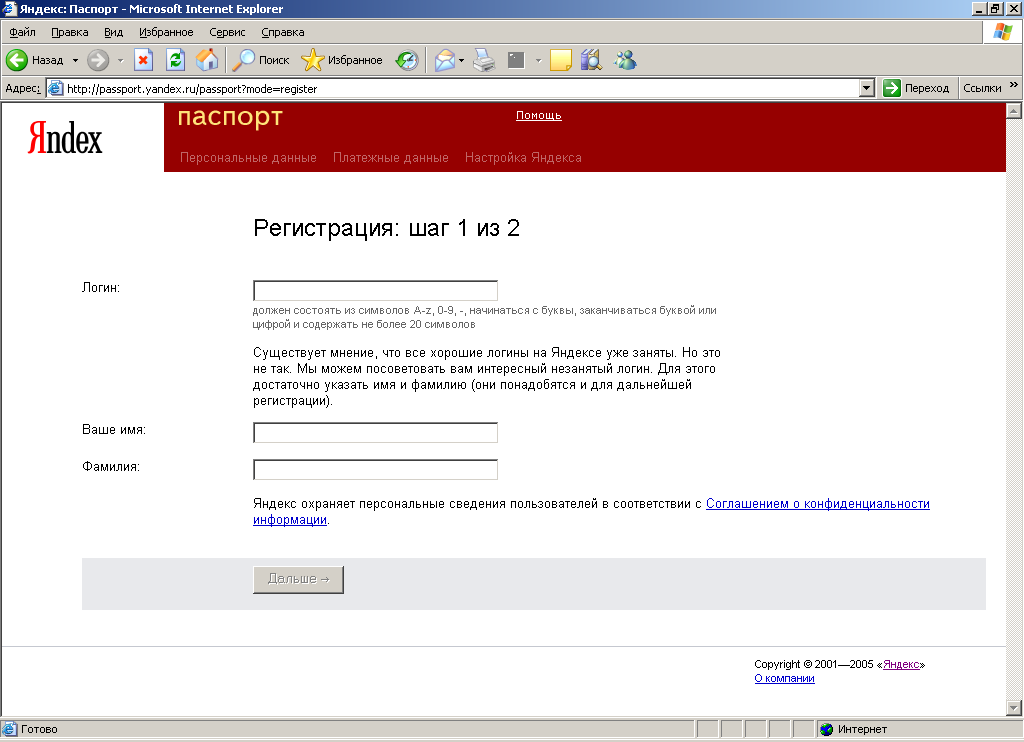


Рисунок 20

Заполняя форму, вы тем самым готовите сообщение для технической службы поддержки системы поиска или службы каталогов. Это не означает, что регистрация после заполнения формы будет выполнена автоматически. Фактически регистрацию выполняет уполномоченное для этой цели лицо на основе информации, которую вы указали в заполненной форме. Поскольку заявки на регистрацию в соответствующие службы поступают в значительном количестве, то процесс регистрации в службе каталогов может оказаться достаточно длительным - от нескольких дней до нескольких недель. После выполнения регистрации вас должны известить об этом по электронной почте. После получения сообщения о регистрации следует проверить, где и как вас зарегистрировали, и исправить возможные ошибки в описаниях и определениях.

Если созданная вами веб-страница посвящена определенной теме, то вы должны подумать, следует ли разместить о ней рекламу в группах новостей UseNet. В UseNet существуют отдельные группы новостей, посвященные практически любым темам. В них можно анонсировать новые веб-серверы. Так, например, группа новостей соmр.infosystem.www.announce служит для размещения сообщений о создании новых веб-страниц или об изменении содержимого существующих. Эти сообщения рассылаются пользователям Интернета. Недостатком такого способа популяризации вашей веб-страницы является то, что читает материалы подобных групп сравнительно небольшое количество пользователей. Другим способом популяризации является помещение URL созданных веб-страниц в шаблон надписи сообщений электронной почты и групп новостей. Существуют и иные формы популяризации веб-страниц.

# Литература:

* 1. Мельников П.П. Технология разработки HTML-документов: Учебное пособие.-М.:Финансы и статистика, 2005.
  2. Хэлверсон М., Янг М. Эффективная работа: Office XP.-СПб.: Питер, 2005.

**СОДЕРЖАНИЕ**

[1 Организационно-методический раздел 2](#_Toc139969479)

[1.1 Цель учебной дисциплины 2](#_Toc139969480)

[1.2 Задачи учебной дисциплины 2](#_Toc139969481)

[1.3 Место учебной дисциплины в профессиональной подготовке дипломированного специалиста 2](#_Toc139969482)

[1.4 Требования к уровню освоения содержания учебной дисциплины 2](#_Toc139969483)

[2 Объем дисциплины и виды учебной работы 3](#_Toc139969484)

[3 СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ 4](#_Toc139969485)

[3.1 ОСНОВНЫЕ СВЕДЕНИЯ О ТЕХНОЛОГИЯХ WEB – ПРОЕКТИРОВАНИЯ 4](#_Toc139969486)

[3.2 ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА И СТРУКТУРА HTML-ДОКУМЕНТА 9](#_Toc139969487)

[3.3 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ ВВОДА И РЕДАКТИРОВАНИЯ HTML-ДОКУМЕНТОВ 11](#_Toc139969488)

[3.4 СОЗДАНИЕ WEB-ПРОЕКТА С ПОМОЩЬЮ МАСТЕРА WEB СТРАНИЦ И MS WORD 13](#_Toc139969489)

[3.5 РЕДАКТОР СЦЕНАРИЕВ 20](#_Toc139969490)

[3.6 ПРИМЕНЕНИЕ HTML ДЛЯ ОПИСАНИЯ ВЕБ-ДОКУМЕНТА 25](#_Toc139969491)

[3.7 СОЗДАНИЕ ИНТЕРАКТИВНЫХ HTML-ДОКУМЕНТОВ 42](#_Toc139969492)

[3.8 ОБЗОР ПРИЛОЖЕНИЯ FRONT PAGE 47](#_Toc139969493)

[3.9 ПУБЛИКАЦИЯ ВЕБ-СТРАНИЦ 49](#_Toc139969494)

[4 Литература: 52](#_Toc139969495)

**-----------------------------------------------------------------------------------------------------------------------**

[Российский портал информатизации образования](http://portalsga.ru/) [содержит: законодательные и нормативные правовые акты государственного регулирования информатизации образования, федеральные и региональные программы информатизации сферы образования, понятийный аппарат информатизации образования, библиографию по проблемам информатизации образования, по учебникам дисциплин цикла Информатика, научно-популярные, документальные видео материалы и фильмы, периодические издания по информатизации образования и многое другое.](http://portalsga.ru)

