

4. План практических занятий

Практические занятия по дисциплине не предусмотрены

[На начало документа](#)

[На содержание](#)

5. Методические указания по выполнению лабораторных работ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ

Л.Е.Рудельсон, М.М.Тверитнев

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

ПОСОБИЕ

к выполнению лабораторных работ

для студентов III курса

специальности 230101

дневного обучения

Москва – 2005

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ**

Кафедра вычислительных машин, комплексов, систем и сетей

Л.Е.Рудельсон, М.М.Тверитнев

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ

ПОСОБИЕ

к выполнению лабораторных работ

*для студентов III курса
специальности 230101
дневного обучения*

Москва – 2005

Р

Рецензент профессор Н.Н.Горнец
Рудельсон Л.Е., Тверитнев М.М.

Р

Пакеты прикладных программ: Пособие к выполнению лабораторных работ. – М.: МГТУ ГА, 2005.-48 стр.

Пособие издается в соответствии с учебным планом для студентов специальности 220100 дневного обучения.

Рассмотрено и одобрено на заседаниях кафедры ВМКСС 11.05.05 и методического совета 11.05.05.

Редактор Г.В.Токарева

Подписано в печать

Печать офсетная

усл.печ.л.

Формат 60х84/16

Заказ №

3,0 уч.-изд. л.

Тираж экз.

Московский Государственный Технический Университет ГА
125993 Москва, Кронштадтский бульвар, д.20

Редакционно–издательский отдел
125493 Москва, ул. Пулковская, д.6а

© Московский Государственный Технический
Университет Гражданской Aviации, 2005

РАЗРАБОТКА СТРУКТУРЫ ВЕБ-САЙТА. СОЗДАНИЕ ПРОСТЕЙШИХ ВЕБ-СТРАНИЦ

1. ЦЕЛЬ РАБОТЫ

Целью работы является получение навыков проектирования и тестирования (исследования) веб-сайтов с помощью редактора HTML-страниц. В качестве редактора используется пакет прикладных программ Microsoft Front-Page.

2. СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ВЕБ-САЙТА

Разработка структуры сайта необходима для:

- создания четкой и логичной схемы навигации;
- организации простой технологии внесения изменений при редактировании сайта.

Для достижения этих целей процесс создания структуры принято рассматривать в двух аспектах. Фактически проектируются две структуры: логическая и физическая. Логическая структура определяет, в какой последовательности материалы будут доступны пользователю, какие ссылки следует выбирать для доступа к информации, размещенной на сайте. Хорошо продуманная логическая структура гарантирует, что на поиск необходимых данных будет затрачено меньше времени, и что они всегда будут найдены. Для создания полноценной логической структуры достаточно следовать нескольким простым правилам:

- любой документ сайта должен оказываться доступным не более чем с помощью трех переходов с главной страницы сайта;
- все навигационные элементы должны отображаться сразу после загрузки страницы;
- все внутренние связи должны быть двунаправленными, то есть позволять перемещаться между документами в обоих направлениях;
- с любой страницы должен быть предусмотрен возврат на главную страницу сайта;
- названия рубрик и распределение материала между ними должно быть понятным каждому посетителю сайта.
- если сайт имеет более одного уровня навигации, он обязательно должен содержать навигационную карту.
- для удобства посетителей, каждый сайт должен иметь простую, четкую и логичную схему навигации.

Пример логической структуры веб-сайта показан на рисунке 1.1.

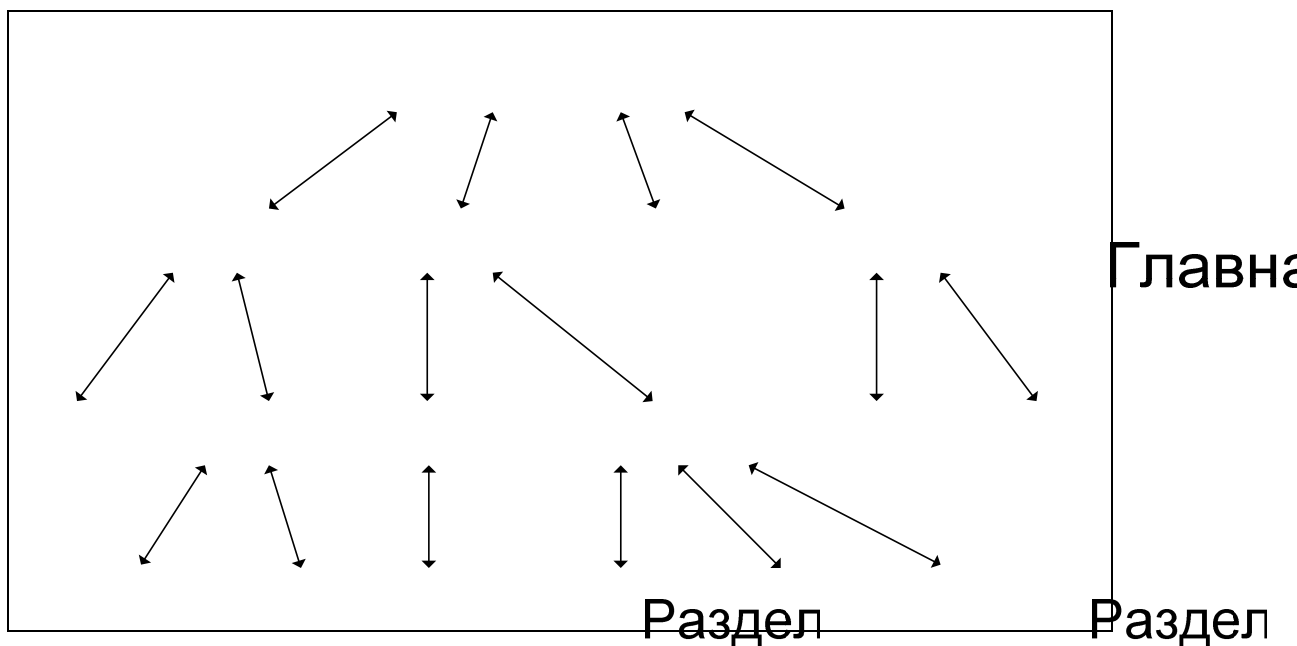


Рис. 1.1. Логическая структура сайта

Существует несколько видов логических структур сайта. Самая простая из них – линейная [1]. В ней страницы следуют одна за другой, и пользователь должен просматривать их последовательно, как слайд-шоу. Недостатков у такой структуры достаточно много. Как следствие, область ее применения ограничена. Она может использоваться на сайтах-презентациях и в онлайн-учебных пособиях. Самый простой вариант сайта с линейной структурой – набор страниц, из которых каждая ссылается на следующую и на предыдущую. Однако и здесь начинающие авторы допускают ошибки. Дело в том, что на каждой странице обязательно должны присутствовать соответствующий заголовок и ссылка на первую страницу. В противном случае посетители, попавшие в середину сайта – например, в результате обращения поисковой системы, не смогут сориентироваться в ситуации, и покинут проект разочарованными. Кроме того, полезно показывать общее число страниц в отображаемом документе и выделять номер той из них, которая воспроизводится в данный момент времени.

Следующим вариантом структуры сайта является линейная структура с альтернативами и вариантами. Ее основой остается простое линейное размещение страниц. Однако на сайтах, построенных по данному принципу, посетители могут проявить некоторую инициативу, позволяющую облегчить поиск нужной информации. Под альтернативами здесь понимается выбор между двумя ветвями перемещения. Чаще всего подобная структура используется для сбора информации о посетителе. Примером может служить процесс регистрации клиента на сайте фирмы, оказывающей определенные услуги. Работа всегда начинается со стартовой страницы. Однако затем частным лицам предлагается ввести одну информацию, а представителям организаций – другую. Возможно, ветви альтернатив в дальнейшем вновь смыкаются, и все пользователи попадают на одну и ту же страницу.

Третий вариант носит название линейной структуры с ответвлениями.

Это тоже контролируемая структура перемещения по ресурсам, наглядный аналог которой – дорога с ответвляющимися от нее тупиковыми тропами. Иными словами, посетитель последовательно переходит с одной страницы на другую. Если информация, размещенная на любой из них, его заинтересовала, и он желает ознакомиться с ней подробнее, то ему предоставляется возможность перейти на ответвление, а потом вернуться обратно на основную "дорогу". Главное преимущество такой структуры состоит в том, что к ней легко перейти с обычного линейного размещения страниц. Часто бывает, что однажды созданный сайт перестает удовлетворять возросшим требованиям, а глобальная переделка по тем или иным причинам нежелательна. В этом случае автор может без затруднений расширить свой проект. И его сайт не утратит четкости логических связей.

Следующий вариант – древовидная структура (рис. 1.1) – универсальный и в большинстве приложений наиболее предпочтительный способ размещения страниц. Она хорошо зарекомендовала себя для создания практически любых типов сайтов. Пользователь, попадая на главную страницу, оказывается перед выбором, в каком направлении двигаться дальше. После перехода в нужный раздел, он выбирает необходимый подраздел, затем пункт (параграф) и т.д. У древовидной структуры достаточно много достоинств, однако и она не лишена недостатков. Остановимся на главном из них.

В древовидной структуре очень сложно соблюдать баланс между глубиной и шириной. Формальные критерии либо тривиальны, либо рассчитаны на узкие группы пользователей. Успех зависит от опыта, интуиции и квалификации автора. Если "дерево" сайта будет расти только вглубь, то посетителям, чтобы найти необходимую информацию, придется загрузить и просмотреть слишком много страниц. Естественно, это занимает много времени и раздражает. Если же создается очень широкая древовидная структура, то приходится тратить время по другой причине – для выбора нужной ветви поиска. Использование древовидной структуры сайта вынуждает постоянно следить за ее разрастанием и придерживаться золотой середины. Качество работы автора зависит от его мастерства, а не от соблюдения формальных правил.

Еще одним вариантом является решетчатая структура. Эта структура заметно сложнее рассмотренных ранее. В ней все страницы также размещаются в различных ветвях. Однако пользователю дается возможность перемещаться по ним не только вертикально (вверх и вниз), но и горизонтально (то есть между ветвями разных уровней). Используется «решетка» в основном только в каталогах. При этом перемещение между ветками на глубинных уровнях осуществляется с помощью ссылок на рубрики в других разделах.

Использование решетчатой структуры в других проектах считается нецелесообразным. Во-первых, она относительно сложна в реализации. Во-вторых, обращаться с «решеткой» нужно с очень большой осторожностью. В противном случае в схеме возникают непредусмотренные связи, и поиск информации приводит к непредсказуемым результатам.

Физическая структура не влияет на просмотр страниц посетителями и

служит в основном для удобства создателя сайта при его редактировании, позволяя легко найти нужный файл (документ) [2]. При проектировании физической структуры разработчик может ненадолго забыть о пользователях и немного подумать об удобствах сопровождения сайта. Обычно распределяют отдельные папки для каждого из его разделов и подразделов. Внутри папок также создают отдельные папки для вспомогательных изображений, отдельные папки для мультимедиа-информации (музыка, видео, и т.п.) – то есть файлы сайта разделяются по функциональным признакам. В простом статическом сайте можно четко определить несколько групп файлов:

- страницы сайта, представляющие собой HTML-файлы;
- таблицы стилей;
- клиентские скрипты;
- графические файлы, используемые в дизайне сайта;
- файлы для копирования посетителями.

Особое место занимает вопрос, связанный с выбором имен файлов. Логично и понятно названный файл позволяет сэкономить время при обновлении сайта. Кроме того, при передаче поддержки сайта другому сотруднику, будет гораздо легче объяснить, где находится та или иная информация. Возможно, на первых порах это кажется неважным, однако опыт показывает, что разбираться в структуре неудачно именованного сайта довольно затруднительно.

Прежде всего – не следует называть файлы безликими именами, такими как *page1.htm*, *123.htm* и т.п. Необходимо, чтобы при взгляде на список файлов сразу становилось понятно, что в них содержится. Другими словами, называть файлы следует исходя из содержательного смысла документа. Если страница излагает общую информацию о компании, то принято называть ее *about.htm*, а страницу с контактной информацией – *contacts.htm*.

В некоторых случаях для группы файлов могут использоваться похожие наименования, состоящие из базового имени и цифры. Например, если вы периодически пишете статьи на своем сайте, то файлы удобно называть последовательно: *article_1.htm*, *article_2.htm*, *article_3.htm* и т.д. Заметим, что в качестве разделителя базового имени и цифры используется знак подчеркивания. Он позволяет отделить номер статьи, что способствует быстрому нахождению нужного файла.

При формировании имен файлов можно взять за основу либо русский язык, либо английский. Так как использовать русские символы в именах файлов нельзя (причины этого описаны ниже), то при использовании русского в качестве базового языка необходимо писать имена файлов в транслитерации (русские слова латинскими буквами). В этом случае страницу с описаниями услуг можно назвать, например, *uslugi.htm*, а страницу с информацией об истории фирмы – *istoriya.htm*. Предпочтительно использовать в качестве основного языка английский, потому что в этом случае слова будут корректно индексироваться поисковыми системами, а значит, при корректном запросе по теме вашего сайта релевантность (степень соответствия поисковому за-

просу) вашей страницы будет выше. Не следует лишь забывать о правильном написании английских слов. Кроме этого, если имена файлов используют английский язык, то человек, который попал на вашу страницу и не знает русский язык, сможет догадаться, о чем идет речь. Выбрав язык для использования в именах файлов, придерживайтесь его в пределах всего сайта. Не допускайте чередования английских и русских имен файлов.

Для зависимых файлов, например, для иллюстраций к какой-либо странице, удобно использовать следующее правило: *имя графического файла образуется из названия страницы и идентификатора иллюстрации, разделенных знаком подчеркивания*. В качестве идентификатора иллюстрации может использоваться либо порядковый номер появления ее в документе, либо, что предпочтительнее, некоторый идентификатор, позволяющий легко опознать ее. Допустим, что наша статья называется *article_1.htm*, и в ней используются иллюстрации – пусть это будут фотографии сотрудников отдела и схема их взаимного подчинения. Тогда имена графических файлов, образованных согласно данному правилу, будут соответствовать конкретным фамилиям, например: *article_1_ivanov.jpg*, *article_1_petrov.jpg*, *article_1_sidorov.jpg*, а иерархия отношений – штатной схеме отдела: *article_1_scheme.gif*.

Для части графических файлов, преимущественно участвующих в создании дизайна сайта, удобно использовать префиксы и суффиксы. Если спроектировано графическое меню сайта, которое подсвечивается при наведении курсора манипулятора мышь, то все графические файлы, формирующие меню, можно предварять префиксом «m_», а к названиям изображений, которые появляются при наведении мышью, добавлять суффикс «_over». Тогда название графического пункта меню, например "О компании", будет состоять из двух файлов – «m_about.gif» и «m_about_over.gif».

Префиксы удобно добавлять к таким изображениям, которые могут изменяться в зависимости от типа страницы. Приведем несколько часто используемых префиксов:

- «bg_» (background) – для фоновых изображений;
- «m_» (menu) – для пунктов графического меню;
- «t_» (title) – для графических заголовков;
- «icon_» – для пиктограмм;
- «button_» – для графических кнопок, не являющихся элементами меню.

В качестве «корня» слова, образующего имя файла, удобно использовать название страницы, к которой относится данная графика. В приведенном примере «корнем» выступала страница *about*. Продолжим упражнения с ней и образуем имя файла, используемого как подложки этой страницы – получится имя *bg_about.gif*. Разумеется, имеет смысл это делать, если фоновый рисунок на разных страницах различный. Если же он везде одинаковый, то файл достаточно назвать просто *bg.gif*. Такова основа методики формирования имен файлов.

Перечисленные шаги выполняются с единственной целью – облегчить ориентирование в огромном количестве файлов, из которых состоит любой

современный сайт. В результате нетрудно найти нужную html-страницу и все относящиеся к ней иллюстрации, а также понять содержимое графических файлов, даже не заглядывая в них.

При создании имен файлов следует помнить об ограничениях, накладываемых со стороны операционных систем. Например, в формате UNIX-систем index.html и Index.html – разные файлы, а с точки зрения Windows – одни и те же. Нужна осторожность при использовании русских букв в именах файлов – преобразование символов из одной кодировки в другую превратит файл "галерея.htm" в "ЗБМЕТЕС.htm".

Рассмотрим инструментальное средство разработки сайтов, применяемое в лабораторных работах по курсу «Пакеты прикладных программ».

2.2. КРАТКОЕ ОПИСАНИЕ СРЕДЫ MICROSOFT FRONTPAGE

Достаточно подробное описание среды проектирования изложено в [3]. Помимо редактора HTML-страниц, она включает средства управления веб-сайтами. На рисунке 1.2 представлено окно системы после запуска. Все сред-

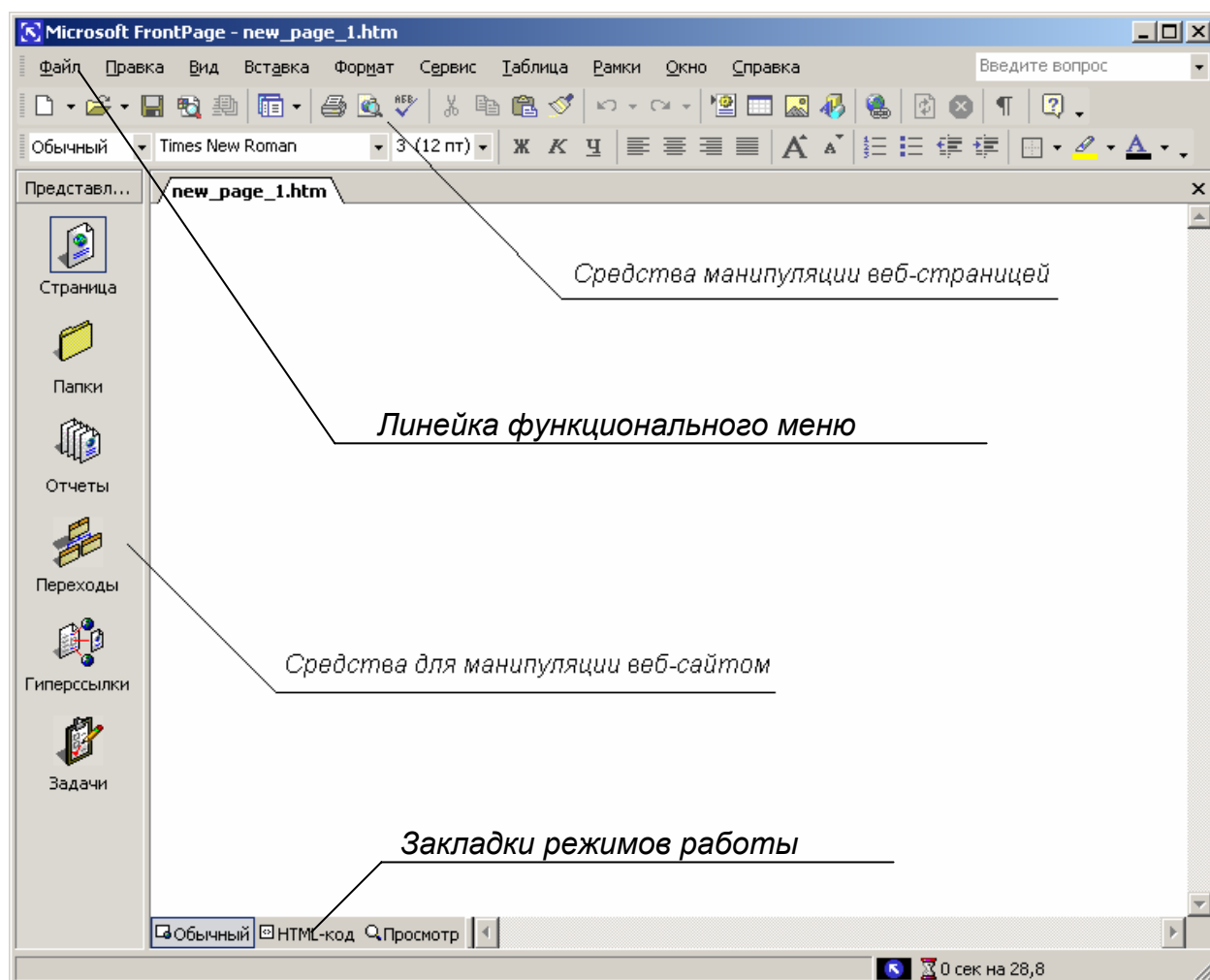


Рис. 1.2. Microsoft FrontPage 2000. Общий вид
ства манипуляции сайтами вызываются с помощью левой панели инструментов. Обращение к средствам редактирования страниц осуществляются через меню и верхнюю панель инструментов. Схема работы проста: задать структуру узла, упаковать в эту структуру страницы и создать связи между ними.

В состав FrontPage входят несколько заготовок сайтов с различной структурой страниц (рис. 1.3). При входе в систему автоматически создается веб-сайт, содержащий одну веб-страницу. Если вы желаете воспользоваться такой заготовкой – необходимо обратиться в меню «Файл»/«Новый»/«Сайт». Появится окно выбора заготовки сайта. Не нужно описывать все возможные заготовки. Для выполнения лабораторной работы достаточно использовать:

- One Page Web – веб-сайт с одной, заглавной, веб-страницей;
- Empty Web – упомянутый веб-сайт с одной безымянной веб-страницей;
- Personal Web – прототип персонального веб-сайта с несколькими страницами (фотоальбомом, списком интересов и т.д.).

Заготовки «Пустой узел» или «Одностраничный узел» помогают создать любой сайт «с нуля», не навязывая определенного стиля или структуры сайта. Для управления структурой сайта существует панель инструментов, расположенная слева (рис.1.2). По нажатию кнопки «Страница» появляется

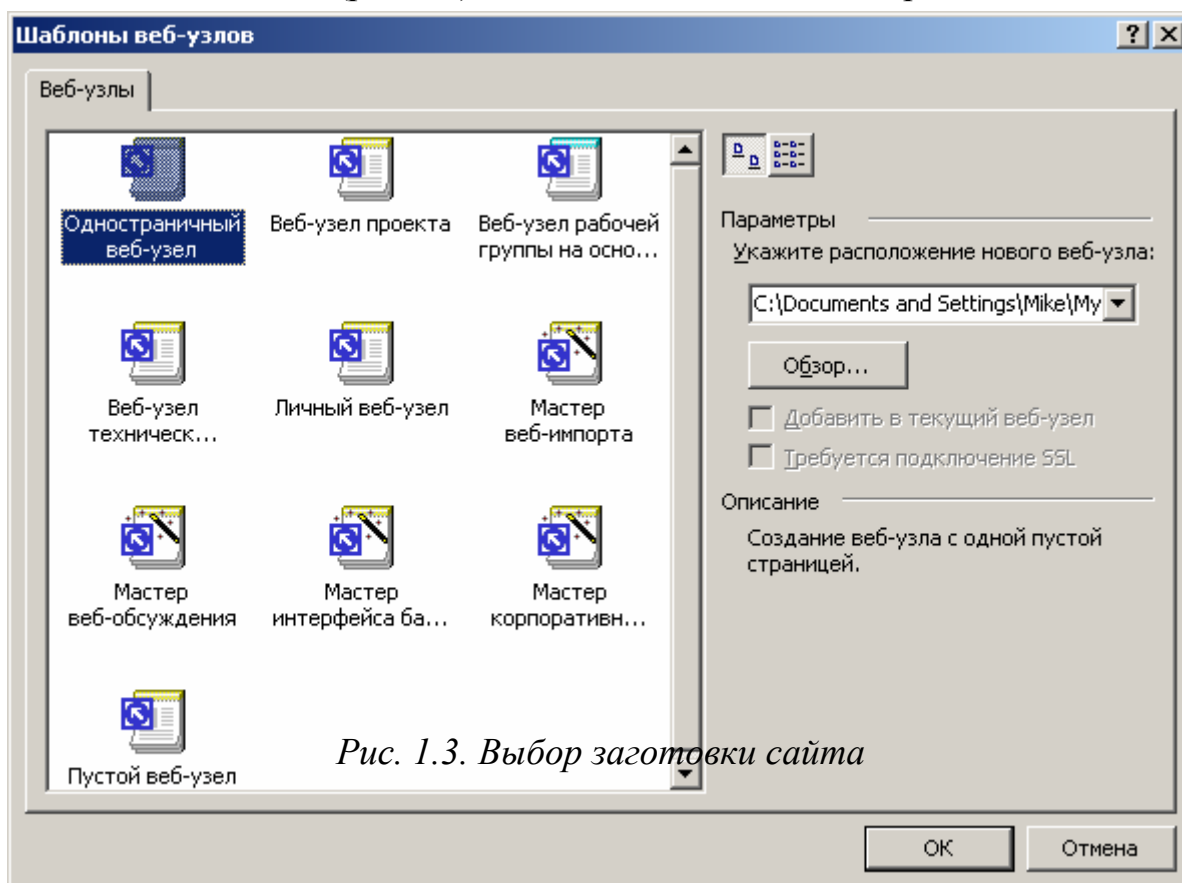


Рис. 1.3. Выбор заготовки сайта

редактор HTML-страниц. Кнопка «Папки» предъявит список папок, входящих в состав сайта. Папками можно манипулировать: создавать, удалять, перемещать. Все эти функции доступны через меню «Файл» и контекстное меню, вызываемое щелчком правой кнопки «мыши» на имени папки. С помощью подменю «Обратить в сайт» папку пакуют в отдельный, не зависимый от ныне редактируемого, сайт. Нажатие кнопки «Отчеты» вызовет список отчетов о страницах, входящих в веб-сайт. Двойной щелчок левой кнопки «мыши» по элементу списка выводит на экран более подробную информацию по выбранной категории отчета.

Перейдем к главному в системе Front-Page, к редактору веб-страниц. Собственно редактор состоит из трех закладок: «Нормальный», «HTML» и «Просмотр». Это – три режима работы со страницей.

Режим «Нормальный» предназначен для редактирования страницы практически точно в таком же виде, как она будет выглядеть в процессе просмотра Web-страницы пользователем сети Интернет. Другое название режима – «видишь то, что получишь», т.е. страница отображается практически точно в таком же виде, как она будет выглядеть в программе просмотра, такой как Internet Explorer. В этом режиме на экране представлены заголовок страницы, панель навигации в виде кнопок под заголовком, панель навигации в виде гиперссылок в левой части страницы – и текст в правой части. Параметры панели навигации в виде кнопок можно изменять. Эти изменения распространятся на все страницы сайта, поскольку созданные автоматически панели находятся в общих для всех страниц областях, расположенных по краям страницы. Можно удалить лишнюю панель навигации, можно добавить новую панель.

По второй закладке вы попадаете в окно редактора HTML. Здесь отображен код, который вы набирали клавишами в первом окне. Существует и обратная связь – все то, что вы написали во втором окне, найдет свое отражение в первом. Режим HTML позволяет редактировать страницу в текстовом представлении. Используйте этот режим, когда хорошо овладеете языком разметки гипертекстов.

По третьей закладке попадаем в окно просмотра, которая покажет, как ваша веб-страница будет выглядеть в браузере. Режим «Просмотр» позволяет не только просмотреть страницу в том виде, в котором ее отобразит браузер, но и проверить функции этой страницы, такие как: переходы по ссылкам, корректное отображение рисунков, использование объектов и форм и т.д.

Первая закладка редактора веб-страниц практически совпадает с окном Microsoft Word. Нас в большей степени привлекает меню, остановимся на нем подробнее.

Первое «собственное» меню в FrontPage – это «Обзор». Помимо дублирования левой панели инструментов, оно содержит пункт «Показать теги», щелчок по которому включает или выключает режим показа тегов при редактировании документа на первой закладке редактора. Меню «Вставка» служит для вставки в документ различных объектов: картинок, форм, компонентов, документов в формате других приложений Microsoft Office. Будьте внимательны при добавлении компонентов в формы – для того, чтобы они заработали, на веб-сервере должны быть установлены FrontPage Web Server Extensions. Дело в том, что Microsoft разработала свой метод обмена данными между клиентским компьютером и сервером, который реализован с помощью так называемых веб-ботов. Веб-ботом и является компонент FrontPage для обеспечения передачи данных. Эта технология поддерживается только веб-серверами, на которых установлен FrontPage Web Server Extensions. Веб-бот может вызвать функцию записи информации, введенной в форму, в файл, функцию поиска по сайту, функцию вывода даты последне-

го обновления страницы и т.д.

В меню «Формат» особого внимания заслуживают два пункта: «Динамик HTML эффекты» и «Тема». «Динамик HTML эффекты» вызывает окно добавления реакций на различные события, происходящие с произвольным элементом веб-страницы. В результате добавления эффекта в теле документа появляется заготовка функции на языке JavaScript, которую вы заполняете в соответствии со своим замыслом.

Пункт «Тема» предназначен для оформления веб-страницы аналогично одному из шаблонов (тем), предлагаемых создателями FrontPage.

3. ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

3.1. Подготовить сменный носитель (дискету, flash-накопитель) для сохранения результатов работы с целью их дальнейшего использования в работах №№ 2 – 4.

3.2. По рекомендованной литературе [1 – 7] ознакомиться с принципами работы в среде Microsoft FrontPage.

3.3. Ознакомиться с базовыми элементами языка HTML.

3.4. Выбрать тематику будущего веб-сайта.

3.5. Ознакомиться с дизайном и структурой существующих веб-сайтов.

3.6. Подготовить текст теоретических разделов отчета о работе.

3.7. Ответить на контрольные вопросы настоящих методических указаний.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. РАЗРАБОТКА ЛОГИЧЕСКОЙ СТРУКТУРЫ ВЕБ-САЙТА. Определите тип и количество информации, которая будет помещена на сайт. В результате должно получиться не менее пяти разделов. К полученным разделам обязательно добавьте разделы «Анкета» и «Гостевая книга».

4.2. ВЫБОР ВИДА ЛОГИЧЕСКОЙ СТРУКТУРЫ САЙТА. Выберите вид логической структуры веб-сайта в соответствии с рекомендациями, данными в разделе 2.1. Предпочтительной является древовидная структура сайта. Подобную структуру имеет сайт, описываемый ниже в качестве примера.

4.3. ЛОКАЛИЗАЦИЯ ПРОЕКТА. Средствами Windows создайте папку для хранения файлов вашего сайта. Создайте в ней подпапки в соответствии с разработанной физической структурой сайта. При разработке физической структуры рекомендуется создавать отдельную подпапку для каждого раздела (остальные рекомендации сформулированы в разделе 2.1).

4.4. ЗАПУСК MICROSOFT FRONTPAGE. Запустите среду создания веб-сайтов Microsoft Frontpage (Пуск – Программы – Microsoft Frontpage).

4.5. СОЗДАНИЕ СТРАНИЦ САЙТА. В появившемся окне редактора веб-страниц введите необходимую информацию.

4.5.1. Нажмите правую кнопку мыши и выберите меню «Свойства страницы». Пройдите по всем закладкам появившегося окна и установите все

необходимые вам свойства. Особое внимание обратите на последнюю закладку, на которой устанавливаются языковые параметры (рис. 1.4).

4.5.2. Введите информацию, необходимую для размещения на титульной странице (рис. 1.5). Сохраните полученный файл.

4.5.3. Создайте новую страницу. Расположите на ней элементы меню сайта в соответствии с созданной логической структурой (рис. 1.6).

4.5.4. Создайте остальные страницы сайта (кроме анкеты и гостевой книги) в соответствии с разработанной структурой. Наполните их информацией. На одну из страниц поместите крупный блок информации, преимущественно текстовой (рис. 1.7).

4.5.5. Позаботьтесь о том, чтобы сделать дизайн сайта привлекательным для посетителей, используя предоставленные вам изобразительные средства.

4.6. ЗАВЕРШЕНИЕ РАБОТЫ. Сохраните получившиеся страницы и перепишите созданные файлы на заранее подготовленный носитель. Выйдите из Frontpage, выключите компьютер.

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет о работе должен содержать следующие материалы в соответствии с полученным индивидуальным заданием:

5.1. Обоснование выбора структуры сайта (по материалам п. 2.1).

5.2. Схему логической структуры созданного сайта (по материалам п. 2.1).

5.3. Схему физической структуры созданного сайта (до уровня файлов – по материалам п. 2.1).

5.4. Словесную постановку задачи лабораторной работы (по материалам индивидуального задания).

5.5. Последовательность выполнения работ в среде Microsoft FrontPage для создания сайта (по материалам п. 2.2).

5.6. Формализацию выполненного индивидуального задания (разделение структуры и оформления создаваемых документов).

5.7. Ответы на контрольные вопросы и выводы по выполненной работе.

Рис. 1.4. Установка свойств страницы в редакторе

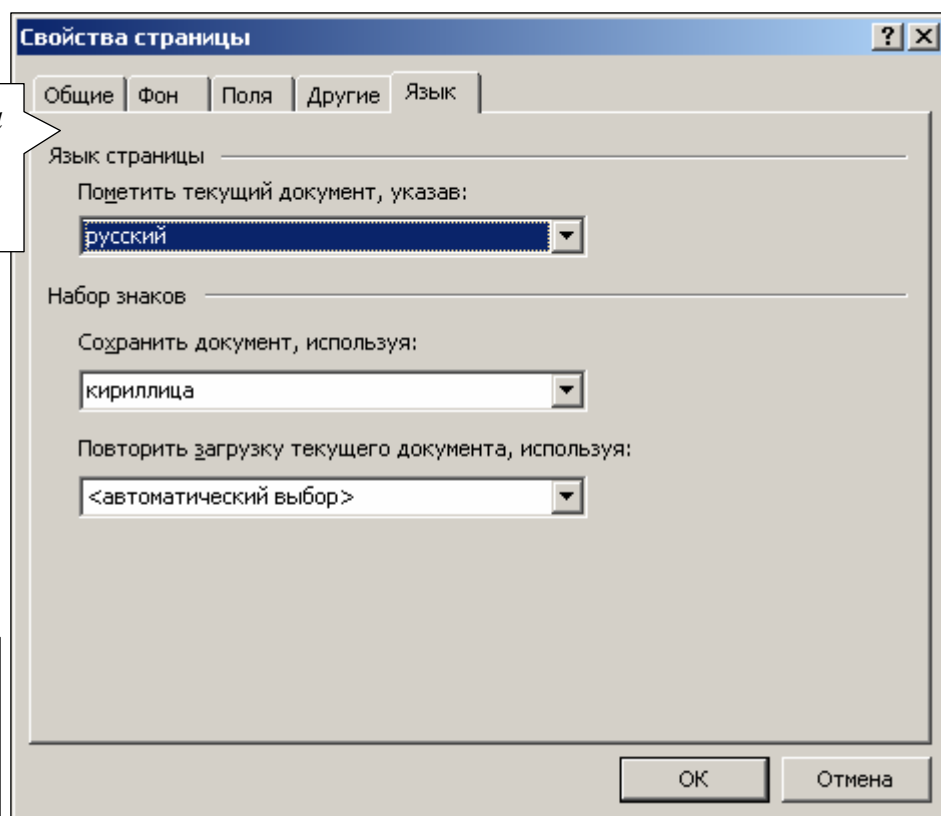
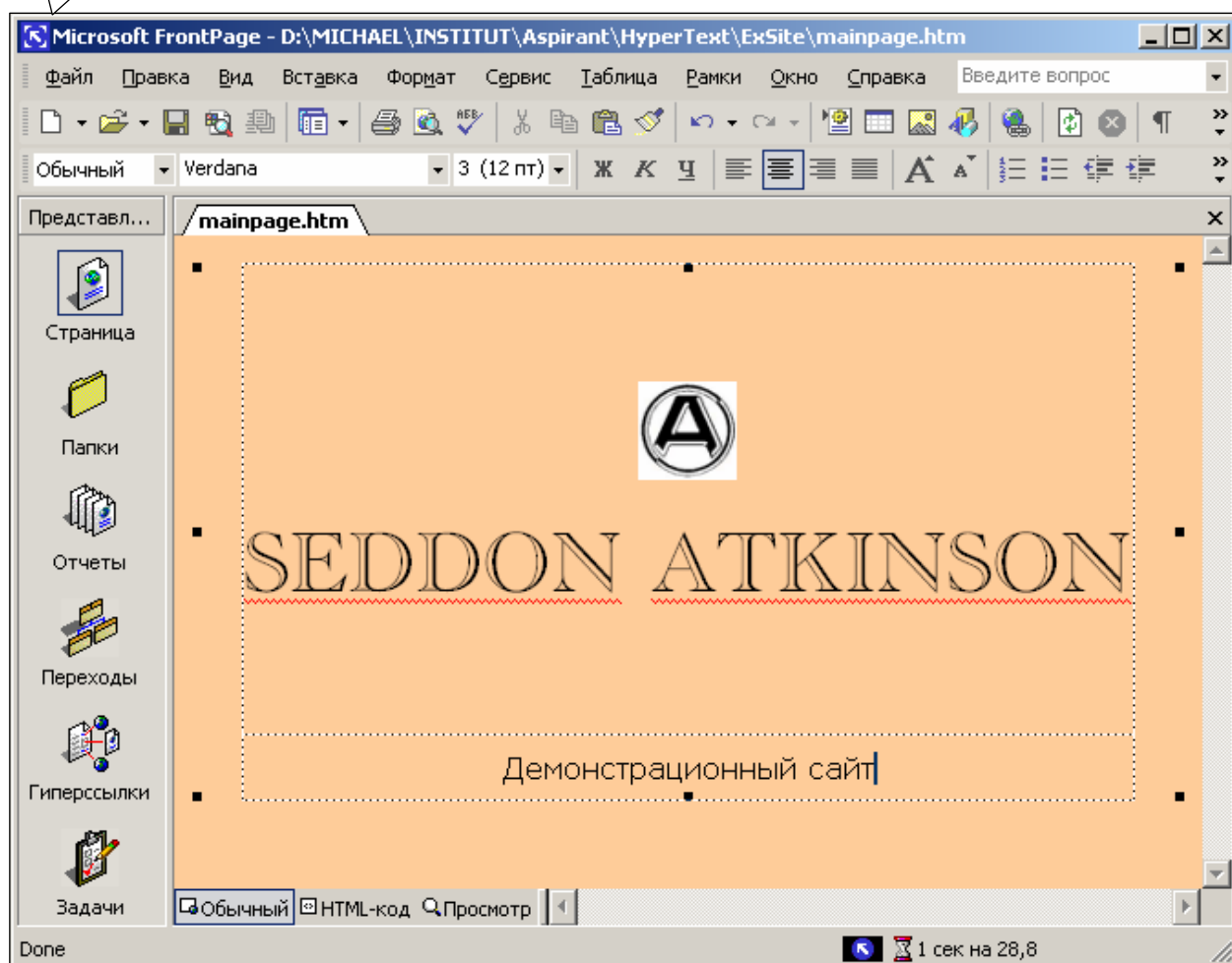


Рис. 1.5. Формирование титульной страницы



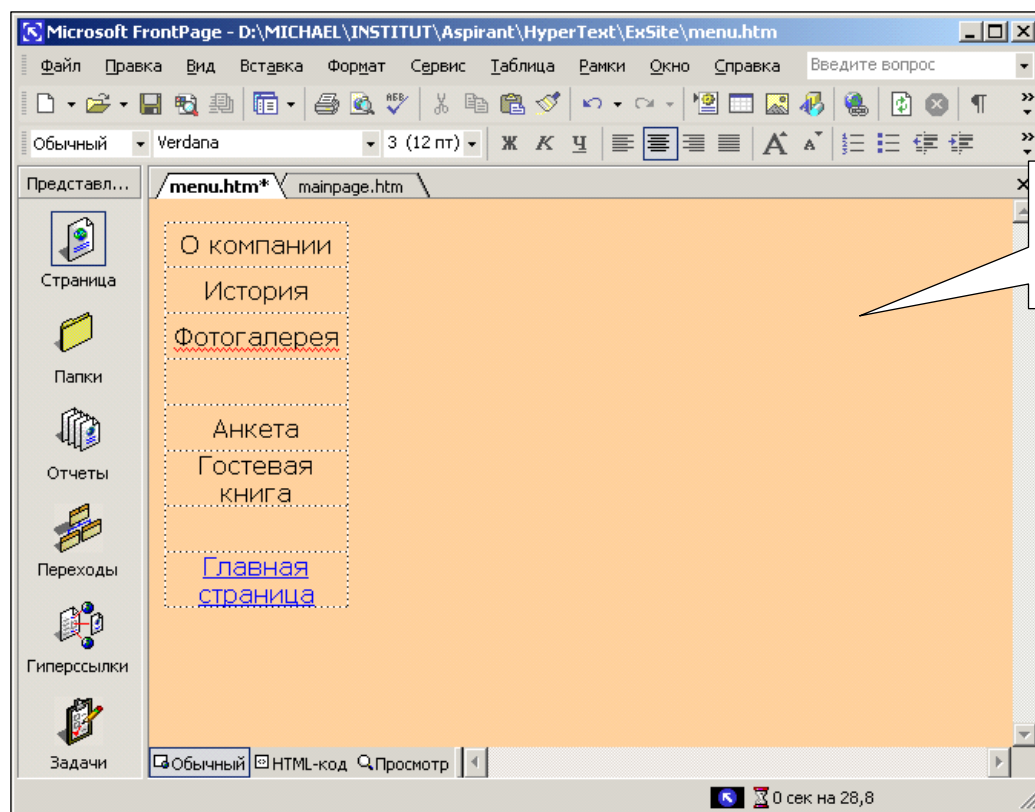


Рис. 1.6.
Разработка
меню сайта

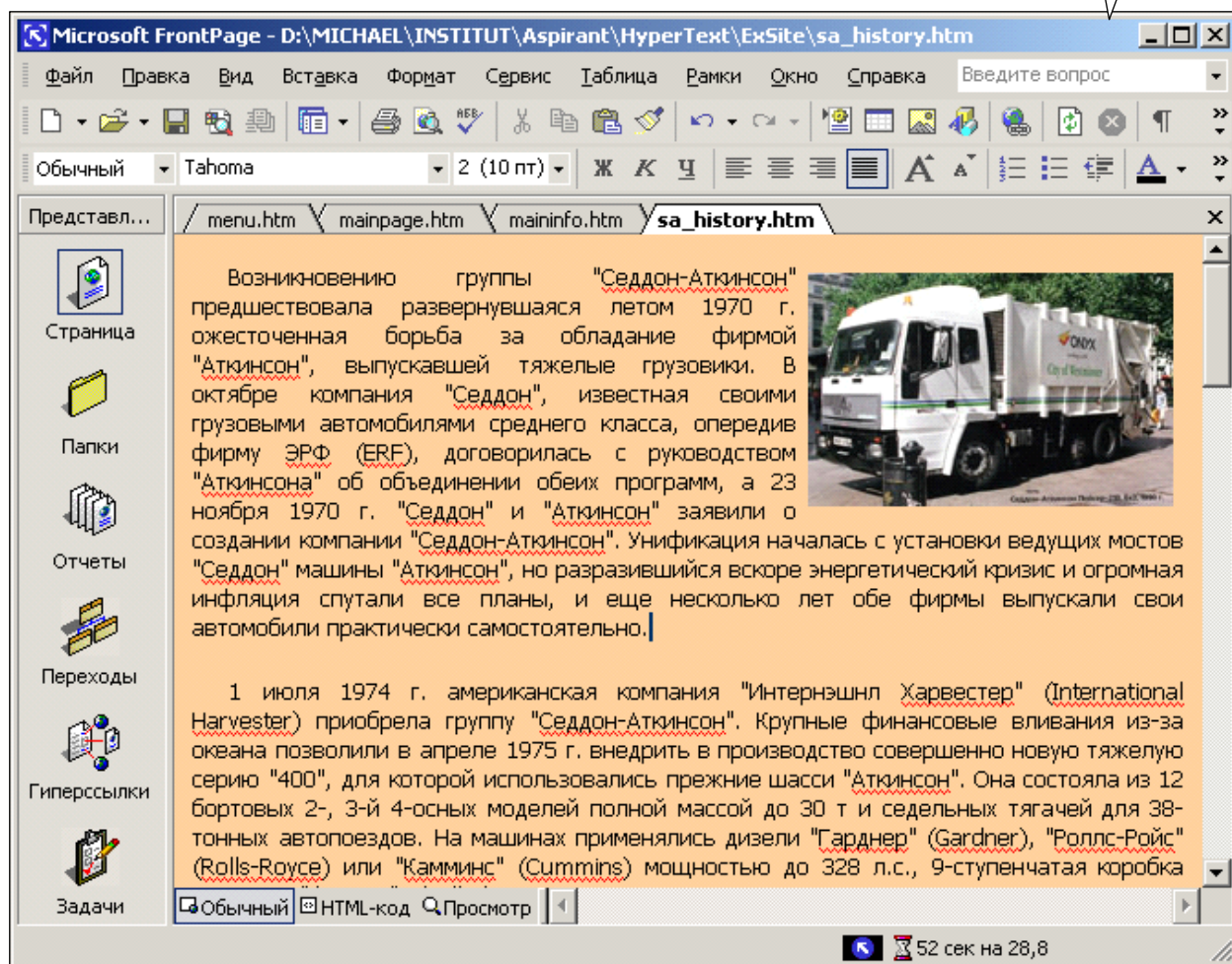


Рис. 1.7.
Новая
страница

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

6.1. Перечислите наиболее распространенные виды логических структур веб-сайтов (по материалам п. 2.1).

6.2. Почему в именах файлов нежелательно использование символов русского алфавита (по материалам п. 2.1)?

6.3. С какой целью (для использования в какой части сайта) мог быть создан файл *bg_about_me.jpg* (по материалам п. 2.1)?

6.4. Сформулируйте принцип организации, преимущества, недостатки и область применения решетчатой структуры сайта (по материалам п. 2.1).

6.5. Укажите достоинства и недостатки пакета Microsoft FrontPage как универсального инструмента для обмена и управления узлами в Web в сравнении с другими средствами редактирования HTML-страниц (см., например, [4]).

6.6. Параметры (атрибуты) какого элемента определяются заново при изменении в редакторе цвета фона веб-страницы (см., например, [4])?

6.7. Какими атрибутами или какими элементами HTML устанавливаются наклон, толщина, подчеркивание и другие особенности шрифта в тексте и его фрагментах (см., например, [4])?

ЛАБОРАТОРНАЯ РАБОТА 2

ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ ВЕБ-САЙТА

1. ЦЕЛЬ РАБОТЫ

Целью работы является получение навыков проектирования дизайна и тестирования (исследования) веб-сайтов с помощью редактора HTML-страниц. На первый план выдвигаются проблемы оформления документов, а не организации их структуры, как это было при выполнении лабораторной работы № 1.

2. СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1. ТЕРМИНОЛОГИЯ

Определим основные понятия, которые используются в разделе.

Интернет – это всемирная компьютерная сеть, объединяющая десятки тысяч небольших сетей и миллионы коммерческих, образовательных и персональных компьютеров. Упомянутые небольшие корпоративные сети (*intranet*) обеспечивают внутренние информационные потребности организаций и построены на базе технологий Интернета, таких как протоколы HTTP и TCP/IP.

Web (World Wide Web, WWW) представляет собой систему для поиска

данных в Интернете, основанную на механизме гиперссылок, и для работы с ними. При использовании программы просмотра система *Web* выглядит для пользователя как набор *Web*-страниц, взаимосвязанных по смыслу содержащимися в них текстами, изображениями, звуковыми и видео фрагментами, объединенными структурно возможностью перехода с помощью гиперссылок.

Страница Web – это документ в формате HTML, как правило, содержащий гиперссылки на другие страницы.

Гиперссылка есть механизм перехода от одного ресурса сети Интернет к другому. Ресурсом может быть любая другая *Web*-страница, любой документ HTML, файл, часть файла. Гиперссылка не является, вопреки мнению многих практиков создания сайтов, ни элементом языка HTML, ни атрибутом какого-либо элемента. Это механизм, средство простого перемещения по ресурсам сети. Существуют элементы, наделенные способностью образовывать ссылки: это якорь, связь, карта, область. Для этого они поддерживают необходимые атрибуты, управляющие переходом: адреса, источники, идентификаторы, имена. На экране гиперссылка выглядит как выделенный (цветом или подчеркиванием) фрагмент текста, либо как графический объект, щелчок мыши по которым позволяет реализовать переход к новому ресурсу.

Начальная страница – это страница, которая первой выводится на экран при запуске. В этом качестве может использоваться любая страница *Web* или любой документ, хранящийся в компьютере.

Страница поиска предназначена для перехода по узлам *Web* или intranet. Они позволяют осуществить поиск по заголовкам, по ключевым словам или по условиям запроса. Страница поиска может быть открыта с помощью панели *Web* из любого приложения Microsoft Office.

Браузер представляет собой программу интерпретации файлов, записанных на языке HTML, в виде документов, воспроизводимых пользователю. Браузер принимает поток байтов из сети и преобразует их в изображение на вашем экране.

Сайтом или узлом Web называют совокупность HTML-страниц, таблиц стилей, скриптов, графических файлов, музыкальных файлов, и т.д., связанных между собою гиперссылками и объединенных общей темой и целью. Если вы сохраните часть файлов своего сайта на логическом диске C:, а часть – на логическом диске D:, связав их гиперссылками, это будет один сайт. Даже если эти диски будут не логическими, а физическими. Даже если вы разместили свои файлы на одном сервере, а гостевую книгу создали на другом. Место размещения компонентов сайта не является существенным признаком.

2.2. ТЕХНОЛОГИЯ СОЗДАНИЯ САЙТА

Руководством к действию авторов, намеревающихся обеспечить высокую посещаемость своего сайта, обычно служит известная аксиома:

«Посетители не должны разбираться в том, какова иерархия и организация сайта. Посетителям важно видеть, куда они могут пойти».

При создании дизайна сайта следует придерживаться убежденности в том, что его будут посещать огромное число пользователей с разными вкусами и потребностями. Даже при выполнении лабораторной работы, где эта армия представлена лишь вашим преподавателем, полезно надеяться, что сфера распространения результата вашего труда многократно расширится. Подобный подход позволит создать универсальный интерфейс, элементы которого вы будете использовать в будущем, с которым будет комфортно работать большинству посетителей. Абсолютно удобный во всем и для всех интерфейс можно создать лишь теоретически, но к нему можно существенно приблизиться на практике. В [1 – 3] приведены, будем говорить, азбучные истины этого процесса, перечисленные ниже. Заметим, что в настоящей работе нам придется отступить от совета №3 – при ее выполнении нам придется воспользоваться фреймами, чтобы усвоить навыки работы с этими конструкциями HTML.

1. Старайтесь не перегружать страницу графикой, особенно если она заимствована из документов с бесплатных серверов. Это касается также общеупотребительных значков типа E-mail, Welcome, Next и т.д. Старайтесь, по возможности, качественно создавать собственную графику, от этого у посетителей вашей страницы останется хорошее впечатление. Они захотят вернуться.

2. Избегайте избытка и диссонанса красок в цветовой гамме своей страницы. Она должна быть выдержана в едином стиле. Пестрота и зрительная несовместимость тонов утомляет и порождает подсознательное раздражение.

3. Если есть возможность (в рамках курса «Пакеты прикладных программ» такая возможность исключается), постарайтесь не использовать на странице фреймы. Это упростит индексирование сайта в поисковых системах и сделает документы доступными на всех компьютерах, подключенных к сети.

4. Если вы используете таблицы, тогда для упрощения навигации старайтесь задавать параметры ширины в пикселях, а не в процентах. Иначе у машин с разным разрешением экрана страница будет выглядеть по-разному, и разметка ссылок окажется менее стабильной.

5. Никогда не ориентируйте свою страницу на определенный браузер. От этого вы можете потерять до половины своих посетителей ежедневно. Постарайтесь найти компромисс.

6. Не делайте навигацию слишком запутанной. Доступ к любой странице сайта должен быть не более чем с помощью трех щелчков мыши.

7. Если вы решили вступить в какую-либо рекламную сеть, не стоит перегружать страницу баннерами, большой выгоды от этого все равно не будет. Одного или двух баннеров на страницу вполне достаточно.

8. Не используйте (как и при выборе цветовой палитры) большого разнообразия шрифтов, от этого у посетителя будет рябить в глазах, и не

исключено, что заболит голова.

9. Делайте удобную и продуманную навигацию, стараясь максимально увеличить комфорт "перемещения" по ресурсам вашего сайта.

10. Всегда задавайте цвет основного фона вашего сайта. Даже если вы используете в качестве фона картинку, все равно укажите подходящий цвет (гармонирующий с картинкой).

11. Не перегружайте страницу графикой. Посетителей не привлекают "шедевры" вашей живописи, дожидаться появления которых на экране приходится по несколько минут. Если все-таки требуется поместить большое изображение, лучше «порежьте» картинку на более мелкие части и вставьте их в таблицу. Этим вы существенно сократите время загрузки графики.

12. Возьмите за правило указывать в открывающем теге элемента линейные размеры изображения WIDTH и HEIGHT. Иначе при загрузке картинок текст будет прыгать в разные стороны, отыскивая себе подходящее место. Посетителю это не понравится.

13. Задавайте в описании изображения альтернативный (ALT) текст. Не заставляйте посетителей ждать окончания загрузки, чтобы узнать, какую картинку вы собираетесь ему представить.

14. Не используйте в тексте слишком много восклицательных знаков и других меток привлечения внимания. Этим вы добьетесь обратного: посетитель просто не заметит действительно важной информации.

15. Если вы полагаете, что обилие мерцающих ссылок привлекают внимание, то вы ошибаетесь. Не следует думать, что Интернет излишне статичен, что на сайтах солидных учреждений запечатлен дух академичности или пресыщения. Как и в рассмотренных выше случаях крикливости цветов или сочетания латиницы с готикой, кириллицей, иероглифами и арабской вязью, большое количество зазывающих ссылок совершенно не радует глаз. После просмотра такого сайта можно получить головную боль до вечера.

16. Проверяйте, хорошо ли выглядит страница в разных разрешениях. Вы должны позаботиться обо всех своих посетителях.

17. Старайтесь не прибегать к использованию полосы горизонтальной прокрутки.

18. Следите за правописанием на странице. Многих пользователей раздражает безграмотность авторов. Подумайте сами, какими сокровищами знаний может поделиться человек, не удосужившийся усвоить правила орфографии? Если вы собираетесь создать действительно хороший сайт, то не стыдитесь держать под рукой орфографический словарь.

19. Не помещайте сайт в одну большую таблицу. Во-первых, простым изменением размеров окна посетитель может нарушить весь ваш дизайн. Во-вторых, содержимое таблицы в большинстве случаев будет показано только тогда, когда она полностью загрузится – для немедленного отображения начала таблицы нужно предпринимать специальные меры, которые не приносят эффекта на старых браузерах.

20. Всегда проверяйте корректность внутренних ссылок. Очень часто web-мастера оставляют такие ссылки: file:///C:/...

21. Проверяйте действенность специальных эффектов, воплощенных вами в законченный труд, на близких людях – они не обидят вас несправедливой критикой, но доброжелательно укажут многие недостатки, ухудшающие навигацию по сайту и восприятие подборки документов.

Для навигации по сайту, содержащему множество вложенных подразделов, идеально подходит отображение наименований всех вышерасположенных подразделов так, как это делается, во многих приложениях, например, на сайте Sun Microsystems <http://www.sun.com> (см.



Рис. 2.1. Пример расположения ссылок на разделы сайта

рисунок 2.1).

Другим примером создания грамотной и удобной навигации по сайту является дублирование ссылок, размещенных в начале документа, в его завершающей части. Пользователь, бегло осмотрев начальную страницу, может сделать свой выбор, не возвращаясь к первым строкам. Так организован сайт компании Арманд <http://www.armand.ru> (см. рисунок 2.2). В нем помимо, постоянно присутствующего на экране меню в верхней левой части окна, имеется дополнительное меню в нижней части экрана. Подобное расположение меню дает возможность пользователю переходить из раздела в раздел с минимальными затратами времени.

При создании сайта следует уделить особое внимание удобству чтения его содержимого. Как говорилось выше, не следует задавать линейные

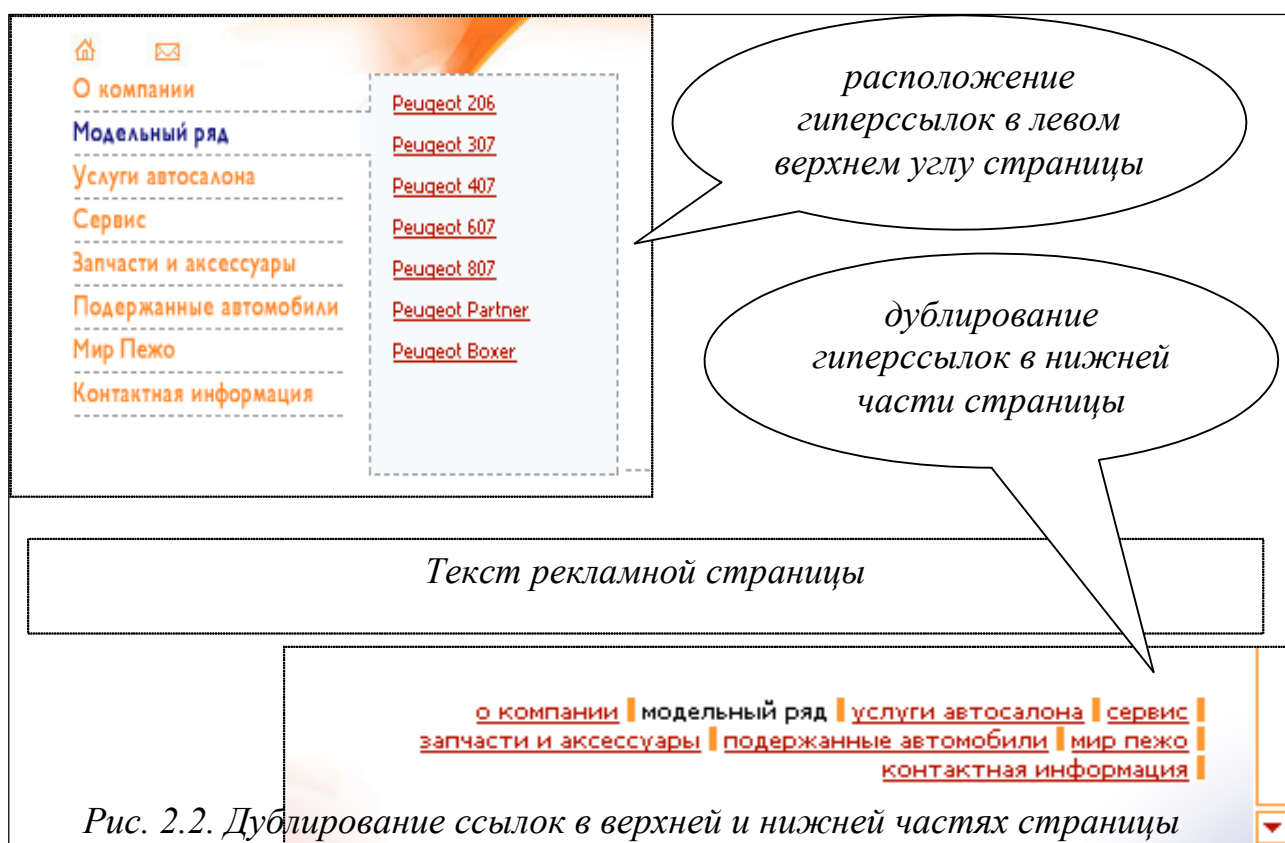


Рис. 2.2. Дублирование ссылок в верхней и нижней частях страницы

размеры вашего сайта задана процентными отношениями от габаритов окна браузера. Вы рискуете вынудить пользователя работать с полосой горизонтальной прокрутки и переключать слева направо и обратно каждую строчку текста, если его монитор имеет меньшую диагональ, чем ваш. И напротив, тот же пользователь, установив новый монитор с большей диагональю, будет вынужден «мотать головой» из стороны в сторону, читая длиннющие строчки. Такое занятие утомительно. Если читатель испытывает дискомфорт от чтения, все остальное – дизайн, скорость загрузки, информационное наполнение – не создает серьезных преимуществ. Восприятие текста, его разборчивость становятся определяющими факторами.

Установлено, что большинство пользователей предпочитают видеть перед собой текст «с засечками» (например, шрифт Times New Roman), в то время как текст, служащий для комментариев или подписей, лучше воспринимается, если он представлен «рубленным» шрифтом (например, Arial). Текст и фон, на котором он воспроизводится, должны быть максимально контрастными. В идеальном случае создавайте отдельный вариант сайта для каждого разрешения экрана. Если же такой возможности нет, старайтесь угодить вкусам большинства пользователей. Как вариант, можно при загрузке анализировать информацию о разрешении экрана пользователя и до определенного размера (например, 1024×768) устанавливать ширину сайта в процентах от ширины окна, а при превышении этого размера фиксировать ее на тех же 1024 точках.

3. ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

3.1. Подготовить сменный носитель (дискету, flash-накопитель) с результатами работы №1. На созданном в ней сайте будет производиться доработка дизайна, составляющая содержание данного занятия. Результаты будут сохранены с целью их дальнейшего использования в работах №№ 3 – 4.

3.2. По рекомендованной литературе [5 – 7] продолжить ознакомление с принципами работы в среде Microsoft FrontPage.

3.3. Обосновать выбор изобразительных средств рекламной страницы – формы, объекты, шрифты, цвет фона, расположение фотографий, схем и т.д.

3.4. Обосновать выбор состава и взаимного положения основных конструкций навигации (типы элементов, образующих ссылки и т.п.).

3.5. По настоящим методическим указаниям составить план действий за компьютером с распределением обязанностей между членами бригады.

3.6. Подготовить текст теоретических разделов отчета о работе.

3.7. Ответить на контрольные вопросы данных методических указаний.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. СОЗДАНИЕ ОСНОВНОЙ СТРАНИЦЫ САЙТА (С ФРЕЙМАМИ). С помощью меню «Шаблоны страниц» или вручную создайте страницу с фреймами (пример показан на рисунке 2.3). Сохраните ее под именем index.htm или index.html. Помните, что эти имена уже использовались предшественниками, и могут вступить с ними в конфликт. Для каждого фрейма явно задайте страницу, которая будет в нем размещена. Обязательно укажите, в каком из фреймов будут открываться вызываемые по ссылкам страницы.

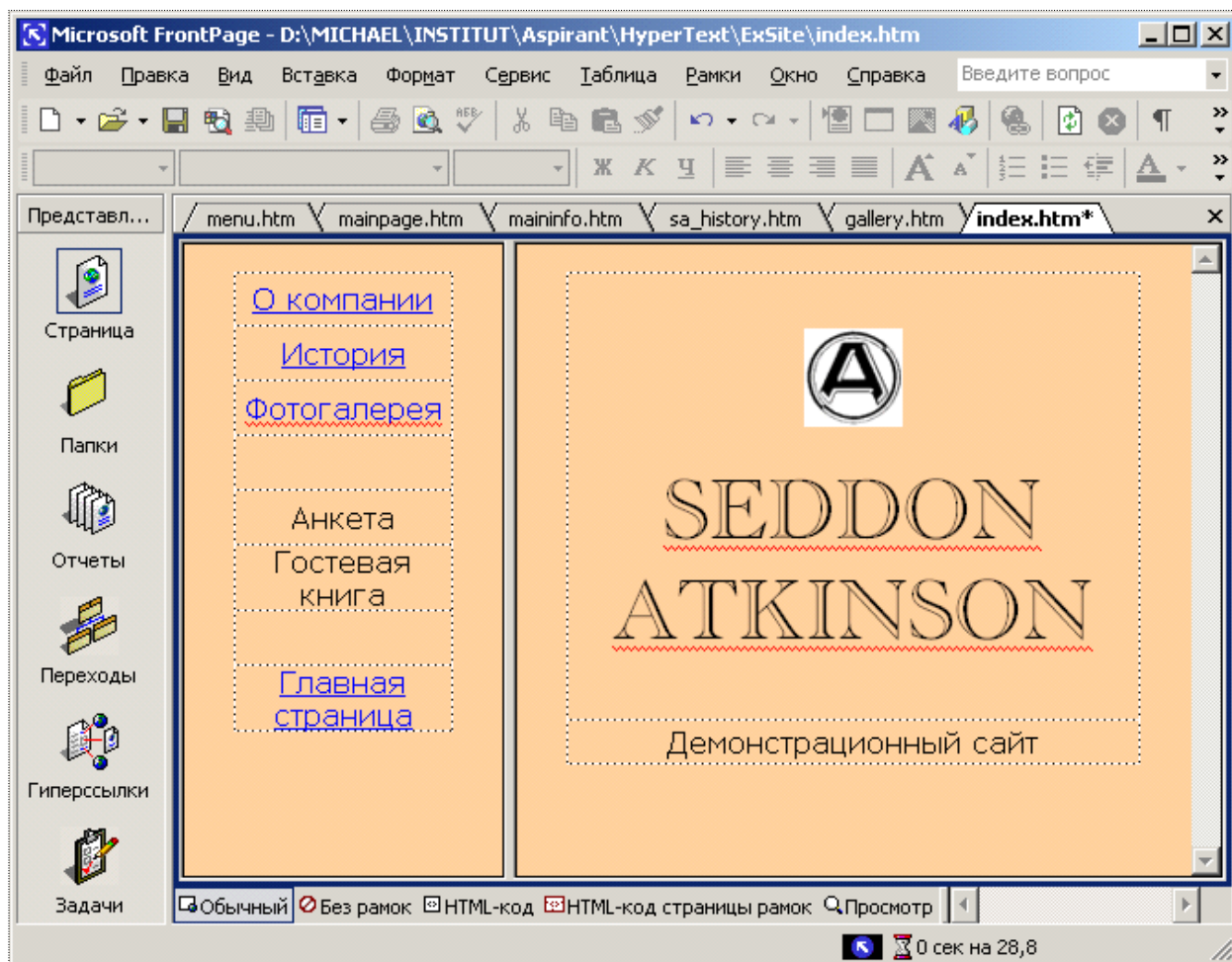


Рис. 2.3. Создание страницы с фреймами

Результат представлен на рис. 2.3. Перейдем к оформлению страниц, входящих в подразделы сайта. Нажатие кнопки «Отчеты» вызовет список отчетов о страницах, входящих в веб-сайт (см. рис. 2.4). Двойной щелчок левой кнопки «мыши» по элементу списка выводит на экран более подробную информацию.

На рис. 2.4, в разделе «Описание», отображены такие фразы, как:

- «за последние 30 дней»;
- «дольше, чем 30 секунд»;

- «за последние 72 дней».

Имя	Подсчет	Размер	Описание
Все файлы	0	OKB	Все файлы на текущем сайте
Картинки	0	OKB	Файлы картинок в текущем сайте (GIF, JPG, BMP, и т.д.)
Несвязанные файлы	0	OKB	Файлы в текущем сайте, которые не могут быть достигнуты стартуя с вашей Домашней страницы
Связанные файлы	0	OKB	Файлы в текущем сайте которые можно достучнуть стартуя от вашей Домашней страницы
Медленные страницы	0	OKB	Страницы в текущем сайте которые грузятся дольше чем 30 секунд при скорости 28.8
Старые файлы	0	OKB	Файлы в текущем сайте не измененные за последние 72 дней
Последние добавленные страницы	0	OKB	Файл в текущем сайте добавленные за последние 30 дней
Гиперссылки	0		Все гиперссылки в текущем сайте
Непроверенные гиперссылки	0		Гиперссылки указывающие на непроверенные конечные файлы
Сломанные гиперссылки	0		Гиперссылки указывающие на недоступные файлы
Внешние гиперссылки	0		Гиперссылки указывающие на файлы вне текущего сайта
Внутренние гиперссылки	0		Гиперссылки указывающие на другие файлы внутри текущего сайта
Ошибки компонента	0		Файлы в текущем сайте с компонентами выдают ошибку
Невыполненные задания	0		Задания в текущем сайте которые не отмечены как выполненные
Неиспользованные темы	0		Темы в текущем сайте которые не применены ни к какому файлу

Рис. 2.4. Отчет о создаваемом веб-сайте

Эти параметры настраиваются через меню «Инструменты» / «Опции» / «Просмотр отчетов». Нажатие кнопки «Навигация» вызывает схему вашего веб-сайта со всеми его страницами и гиперссылками на них. По нажатию кнопки «Гиперссылки» появятся: список страниц сайта и указание всех ссылок на выбранную из этого списка страницу со всех страниц сайта, а также все ссылки с этой страницы на все другие страницы сайта.

Главным в системе Front-Page является редактор веб-страниц. Собственно редактор состоит из трех закладок: «Нормальный», «HTML» и «Просмотр». По первой закладке вы попадаете в окно редактора, в котором, как в редакторе Word, можно создавать текст документа (рис. 2.5). Рисунок состоит из трех частей. Верхняя представляет окно «Нормальный», средняя – окно «Просмотр», нижняя – окно редактора. Это – три режима работы со страницей.

Режим «Нормальный» предназначен для редактирования страницы практически точно в таком же виде, как она будет выглядеть в процессе просмотра Web-страницы пользователем сети Интернет. Другое название режима – «видишь то, что получишь», т.е. страница отображается практически точно в таком же виде, как она будет выглядеть в программе просмотра, такой как Internet Explorer. В этом режиме на экране представлены заголовок страницы, панель навигации в виде кнопок под заголовком, панель навигации в виде гиперссылок в левой части страницы – и текст в правой части. Параметры панели навигации в виде кнопок можно изменять. Эти изменения распространятся на все страницы сайта, поскольку созданные автоматически панели находятся в общих для всех страниц областях, расположенных по краям страницы. Можно удалить лишнюю панель навигации, можно добавить новую панель.

По второй закладке вы попадаете в окно редактора HTML. Здесь отображен код, который вы набирали клавишами в первом окне. Существует и обратная связь – все то, что вы написали во втором окне, найдет свое отражение в первом. Режим HTML позволяет редактировать страницу в текстовом представлении. Используйте этот режим, когда хорошо овладеете язы-

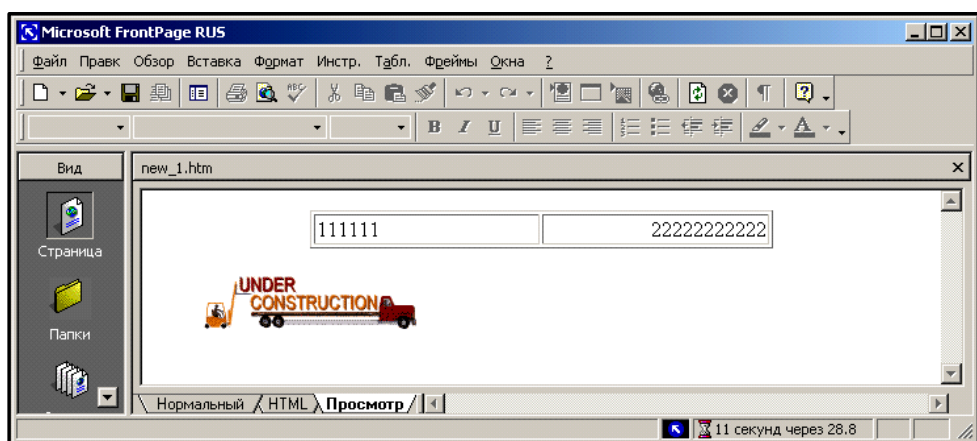
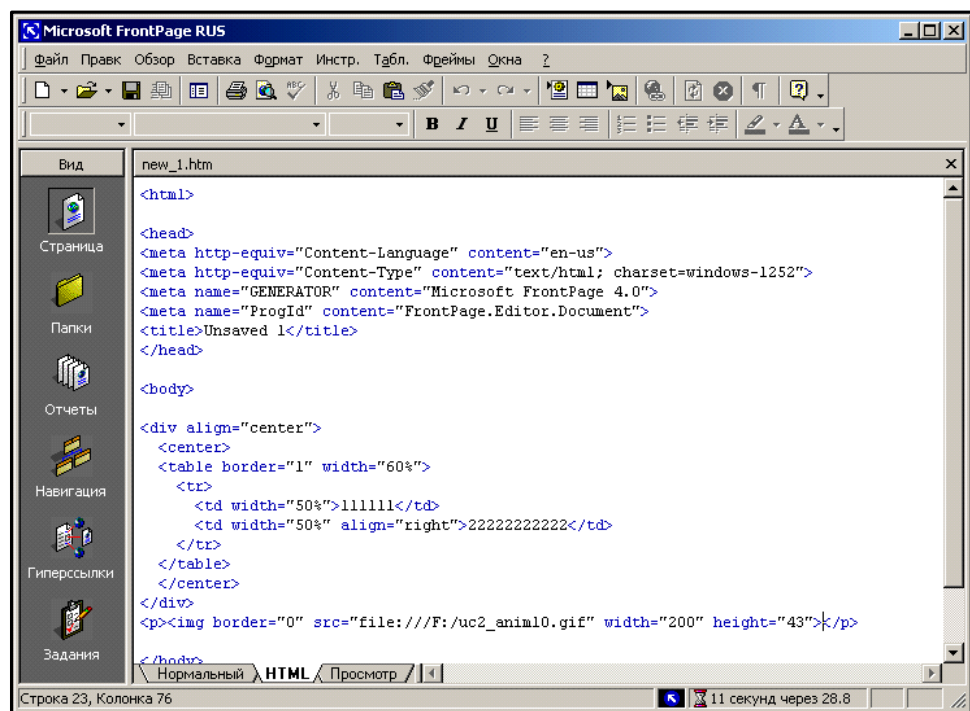
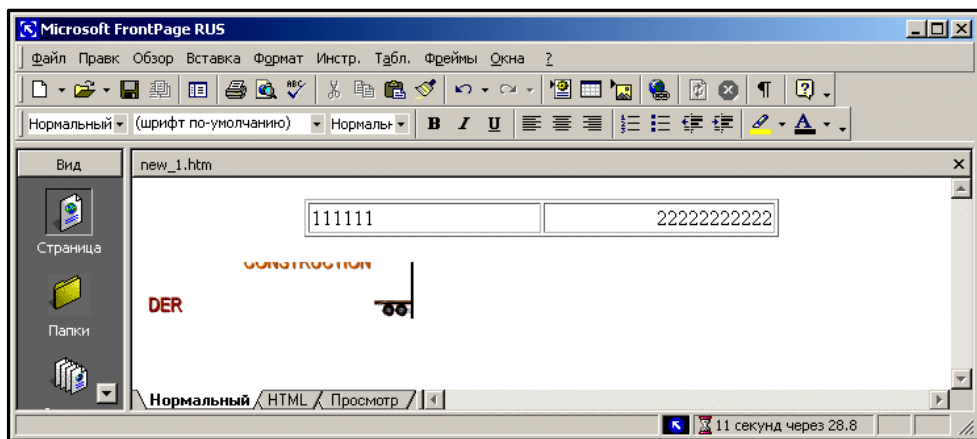


Рис. 2.5. Закладки окна редактора веб-страниц

ком разметки гипертекстов.

По третьей закладке попадаем в окно просмотра, которая покажет, как ваша веб-страница будет выглядеть в браузере. Режим «Просмотр» позволяет

не только просмотреть страницу в том виде, в котором ее отобразит браузер, но и проверить функции этой страницы, такие как: переходы по ссылкам, корректное отображение рисунков, использование объектов и форм и т.д.

Переход из режима в режим осуществляется, как и в других пакетах, щелчком мыши по закладке нужного режима. Начнем с первой закладки.

Щелчком правой кнопки мыши на каждом редактируемом объекте можно получить всплывающее контекстное меню, как это показано на примере элемента «таблица» (см. рис. 2.6). При этом всегда можно отредактировать свойства страницы: язык, кодировку символов, заголовков, фон, шрифт, поля и т.д.

Первая закладка редактора веб-страниц по содержанию фактически совпадает с хорошо известным окном редактора Microsoft Word. Обычно работа с ней не вызывает затруднений, и в данном изложении касаться ее мы не будем.

Исходное меню в FrontPage – это «Обзор». Помимо дублирования левой панели инструментов, оно содержит пункт «Показать теги», щелчок левой кнопкой мыши по которому включает или выключает режим показа тегов при редактировании документа на первой закладке редактора (см. рис. 2.7).

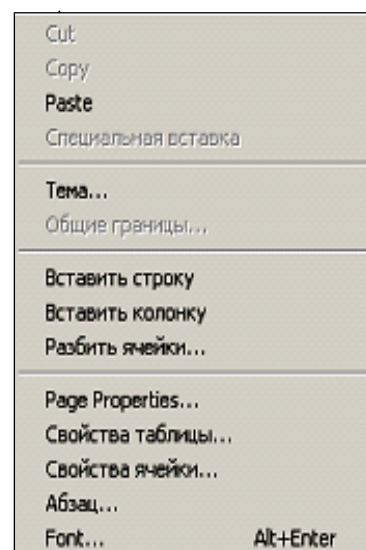


Рис 2.6.
Контекстное меню

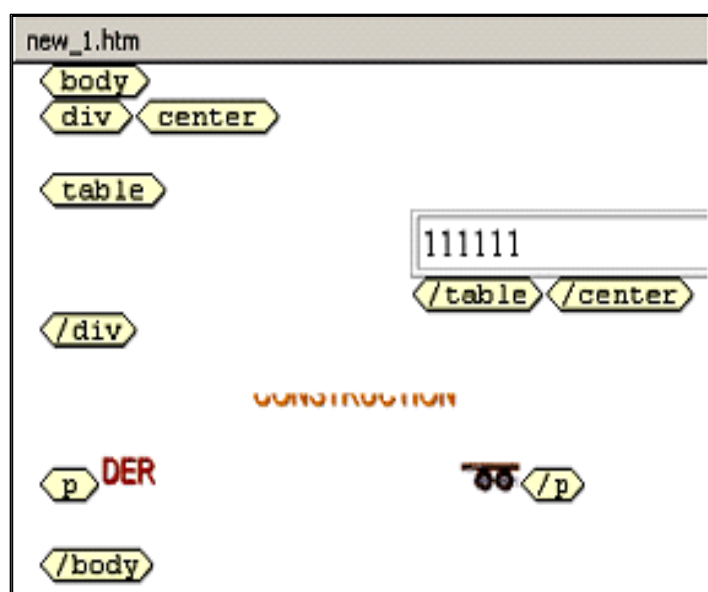


Рис. 2.7. Режим отображения тегов

Меню «Вставка» используется, чтобы внедрять в документ различные объекты: изображения, компоненты, формы, документы в формате других изделий Microsoft Office и т.д. Как уже отмечалось, нужно быть очень внимательными при добавлении вставок в формы. Для того чтобы они работали, на веб-сервере должны быть дополнительно установлены расширения FrontPage Web

Server Extensions. Взаимный обмен данными между клиент-

ским веб-сервером и компьютером основан на методе, который реализован с помощью веб-ботов. Они так и называются: «обменные» компоненты FrontPage для обеспечения передачи данных. Эта технология поддерживается только веб-серверами, на которых установлен FrontPage Web Server Extensions. Веб-бот может вызвать функцию записи информации, введенной в файл, в форму, функцию поиска информации по сайту, функцию вывода даты последнего обновления веб-страницы и т.д. Введены эти компоненты

для того, чтобы начинающим проектировщикам не составило труда создать, например, страницу поиска информации по своему сайту, или гостевую книгу. В тексте HTML-документа веб-бот записывается так:

```
<!--webbot bot="SaveResults" U-  
File="fpweb:///private/form_results.txt"S-  
Format="TEXT/CSV" S-Label-Fields="TRUE" -->
```

В меню «Формат» особого внимания заслуживают два пункта: «Динамик HTML эффекты» и «Тема». «Динамик HTML эффекты» вызывает окно добавления реакций на различные события, происходящие с произвольным элементом веб-страницы. В результате добавления эффекта в теле документа появляется заготовка функции на языке JavaScript, которую вы заполняете в соответствии со своим замыслом.

Пункт «Тема» предназначен для оформления веб-страницы аналогично одному из шаблонов (тем), предлагаемых создателями FrontPage.

4.2. СОЗДАНИЕ ФОТОГАЛЕРЕИ. Если в вашем сайте предусмотрена фотогалерея, то предпочтительнее создавать ее путем размещения на странице уменьшенных копий изображений, разворачивая полноразмерное изображение при нажатии мышью на уменьшенное (см. рисунок 2.8).

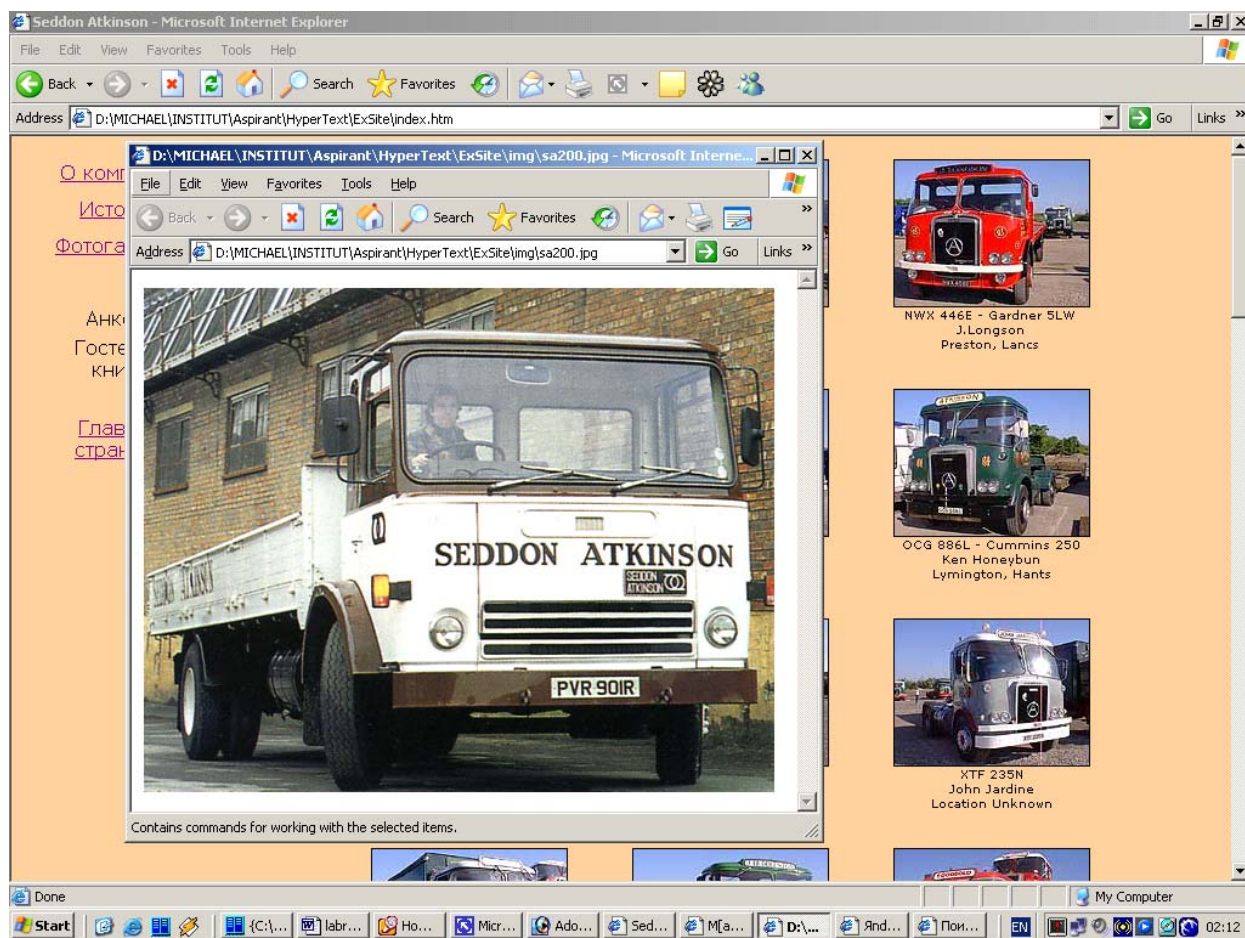


Рис. 2.8. Пример фотогалереи

4.3. ДОРАБОТКА ДИЗАЙНА. Подправьте дизайн ваших страниц в соответствии с рекомендациями 1 – 21, сформулированными в разделе 2. Для сайта, создающегося в качестве примера, желательно исключить отображение границы между фреймами, а также изменить цвет шрифта, которым отображается текст ссылки. Результат предложенных корректировок представлен на рисунке 2.9. По выполнении пункта 4.3 разработка информационной части сайта считается завершенной.

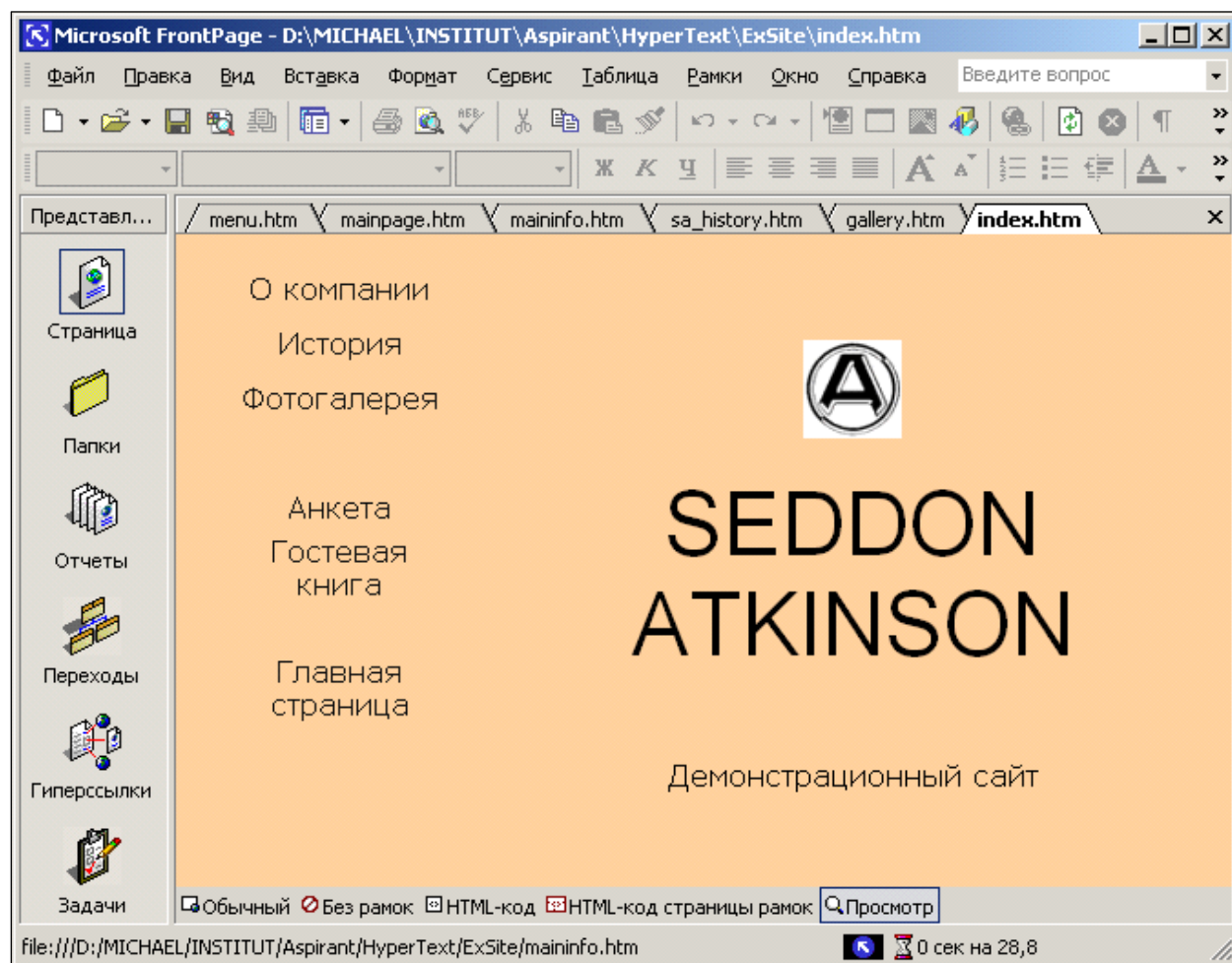


Рис. 2.9. Пример доработки стиля представления страницы

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет о работе должен содержать следующие материалы в соответствии с полученным индивидуальным заданием:

5.1. Схема перемещения по ресурсам сайта (по материалам раздела 2 данной работы).

5.2. Критический анализ списка «советов веб-дизайнеру» (раздел 2.2 настоящего описания).

5.3. Обоснование выбора для выполнения работы № 2 элементов HTML, образующих ссылки (по материалам раздела 2 данной работы).

5.4. Обоснование выбора палитры цветов и других изобразительных

возможностей HTML для выполнения работы № 2 (по материалам раздела 2 данного описания).

5.5. Предложения по реализации рекомендованного в п. 4.2 способа построения галереи фотографий – перехода от мелкого масштаба изображения к полноразмерному воспроизведению на экране.

5.6. Содержание выполненных индивидуальных заданий.

5.7. Ответы на контрольные вопросы и выводы по выполненной работе.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

6.1. В чем состоят преимущества и недостатки формирования страниц с помощью больших таблиц HTML (по разделу 2 данного описания).

6.2. Чем определяется, по вашему мнению, качество стиля представления документов: научно обоснованными критериями или интуитивно воспринимаемым чувством целесообразности формы, вкусом, уровнем общей культуры автора (по материалам разделов 2 – 4 данного описания)?

6.3. Какими средствами обеспечивается привлекательный дизайн при воспроизведении документов на мониторах с различными размерами и различным разрешением (по материалам разделов 2 – 4 данного описания)?

6.4. Для чего требуется дорабатывать дизайн сайта (по п. 4.3)?

6.5. По какой причине не рекомендуется чрезмерно разнообразить цветное и стилистическое оформление документа (по материалам раздела 2 данного описания)?

6.6. Почему при указании параметров изображения (элемент IMG) требуется добавлять его текстовое описание, и как это осуществляется практически (по материалам п. 2.2)?

6.7. Какими «советами веб-дизайнеру» пренебрег автор сайта-примера (по материалам п. 2.2 и раздела 4)?

ЛАБОРАТОРНАЯ РАБОТА 3

СОЗДАНИЕ СЦЕНАРИЕВ НА ЯЗЫКЕ JAVASCRIPT

1. ЦЕЛЬ РАБОТЫ

Целью работы является приобретение навыков создания и тестирования сценариев на языке JavaScript. Рассматриваются проблемы добавления динамики в документы, а также обеспечения правильной реакции на действия пользователя, позволяющей в удобном для восприятия виде предоставить необходимый минимум информации о возможностях взаимодействия.

2. СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1. ЯЗЫК JAVASCRIPT

JavaScript появился в 1995 году, когда язык Java был уже достаточно широко известен. К тому времени, однако, существовал прототип языка JavaScript — язык описания скриптов (сценариев) LiveScript, встроенный в браузер Netscape Navigator 2.0. Впоследствии компания Netscape отказалась от названия LiveScript и в сотрудничестве с компанией Sun Microsystems, создавшей язык программирования Java, начала разработку нового языка, получившего название JavaScript [6,7].

Несмотря на некоторое сходство в синтаксисе и методах создания объектов, языки Java и JavaScript мало похожи. Определение языка JavaScript как "облегченной версии" языка Java является неточным. Многие утверждают, что язык JavaScript — это язык *описания сценариев*, а не язык программирования. Однако описание сценариев и программирование тесно связаны между собой, несмотря на то, что скрипты и программы применяются для различных целей. Создание скриптов часто рассматривают как "упрощенное программирование", т.е. средство для решения простых задач, хотя фактически именно скрипты являются основой многих программных продуктов, работающих в Internet. Подобно программам на языке Java, JavaScript-программы компилируются во внутреннее представление, известное как *байт-код*, который затем выполняется интерпретатором. Главной целью языка JavaScript является поддержка активного взаимодействия документов HTML с пользователем. Этот язык не претендует на то, чтобы быть полномасштабным средством программирования, таким как Java или C++. Скорее, он является расширением языка HTML, облегчающим работу пользователя с конкретным браузером. Язык JavaScript расширяет возможности стандартных элементов HTML, позволяя конструкциям веб-страницы взаимодействовать с объектами и свойствами языка JavaScript. С введением обработчиков событий HTML-документы стали более "живыми", поскольку разработчики получили возможность определять их поведение в зависимости от действий пользователя. Важен тот факт, что JavaScript-программы действительно являются выполняемым содержимым документов; они физически находятся внутри HTML-документов, в отличие от Java-апплетов, которые существуют вне документов, их активизирующих. Пример реакции системы на действия человека приведен на рис. 3.1. Курсор мыши в своем движении по экрану остановился на символе сноски. В результате обработки произошедшего события поверх основного текста документа всплыл текст сноски. Читателю не нужно использовать прокрутку документа, чтобы добраться до сноски, она сама появляется перед ним и сама исчезает, если курсор мыши сдвинуть в сторону от символа сноски.

Все события, происходящие в браузере Navigator, например, нажатие кнопки или переход к другой странице, обнаруживает и обрабатывает операционная система, передавая результирующие параметры JavaScript-программам. Важность этих событий состоит в том, что они позволяют программным элементам документов и браузеру более тесно взаимодействовать друг с другом. На-

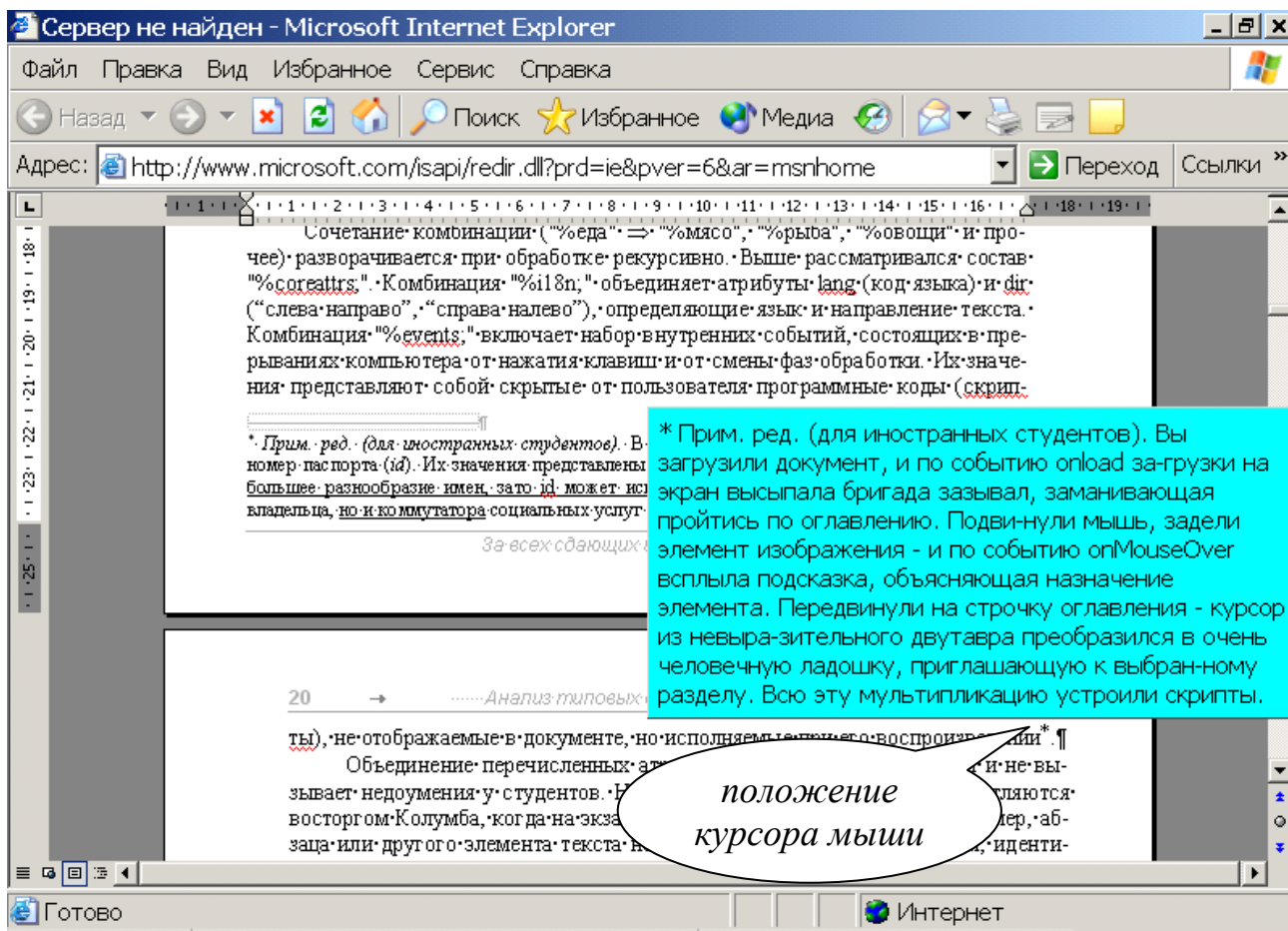


Рис. 3.1. Пример использования скриптов в документах

пример, JavaScript-программа включается, когда пользователь покидает страницу, и выполняет при этом необходимые, запланированные автором действия. JavaScript-программы могут обрабатывать множество других событий, таких как выбор нового элемента списка или инициализация формы. Кроме того, язык JavaScript хорошо приспособлен для проверки правильности описания и заполнения форм, обработки строк и динамического создания HTML-элементов. Он позволяет создавать их динамически, в процессе взаимодействия с пользователем, что называется "на лету". Почти во всех сложных приложениях HTML для управления внешним видом документа используются JavaScript-программы, динамически создающие HTML-элементы и даже целые документы. В частности, чтобы Web-страница приобрела новый внешний вид, ее автору не нужно вручную изменять исходный текст, это делают скрипты.

В результате, с помощью JavaScript-программ вы можете:

- формировать HTML-документы "на лету";
- производить проверку правильности данных HTML-формы перед передачей их на сервер;
- предоставить пользователю возможность вводить локальные данные для управления работой JavaScript-программы, а также выборочно выполнять различные операции;
- создавать окна сообщений и диалоговые окна для вывода предупреждающих сообщений и ввода данных;

- создавать документы с расширенными возможностями навигации, используя фреймы и автономные окна;
- обнаруживать Java-апплеты и подключаемые модули (plug-in) браузера и взаимодействовать с ними.

2.2. ОСНОВЫ ГРАММАТИКИ JAVASCRIPT

Большинство программ на языке JavaScript распространяются по сети упакованными в документах, запрашиваемых пользователями. До тех пор пока ресурс не загружен в браузер, невозможно определить, содержит он JavaScript-программу или нет. Чтобы браузер мог обнаружить такую программу, в язык HTML введен элемент `<SCRIPT>`. Имеется также дополняющий элемент `<NOSCRIPT>`, позволяющий авторам HTML-документов выяснять, когда язык JavaScript использовать нельзя. Скрипты могут находиться в любом месте документа. Обратное неверно: теги элементов языка HTML нельзя помещать внутри JavaScript-программы. Не забывайте заключать JavaScript-программу в теги `<SCRIPT>...</SCRIPT>`, кроме тех случаев, когда она используется как обработчик события. Событием может явиться нажатие кнопки, передача заполненной формы, загрузка новой страницы и т.д. При встрече с открывающим тегом `<SCRIPT>`, браузер построчно анализирует содержимое документа до тех пор, пока не будет достигнут закрывающий тег `</SCRIPT>`. После этого производится проверка скрипта на наличие ошибок и компиляция JavaScript-программы в формат, пригодный для выполнения на компьютере пользователя. Если при проверке или компиляции программы обнаруживаются ошибки, Navigator выводит на экран окна с предупреждениями. Чтобы продолжить работу, необходимо нажать кнопку.

Главная часть JavaScript-программы может быть помещена в контейнер `<HEAD>...</HEAD>`, поскольку он считывается в программу Navigator при загрузке HTML-документа одним из первых. Теоретически скрипт можно размещать в любом месте HTML-документа, хотя лучше это делать перед контейнером `<BODY>...</BODY>`, т.е. в заголовке документа, чтобы функции языка JavaScript загружались в память сразу же после загрузки документа в браузер. Однако некоторые авторы любят размещать скрипты в конце программы. Тогда их не нужно разыскивать по тексту разметки. Окончательный выбор остается за вами. Описания функций лучше всего помещать в заголовки документа, хотя скрипты, выполняемые "на лету" и формирующие новые документы HTML при обращении к определенным частям текущего документа, могут быть размещены там, где они необходимы. Синтаксис элемента `<SCRIPT>` следующий:

```
<SCRIPT [language="JavaScript"] [src=URI]>
  [JavaScript-statements (предложения языка JavaScript)...]
</SCRIPT>
```

Как известно, необязательный атрибут *language* элемента `SCRIPT` устанавливает язык скриптов, используемый в документе. Это может быть «родной» для компании Microsoft язык VBScript, наследующий конструкции Vis-

ual Basic. Здесь значением атрибута *language* является строка "JavaScript". Как правило, то же значение присваивается атрибуту по умолчанию. Далее, с помощью атрибута *src* (*source*, т.е. ресурс, источник) определяется адрес файла, содержащего скрипт. Заметим, что регистр символов в данном случае не имеет значения, как и при указании тегов языка HTML. Если атрибут *src* определен, то тело скрипта [JavaScript-statements...] пропускается, потому что скрипт будет загружен из определенного атрибутом *src* файла. Если значение атрибута *src* не указывается явно, то предполагается, что программа содержится в контейнере <SCRIPT>.. </SCRIPT>. Операторы языка JavaScript подробно рассматриваются в [6,7].

Для того чтобы включить JavaScript-программу в HTML-документ, используйте открывающий тег <SCRIPT> без определения атрибута *src*. Например, при загрузке следующего HTML-документа на экран выводится окно сообщения:

```
<HTML>
<!--Пример сценария 1-1 -->
<HEAD>
<SCRIPT language=" JavaScript ">
alert {"Добро пожаловать !"} ;
</SCRIPT>
</HEAD>

----- тело документа -----
</HTML>
```

Метод alert() – это один из инструментов языка JavaScript. При его выполнении на экране появляется окно сообщения, в котором отображаются заданный текст, кнопка ОК и пиктограмма с восклицательным знаком. Такие предупреждения используются для привлечения внимания пользователя. Приведенная программа выполняется немедленно после загрузки включающего ее документа, хотя в языке JavaScript можно определить и функции, содержащие последовательности операторов – аналогов обработчиков прерываний в пакетах визуального программирования. Такие функции активизируются только при обращении к ним из основной программы.

2.3. СИСТЕМА СОБЫТИЙ ЯЗЫКА JAVASCRIPT

Использование языка JavaScript при обработке событий значительно расширило возможности языка HTML. В элементы HTML, определяющие гиперсвязи и компоненты формы, добавлены необходимые атрибуты. Чаще всего сценарии создаются для контроля реакции системы на любые действия пользователя, от движения курсора мыши по экрану до обработки информации, вводимой в поля форм. Возможности управления элементами форм обеспечиваются главным образом за счет функций обработки событий, которые задаются для всех элементов документа, в частности, для формы. Форма представляет собой контейнер, содержащий поля ввода, области текста, списки и кнопки. Для каждого из этих элементов определяются программы об-

работки событий, что существенно повышает степень интерактивности документа. События делятся на несколько категорий:

- события, связанные с документами в целом – загрузка и выгрузка;
- события, связанные с гиперссылками – активизация гиперссылки;
- события, связанные с формой – щелчки мыши на кнопках (button), группах кнопок выбора варианта (radiobutton), переключателях (checkbox), кнопках передачи данных и восстановления исходных значений элементов, получение и потеря фокуса ввода, а также изменение содержимого полей ввода, областей текста и списков, выделение текста в полях ввода и областях текста;
- события, связанные с рисунками – загрузка рисунка, ошибка загрузки рисунка, прерывание загрузки рисунка;
- события, связанные с клавиатурой – нажатие, давление, отпускание любой клавиши компьютера;
- события, связанные с мышью – помещение указателя мыши на гиперссылку и активизация гиперссылки, движение курсора по экрану, щелчки и двойные щелчки кнопкой мыши.

События, связанные с документами, возникают при загрузке и выгрузке документа, в то время как события гиперсвязей возникают при их активизации или при помещении на них указателя мыши. Очень популярно использование программ на языке JavaScript для обработки событий форм. События, связанные с рисунками, позволяют выполнять ответные действия как в процессе загрузки рисунка, так и при возникновении ошибок. Рисунки теперь тоже представляются как объекты с программными свойствами, которые можно описать массивами. Например, выражение `document.images[0].src` соответствует атрибуту *src* (адрес URI) в открывающем теге первого элемента `<IMG>` текущего документа.

Обработчики событий конкретного объекта задаются в HTML-теге, определяющем этот объект. Например, обработчик события, связанного с рисунком, задается в теге `<IMG>`, обработчик события гиперсвязи – в теге `<A>` и т.д. В целях «перехвата» события программируют функции-обработчики событий. Ими могут оказаться достаточно объемные коды, или только группы из одного или нескольких операторов, разделенных точкой с запятой (;). В таблице 3.1 перечислены имена большинства событий, используемых в языке разметки гипертекстов, и условия их возникновения. В среднем столбце таблицы сосредоточены названия атрибутов, обеспечивающих правильную реакцию на события.

Таблица 3.1

Имя события	Атрибут	Условие возникновения события
Blur	onBlur	Потеря фокуса ввода элементом формы
Change	onChange	Изменение содержимого поля ввода или области текста, выбор нового элемента списка
Click	onClick	Щелчок мыши на элементе HTML
Focus	onFocus	Получение фокуса ввода элементом формы

Load	onLoad	Завершение загрузки документа
MouseOver	onMouseOver	Помещение указателя мыши на элемент
MouseOut	onMouseOut	«Уход» указателя мыши из контура элемента
KeyPress	onKeyPress	Нажатие клавиши на клавиатуре компьютера
Select	onSelect	Выделение текста в поле ввода или во фрагменте текста
Submit	onSubmit	Передача данных формы
Unload	onUnLoad	Выгрузка текущего документа (начало загрузки в браузер нового документа)

3. ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

3.1. Подготовить сменный носитель (дискету, flash-накопитель) с результатами работ №№ 1 и 2. На созданном в них сайте будет производиться доработка дизайна, составляющая содержание данного занятия. Результаты будут сохранены с целью их дальнейшего использования в работе № 3.

3.2. По рекомендованной литературе [6, 7] ознакомиться с принципами написания скриптов на языке JavaScript.

3.3. Подготовить и обосновать предложения по внесению динамики в сайт, разработанный на занятиях 1 и 2. Определить основные события, на которые должна реагировать система.

3.4. Составить программные коды обработчиков прерываний по п. 3.3.

3.5. По настоящим методическим указаниям составить план действий за компьютером с распределением обязанностей между членами бригады.

3.6. Подготовить текст теоретических разделов отчета о работе.

3.7. Ответить на контрольные вопросы данных методических указаний.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

В процессе выполнения работы поощряются инициативные предложения студентов по разработке сценариев для придания динамики проектируемому сайту. По согласованию с преподавателем индивидуальные задания значительно отличаться от рекомендуемого ниже текста. Их состав должен быть подготовлен в соответствии с п. 3.3 настоящих методических указаний. Допускаются варианты с использованием технологии активных серверных страниц, реализация которой позволяет, например, производить запрос по многим ключам к удаленной базе данных, с указанием таких критериев поиска, как фамилия автора, год издания, издательство, ключевые слова, язык документа и т.д. Ниже изложены задания, указывающие минимальные требования к сложности задания.

4.1. ПРОЕКТИРОВАНИЕ СТРАНИЦЫ АНКЕТЫ. Создайте пустую веб-страницу. Настройте цвет фона, шрифты и другие свойства страницы, аналогичные ранее размеченным страницам сайта. Добавьте на страницу объект «форма». Поместите в форму группу кнопок выбора (radiobutton), комбинированный список с выпадающими ответами, несколько строк ввода (обязательны поля ввода адреса e-mail и номера телефона), список с возможностью выбора нескольких вариантов ответа и область ввода текста. Дайте форме и ее объектам понятные («говорящие») имена. Не следует называть форму “f”, радиокнопки “r”, “rr” или “rrr”, а область ввода текста “a”. Оформите страницу в соответствии с правилами дизайна. Пример анкеты показан на рисунке 3.2.

Рис. 3.2. Форма анкеты посетителя сайта

Добавьте ссылку на анкету в меню сайта.

4.2. СОЗДАНИЕ СКРИПТОВ ДЛЯ ПРОВЕРКИ ПРАВИЛЬНОСТИ ЗАПОЛНЕНИЯ ПОЛЕЙ ФОРМЫ.

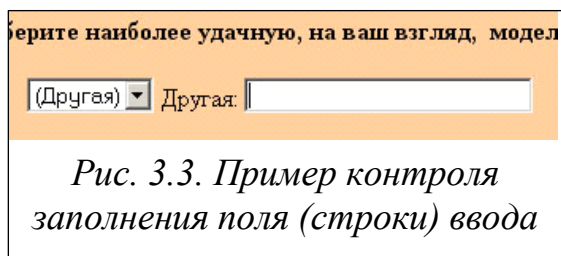
4.2.1. Самый надежный способ ненавязчивого контроля правильного ввода пользователем необходимой вам информации – создать обработчик события onBlur (элемент формы из активного превращается в неактивный). Тогда проверка производится после заполнения поля, в момент перехода к следующему. Можно использовать обработчик onChange, реагирующий на каждый вводимый символ, но это лишит вашего потенциального клиента права на опечатку или на создание экзотического псевдонима. Постоянно всплы-

вающие в процессе ввода информации сообщения могут заставить даже лояльного пользователя отказаться от сотрудничества с вами. Более того, часто такой способ просто неприемлем (например, для проверки правильности ввода адреса e-mail). Дополнительно организуйте функцию общей повторной проверки всех полей формы по событию submit (отправка формы). Опыт показывает, что «изобретательные» корреспонденты легко находят способ обойти событие onBlur.

4.2.2. Для поля ввода имени (см. рис. 3.2) разрешенными символами являются буквы и дефис. Если желательно допустить ввод фамилии, инициалов или имени и отчества, то следует разрешить пробелы и точки. Существуют два варианта скрипта проверки правильности ввода имени: один проверяет наличие разрешенных символов, другой – наличие запрещенных. В обоих случаях в функции проверки задается строка запрещенных или разрешенных для использования в имени символов, а затем в цикле проверяются значения символов из поля ввода имени. С помощью функции indexOf или search проверяется присутствие или отсутствие текущего символа в запрещающей или разрешающей строке. По факту обнаружения несоответствия символа в поле ввода имени указанным ограничениям на экран с помощью функции alert выдается соответствующее сообщение («Обнаружен неверный символ»), курсор помещается в поле ввода имени (метод focus), а вся строка выделяется (метод select), чтобы дать возможность пользователю удалить ее одним нажатием клавиши.

4.2.3. Функции проверки ввода номера телефона или факса принципиально не отличаются от проверки ввода имени. Единственным отличием является то, что при таком виде проверки проще задать строку разрешающих символов, так как их будет заведомо меньше, чем запрещенных. В номерах телефона и факса разрешено указывать цифры, а также скобки, дефис и плюс. Действия системы по факту обнаружения нарушений аналогичны перечисленным в пункте 4.2.2.

4.2.4. Содержимое комбинированного списка для выбора альтернативного варианта ответа в контроле не нуждается, оно введено автором на этапе дизайна. Нужно проверить, введено ли что-нибудь в строку альтернативного варианта ответа (рис. 3.3). Правила дружественного интерфейса советуют



при выборе варианта «Другая модель» автоматически – с помощью обработки события onChange элемента ComboBox – переводить фокус ввода на строку альтернативного варианта. В этом поле ввода допустимые комбинации символов должны принадлежать множеству всех

известных моделей фирмы.

4.2.5. Функция проверки ввода адреса e-mail может быть реализована несколькими способами. Самый простой из них – использовать функцию search. Более надежный и трудоемкий – последовательно перемещаться по строке, аналогично процедуре проверки имени, выставляя флажки-признаки

наличия встреченных символов, проверяя их повторение и т.д. В обозначении e-mail адреса необходимо удостовериться в использовании только разрешенных символов. Это буквы, цифры, символы «_» и «-». Символ «@» должен присутствовать только один раз. Устанавливается наличие символа «.» (точка), отсутствие точки непосредственно после или перед символом «@», отсутствие лидирующей и замыкающей точки в обозначении адреса, отсутствие следующих друг за другом точек (многоточия), а также возможность указания доменов третьего и более высоких уровней (комбинации вида ghf56@hj.dhh.aaa.des). Действия по факту обнаружения нарушений аналогичны указанным выше (см. рис. 3.4). Отличие состоит в том, что сообщения об ошибках должны четко указывать, что именно некорректно в строке, например, «В адресе e-mail встречено два символа @».

4.2.6. Поле ввода текста может быть заполнено произвольным набором символов. Следует проверить, не осталось ли оно пустым.

4.3. СТРАНИЦА ГОСТЕВОЙ КНИГИ. Создайте пустую страницу. Настройте ее свойства по аналогии с имеющимися страницами сайта и разместите на ней объект «форма» с полями ввода (рис. 3.5).

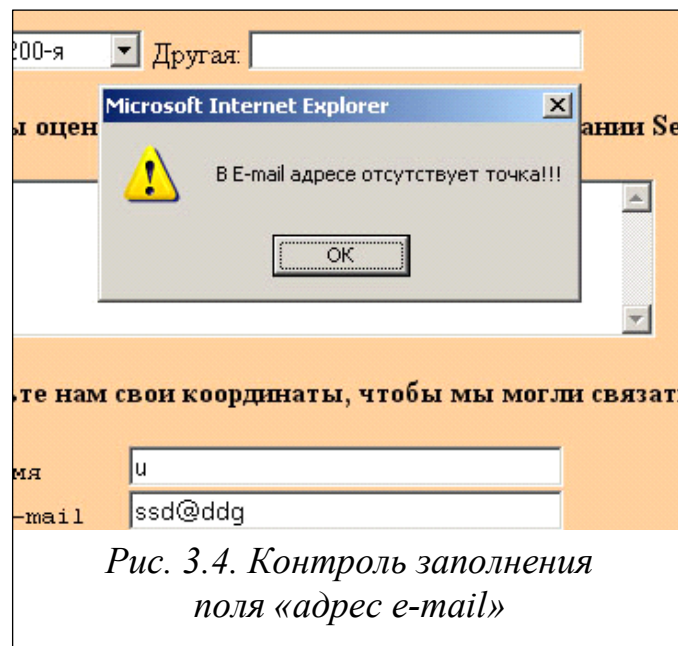


Рис. 3.4. Контроль заполнения поля «адрес e-mail»

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет о работе должен содержать следующие материалы в соответствии с полученным индивидуальным заданием:

5.1. Краткие формулировки назначения, области применения и функциональных характеристик проектируемого сайта (по материалам индивидуального задания).

5.2. Обоснование эффективности использования скриптов в разметке страниц создаваемого сайта (по материалам п.2 данного пособия).

5.3. Обоснование выбора визуальных элементов создаваемых документов (собственного или по п. 4 настоящего пособия).

5.4. Таблица (по аналогии с табл.3.1 пособия), представляющая информацию о событиях прерывания, используемых в лабораторной работе.

5.5. Программные коды обработчиков прерываний по событиям с комментариями или блок-схемами (по материалам п. 2.2 настоящего пособия).

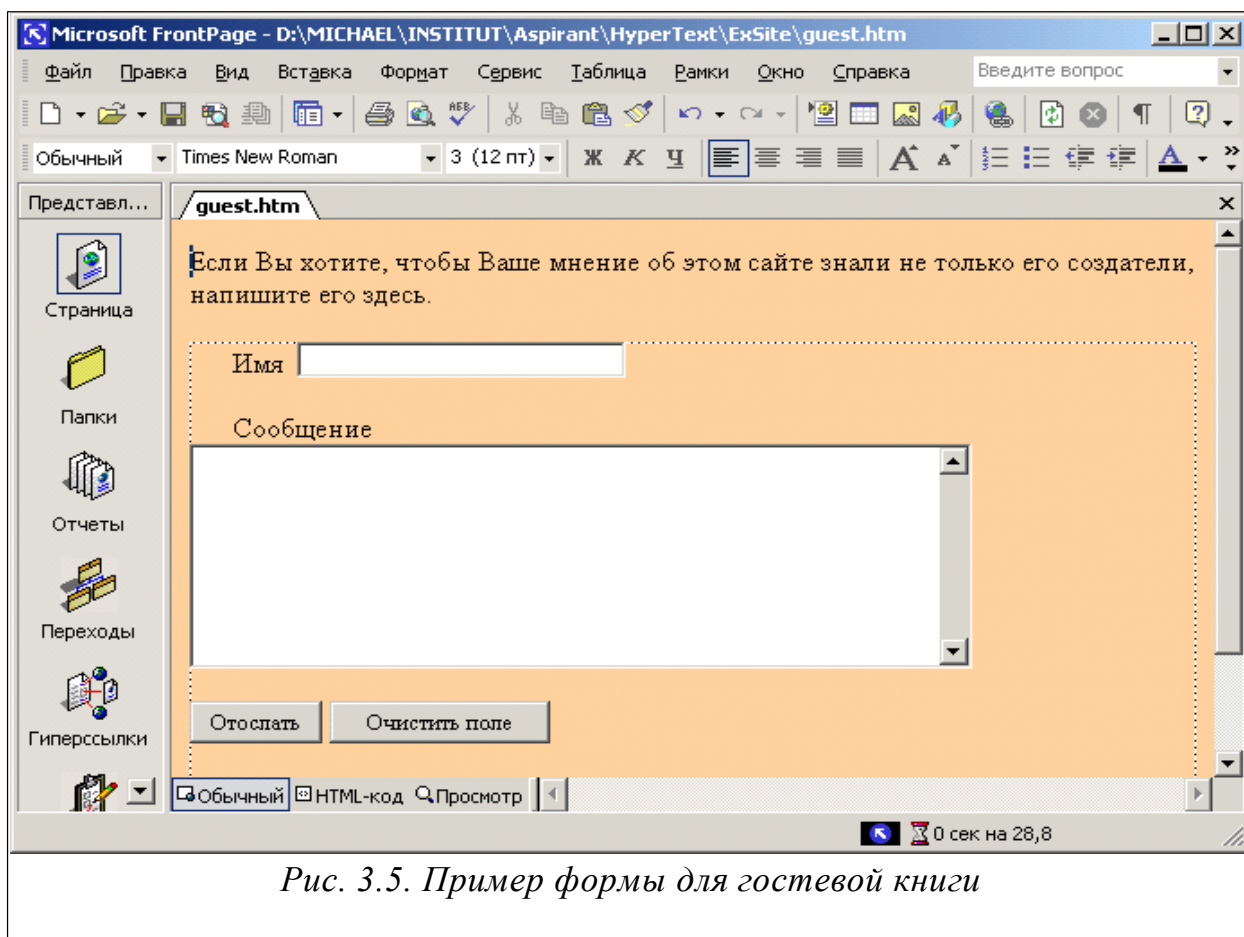


Рис. 3.5. Пример формы для гостевой книги

- 5.6. Содержание выполненных индивидуальных занятий.
- 5.7. Ответы на контрольные вопросы и выводы по выполненной работе.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

6.1. Каковы назначение и функции скриптов (сценариев) в языке разметки гипертекстов? К какой категории языка они относятся - это элементы, атрибуты, комментарии, ссылки, или они не входят ни в какие другие категории HTML?

6.2. В каком разделе документа HTML предпочтительно располагать элемент `<SCRIPT>`?

6.3. Каким образом прерывание от внешнего устройства или программного процесса активизирует выполнение скриптов? Постройте универсальную схему обработки прерываний.

6.4. Какие задачи решаются с помощью скриптов для придания динамики работе с гиперссылками? С какой целью программный код скриптов заключается в теги, воспринимаемые устаревшими браузерами как комментарии?

6.5. Элемент `<A>` (якорь) предназначен как для идентификации цели гиперссылки (атрибуты *id* и *name*), так и для указания адреса связанного ресурса (атрибут *href*). Воспримет ли система документ, в разметке которого не определены значения ни одного из этих атрибутов?

6.8. Какое противоречие содержит в себе рис. 3.1? Какие элементы представленного изображения несовместимы в реальной ситуации?

6.9. Могут ли программные коды сценариев составлять содержание таблиц стилей и можно ли с их помощью автоматически настраивать таблицы стилей в зависимости от конкретных характеристик компьютеров пользователей?

ЛАБОРАТОРНАЯ РАБОТА 4

ДОРАБОТКА ДИЗАЙНА ВЕБ-САЙТА

1. ЦЕЛЬ РАБОТЫ

Целью работы является развитие навыков создания дизайна, а также исследования его характеристик с целью повышения качества воспроизведения материала. Особое внимание уделяется построению единого стиля оформления, удобству навигации, методам ускорения загрузки и оценки посещаемости сайта.

2. СВЕДЕНИЯ ИЗ ТЕОРИИ

2.1. СТИЛЬ ДИЗАЙНА

4.1.1. ОСОБЕННОСТИ СТИЛЯ. Сайт, обладающий хорошим стилем [1-3], характеризуется, прежде всего, своей индивидуальностью. Каждая страница публикации имеет единый почерк, ярко выраженную принадлежность ко всей подборке документов. Наряду с общностью зрительного представления, хороший стиль структуры позволяет посетителю легко ориентироваться в страницах, и поиск информации не угрожает ни опасностью потерять цель перемещений, ни напрасной потерей времени. Еще одно качество – страницы загружаются менее чем за минуту. Стильность web-сайта достигается разными приемами. Можно выделить следующие элементы создания стиля:

- шрифт – в пределах публикации должен иметь одинаковые характеристики, такие как гарнитура (начертание), кегль (высота), цвет;

- абзац – желательно, чтобы преобладал какой-нибудь один из видов выравнивания на странице, например, публикация сделана с отступом от левого края и выравниванием влево;

- цветовая палитра web-сайта предпочтительно опирается на три доминирующих цвета страниц, которые используются для представления обычного текста, ссылок и посещенных ссылок;

- цветовая схема должна выдерживаться на всех страницах публикации, чтобы создавать у посетителя ощущение связности сайта. Цвета ссылок выбирают таким образом, чтобы, с одной стороны, читатель видел, что это

ссылка, а с другой стороны, чтобы она не мешала ему читать основной текст. По поводу ссылок есть два полезных замечания: первое – как бы вам ни хотелось сделать цвета ссылок и посещенных ссылок одинаковыми (без веских на то причин), постарайтесь перебороть себя и придать им немного различия, для этого просто сделайте цвета ранее посещенных ссылок чуть темнее; и второе – исторически сложилось, что подчеркнутый текст в Web символизирует ссылку, поэтому не используйте подчеркивания в тексте, привлекайте внимание к важным фрагментам другим способом. Читатель машинально щелкнет мышью подчеркивание, и не сразу поймет, почему не выполняется переход по ресурсам;

➤ графическое оформление сайта, во-первых, должно подчиняться общей цветовой схеме; во-вторых, вы должны продумать единую концепцию графического оформления. Все графические элементы можно условно разделить на два класса: рисованные и фотографические. Постарайтесь не смешивать эти два типа в оформлении. Кроме этого, в случае, если вы используете фотографии в качестве иллюстраций, то сначала обработайте их – сделайте тоновую и цветовую коррекцию, кадрирование, выберите примерный размер, найдите способ обработать края фотографии. Всегда пишите пояснения к фотографиям, это будет восприниматься как подпись и, кроме того, избавит пользователя от ожидания загрузки в случае, если его не интересует данная страница.

➤ навигация по сайту – именно она не дает посетителю запутаться в переходах. Всегда оставляйте возможность возврата на главную страницу. Не забывайте, что очень многие попадают к вам через поисковые системы, т.е. не на первую страницу, и нужно позволить читателю перейти на нее. Для этого дублируйте навигационную систему на всех страницах или сделайте так, чтобы они хотя бы обеспечивали переход на основную страницу. Если навигационная панель выполнена графическими средствами, то обязательно сделайте и поместите внизу ее текстовую копию, так как текст загружается быстрее графики.

4.1.2. ШРИФТОВОЕ ОФОРМЛЕНИЕ. Всего лишь несколько лет назад в дизайне ощущалась острая нехватка компьютерных шрифтов как по количеству, так и по качеству, однако за последнее время ситуация резко изменилась. Появился огромный выбор шрифтового материала любых стилей, что поставило дизайнера перед другой проблемой – проблемой эстетики и вкуса.

Интернет-технологии накладывают специфические ограничения использования шрифтов в оформлении документов. В частности, нельзя однозначно определять шрифт для оптимального вывода текста на экран читателя. Переопределение `<FONT face=[гарнитура шрифта]>` не является панацеей. На клиентской машине просто может не оказаться нужного шрифта, а замена его другим, выбранным системой по умолчанию, приведет к появлению плохо читаемого документа (рис. 4.1). Как результат, считается правилом хорошего тона не применять элемент FONT. Указанная проблема еще ждет своего решения.

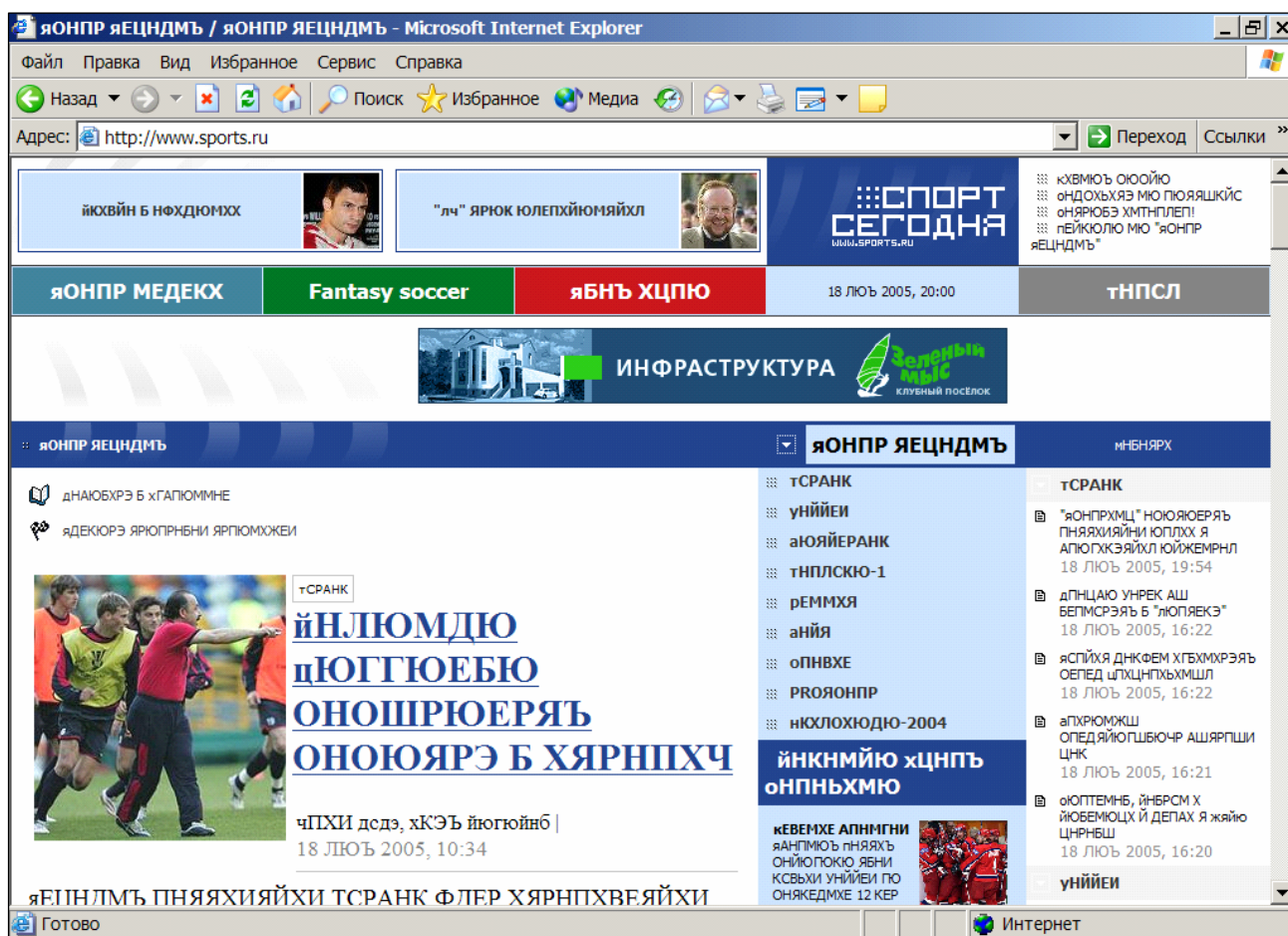


Рис. 4.1. Пример неудачного выбора кодировки символов шрифта

Сегодня популярны другие пути для оформления "тела" документа. В частности, выделение необходимых частей текста цветом, применением прописных символов, их *курсивного* и полужирного начертания, а также манипуляции с кеглем шрифта, выводимого на экран клиента. Важным для удобства восприятия информации оказывается цветовое соотношение текста и фона. Если предполагается передача пользователю большого объема информации, то с точки зрения удобочитаемости желательно, чтобы текст был выполнен темным цветом на светлом фоне, и чтобы они были достаточно контрастными. Применение ярких фоновых подложек является сильным выразительным средством, но необходимо помнить, что читать текст по яркому, а тем более, по пестрому фону иногда просто невозможно, так как буквы теряются в его живописи.

Одним из показателей читабельности является ширина строки документа. С появлением мониторов, поддерживающих большое разрешение экрана, стало возможным "упаковывать" в строку до нескольких сотен букв, однако строка "идеальной ширины" должна содержать 50-70 знаков. При большем количестве скорость чтения замедляется, и утомляемость наступает значительно быстрее.

Нередко при создании проектов следует учитывать, что удобочитаемостью можно иногда пожертвовать в пользу определенных стиливых и эстетических решений. Это в первую очередь относится к созданию продуктов, вы-

полняющих, например, рекламные функции. Такой "плакат" должен вызывать поток посетителей, какие бы "антигуманные" средства для этого ни использовались, вплоть до полного пренебрежения удобствами чтения.

Следует всегда помнить о способности шрифта привлекать или отвлекать внимание. Его эстетика открывает широкий простор для изобретательности и применения специальных эффектов, так как без визуального членения и акцентирования смысла частей, всего вероятнее, текст останется непрочитанным просто потому, что его не заметят в океане информации. Конечно, при этом не следует забывать об оптимизации Web-графики для облегчения доступа к ресурсам. При нарушении этого правила возрастает риск того, что ваш труд останется невостребованным просто в силу технических причин.

2.2. УЛУЧШЕНИЕ ХАРАКТЕРИСТИК ПОЛЬЗОВАНИЯ САЙТОМ

2.2.1. УДОБСТВО ПОЛЬЗОВАНИЯ. «Существуют неписанные правила, позволяющие не допускать ошибок на пути создания удобства пользования страницей. Они учат использовать пространство, отводимое для отображения страницы, учат уделять внимание мелочам, не раздражать посетителя, а напротив – располагать его к себе. Содержание, конечно, играет важнейшую роль в проявлении интереса к вашему проекту, но следует позаботиться, чтобы оно было комфортно подано. Если полезная нагрузка экрана составляет менее 25 процентов от его площади, или пользователь должен вновь и вновь загружать страницы, чтобы дочитать интересную статью, которую автор разбил на множество кусочков, то посетитель рано или поздно откажется от дальнейшего просмотра и вряд ли вернется. Наша же задача – не только удержать его, но и преподнести материал так, чтобы возникало желание возвращаться вновь и вновь. Страницу следует четко разбить на области – область для навигации, область полезной информации и область для рекламы, если вы не в состоянии от нее отказаться. Область, отводимая для рекламы, не должна превышать 10% площади страницы. Области навигации и основной информации не должны перекрывать друг друга. Необходимо предельно ясно показать пользователю, что и где расположено, и выдерживать подобный стиль на протяжении всех страниц сайта».

Приведенный выше абзац принадлежит одному из авторов сайта, главная страница которого представлена на рис. 4.2. Нетрудно подсчитать, сколько провозглашенных «правил хорошего тона» нарушено здесь на практике. Заметим, что чрезмерное стремление к неожиданным решениям может довести некоторые советы до абсурда. Изысканный силуэт сосуда на рис. 4.3 при первом знакомстве, безусловно, способен увлечь посетителя предвкушением интеллектуального общения с автором. Прием достаточно спорный, так как проектированием документов HTML, как правило, занимаются натуры не столько иронично-созерцательные, сколько задорно-деятельные. Более того,

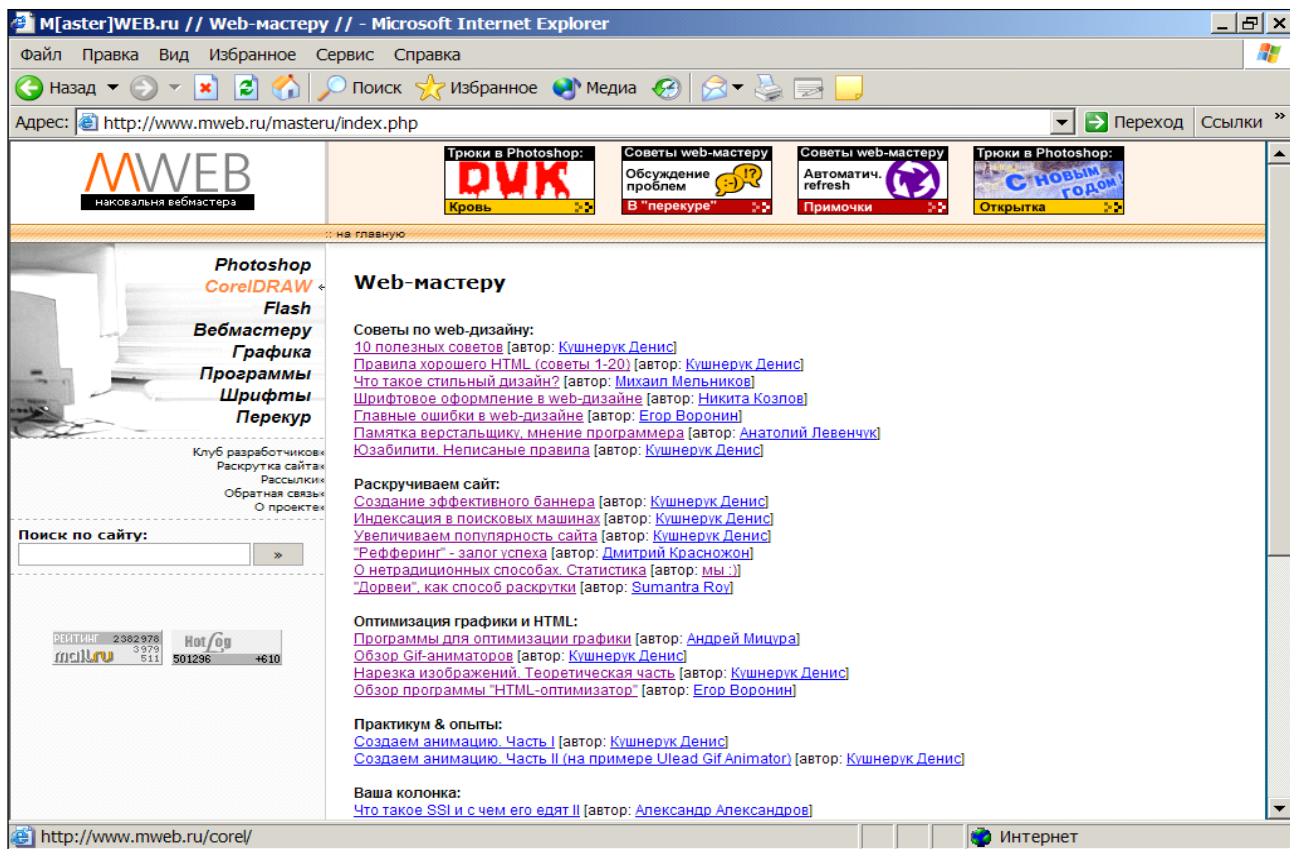


Рис. 4.2. Пример нерационального использования площади страницы

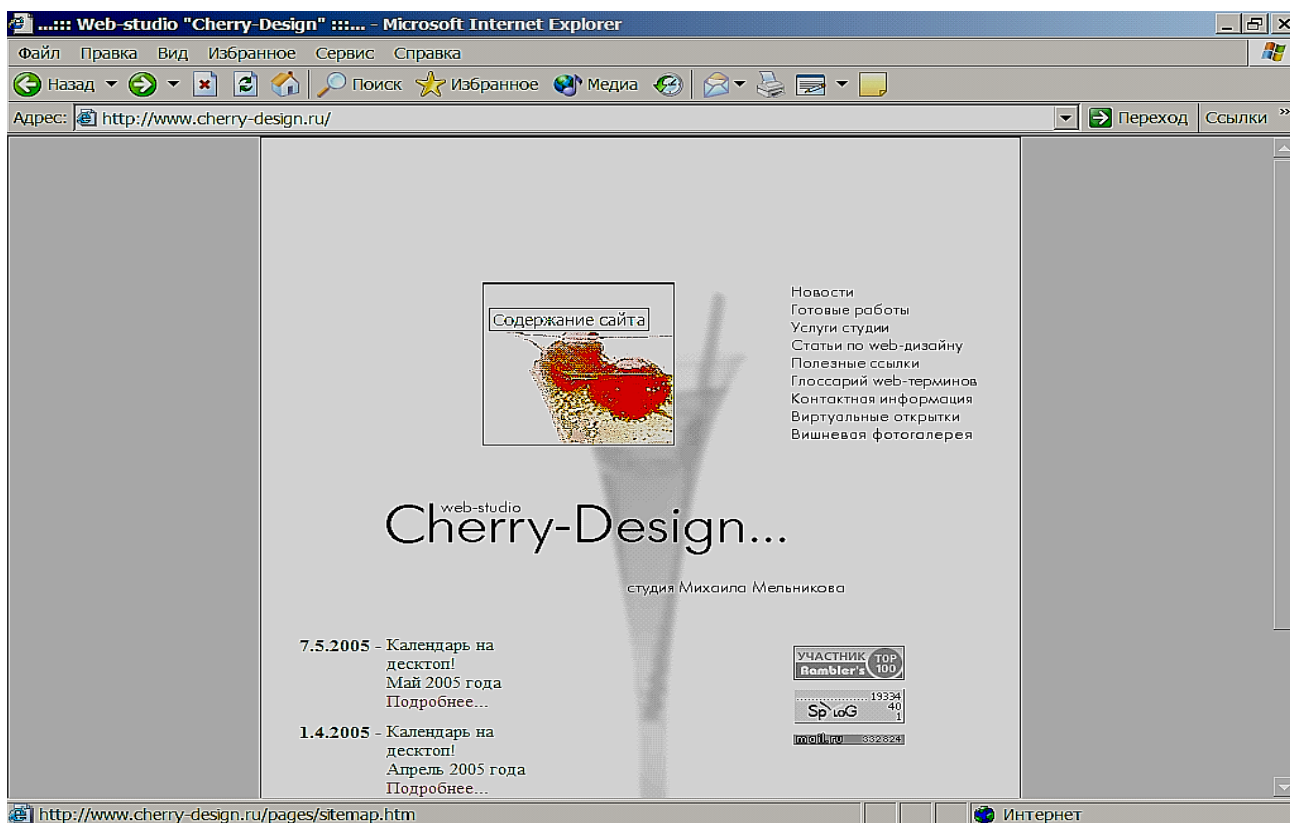


Рис. 4.3. Дискуссионная символика главной страницы

наведение курсора мыши на изображение вишен вызывает всплывающую

подсказку «содержание сайта», разбивая все иллюзии, за исключением, быть может, «*in vina veritas*».

Лучшие сайты – быстрые. Трафик прямо зависит от размера загружаемых страниц. Сократите объем графики (в разумных пределах) – возрастет трафик. Это экспериментально подтвержденные специалистами данные. Не увлекайтесь «тяжелой» графикой, чем дольше загрузка, тем больше пользователей уйдут, не дождавшись ее окончания. Предельно допустимый лимит страницы – 100 кб, в идеале 33-34 кб. Страница размером 34 килобайта загружается ровно одну секунду на скоростном канале T1 или 10 секунд на хорошем аналоговом соединении. Разумный предел составляет 60-70 килобайт, допустимый предел 100. Если размеры ваших страниц превышают его – не рассчитывайте на хороший трафик и длинные сеансы серфинга по вашему сайту.

В настоящее время в сети популярны два графических формата: GIF и JPEG. Основное различие между ними состоит в том, что предназначены они для хранения разных видов графики. GIF преимущественно используется для сохранения рисованных изображений, векторной графики, картинок без полутонов, градиентов (переходов от цвета к цвету) и большого количества мелких деталей разного цвета, тогда как JPEG – для фото и полутоновой графики, живописи, градиентов и изображений с множеством мелких разноцветных деталей.

Оба формата являются компрессионными, то есть данные в них уже находятся в сжатом виде. Сжатие, тем не менее, представляет собой предмет выбора решения. Каждый из этих форматов имеет ряд настраиваемых параметров, позволяющих управлять соотношением «качество-размер» файла. Таким образом, за счет сознательного снижения качества изображения, практически не влияющего на восприятие, можно добиваться уменьшения объема графического файла, иногда значительного.

2.2.2. АНАЛИЗ СТАТИСТИКИ ПОСЕЩЕНИЙ САЙТА. Сегодня ни один грамотно поддерживаемый сайт не обходится без анализа данных о его посетителях. Статистика необходима, во-первых, для наблюдения за темпом посещаемости. Во-вторых, она открывает дополнительные резервы роста вашей популярности.

Для набора статистики следует воспользоваться счетчиком. В зависимости от правил владельца, его можно размещать как на одной странице, так и на всех. Второй вариант предпочтителен, потому что почти каждое действие пользователя будет зафиксировано, и статистика окажется более полной.

Напомним два термина. Хост – это уникальный посетитель вашего сайта. Другими словами – если один и тот же человек побывал у вас на сайте в течение дня несколько раз, то на счетчике хостов он будет засчитан только однажды. Хит – это количество загрузок всех страниц сайта, если счетчик установлен на каждой странице. Если один посетитель в течение одного дня возвращался к вам десять раз или в течение одного сеанса просмотрел десяток страниц – на счетчике хитов отобразится именно это количество. Сумма будет накапливаться с приходом каждого пользователя.

Информация о посещениях сайта поможет подобрать пути его «раскрутки» или ликвидации тех факторов, которые препятствуют этому процессу. Зафиксируйте в конце дня данные о количестве хитов и хостов на вашем сайте. Подсчитайте отношение хиты/хосты, и получившаяся цифра даст вам представление о том, сколько в среднем страниц загружает один посетитель. Если она окажется меньше трех – это свидетельствует, что у вас не все в порядке либо с навигацией, либо с отображением. Следует обратить внимание на содержание и скорость загрузки. Без устранения недостатков раскрутка не состоится.

Далее следует изучить наиболее популярные пути по сайту или наиболее популярные разделы, и уделить внимание менее посещаемым, может быть, и вовсе заменить их. Эти действия представляют собой первоначальную подготовку к раскрутке. Следующий шаг – заимствование "рефералов" Так называются ссылки на страницу, с которой к вам пришел пользователь. Это – один из важнейших показателей статистики. Вы можете контролировать источники своей аудитории. Известно, что сайт, зарегистрировавший счетчик, невольно попадает в своеобразный рейтинг. Обращаться нужно именно к нему. При этом неважно, зарегистрированы вы в этом рейтинге или нет. Из рейтинга в большинстве случаев можно получить доступ к статистике. Часть сайтов закрывает ее паролем, другая оставляет "свободной". В статистике "чужого" сайта, в отчете по "рефералам", можно найти наиболее прибыльные из них. При удачном исходе вы «вернетесь» с большой отдачей. Не забывайте и про рефералы в собственной статистике. Сайты, которые снабжают вас аудиторией, следует поддерживать и находить пути углубления сотрудничества.

Еще один способ повышения популярности – забота о своем рейтинге в поисковых машинах. Подготовьте список ключевых слов, которые полно и точно опишут смысл вашего проекта. Создайте "входные страницы" (дорвеи) для сайта, используя комбинации ключевых слов на странице. От текста, расположенного в начале страницы, зависит многое, практически все. Содержанием текста вы должны направить посетителя к главной странице. Вставьте ссылку на видном месте, чтобы читатель, не напрягаясь, мог увидеть ее. Избегайте лишних ссылок и баннеров – не давайте гостю возможности легко уйти от вас.

Руководствуйтесь следующими соображениями.

- Некоторые поисковые системы не анализируют элементы META, но используют первые слова на странице в качестве ключевых. Следовательно, начальные строки должны быть составлены с учетом этого обстоятельства.

- Обязательно наличие в них ключевых слов – по одному в предложении. При этом текст должен сохранять грамматический и содержательный смысл, а не представлять собой несвязную смесь фраз. Посетитель должен получать приятное впечатление от его чтения.

- Не делайте ваши параграфы слишком длинными. Каждый должен содержать не более трех или четырех предложений. Пользователи просто не будут читать большие громоздкие страницы.

➤ Старайтесь сделать так, чтобы текстовые ссылки включали в себя ключевые слова. Зачастую поисковые машины высоко ранжируют такие страницы.

➤ Если возможно, делайте ссылки на страницы, которые имеют ключевое слово в имени файла. Поисковые системы учитывают это.

➤ Не существует конкретного ограничения на количество слов в странице, но постарайтесь создать дорвей, содержащий 500-600 слов.

➤ Созданную страницу назовите ключевыми словами, разделенными дефисом, например: `travel-to-australia.html`. Это высоко оценят поисковики, которые обращают внимание на присутствие ключевого слова в URI.

Действенный сегодня способ повысить рейтинг состоит в том, чтобы создать ссылку на дорвей с вашей основной страницы, и сделать это так, чтобы ссылка существовала только для поисковой машины, а не для посетителя. Не маскируйте текстовые ссылки под цвет фона – 99% поисковиков занесут вас в черный список за такие проделки. Создайте незаметную картинку цвета фона. Назовите рисунок именем дорвея (`travel-to-australia.gif`). Расположите его в конце основной страницы и образуйте на нем ссылку на дорвей. В ALT рисунка также впишите ключевые слова. Не забудьте установить для рисунка `border=0`.

Создайте аналогичные дорвеи для всех ключевых слов, копируя рисунок на основной странице еще и еще раз, меняя его имя, ALT и ссылку. Затем пропишите основную страницу сайта в поисковые системы. Не регистрируйте все страницы. Разумеется, на обход всех страниц по образованным из главной страницы ссылкам понадобится много времени. Однако, в результате, поисковый робот, попадая на дорвеи и не распознавая их, ранжирует вас выше.

3. ПОДГОТОВКА К ВЫПОЛНЕНИЮ РАБОТЫ

3.1. Подготовить сменный носитель для сохранения результатов цикла лабораторных работ.

3.2. Ознакомиться с дизайном и структурой существующих веб-сайтов.

3.3. Обосновать предложения по доработке ранее созданного сайта для придания ему способности реагировать на события в системе, улучшения стиля представления, удобства навигации и т.д.

3.4. Написать коды скриптов для обеспечения запланированной реакции на события – движения курсора мыши, клавишные операции, загрузка и т.п.

3.5. Составить план действий за компьютером с распределением обязанностей между членами бригады.

3.6. Зафиксировать текст теоретических разделов отчета о работе.

3.7. Ответить на контрольные вопросы данных методических указаний.

4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Реализовать согласованные с преподавателем предложения по улучшению стиля дизайна сайта по п. 3.3 настоящих методических указаний.

- 4.2. Запустить проект и убедиться, что его работа соответствует замыслу.
- 4.3. Убедиться, что страницы выдержаны в едином стиле оформления.
- 4.4. Исследовать пути уменьшения времени загрузки и навигации в сайте.
- 4.5. Сократить файлы изображений за счет их нарезки и форматирования.
- 4.6. Предусмотреть конкурирующие и дублирующие схемы навигации.
- 4.7. Ввести в страницы счетчики посещений.

5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет о работе должен содержать следующие материалы.

- 5.1. Формулировка цели лабораторной работы (по п. 1 указаний).
- 5.2. Краткое описание известных путей улучшения стиля дизайна (п. 2).
- 5.3. Предложения по доработке дизайна ранее созданного сайта (п.3.3).
- 5.4. Сравнительный анализ показателей времени до и после доработки.
- 5.5. Коды вновь введенных скриптов (п. 3.4 настоящих указаний).
- 5.6. Изображения фрагментов с усовершенствованным стилем дизайна.
- 5.7. Ответы на контрольные вопросы и выводы по выполненной работе.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 6.1. Дайте определение стиля дизайна (п. 2.1).
- 6.2. Каково, на ваш взгляд, соотношение ценности содержания, удобства навигации и привлекательности оформления в проблеме достижения популярности и успеха сайта (п. 2)?
- 6.3. С какой целью рекомендуется выдерживать единство стиля (п.2)?
- 6.4. Почему изображения следует сопровождать пояснительным текстом?
- 6.5. Каким образом можно предотвратить неправильный выбор браузером клиента кодировки символов текста, чтобы избежать ситуации по рис. 4.1?
- 6.6. Какие меры повышения посещаемости сайта вам известны (п. 2.2)?
- 6.7. Существуют ли универсальные стили оформления и навигации по ресурсам, удовлетворяющие вкусам любого посетителя (п. 2)?

ЛИТЕРАТУРА

- 1. Веб-мастеру. Советы по веб-дизайну.: - <http://www.mweb.ru>, 2005.
- 2. Веб-студия. Статьи по веб-дизайну.: - <http://www.cherry-design.ru>, 2005.

3. Справочная информация и практические советы. Толковый словарь по веб-дизайну.: - <http://www.hostinfo.ru>, 2005.
4. Рудельсон Л.Е., Тверитнев М.М. Пособие по изучению дисциплины «Пакеты прикладных программ». М: - МГТУГА, 2004.
5. Омельченко Л.Н., Федоров А.Ф. “Microsoft FrontPage 2002. Самоучитель”. СПб.: - BHV, 2002.
6. Мэйнджер Дж. «JavaScript: основы программирования». Киев: - BHV, 1997.
7. Дарнелл Р. «JavaScript. Справочник». СПб.: «ИД Питер», 2000.