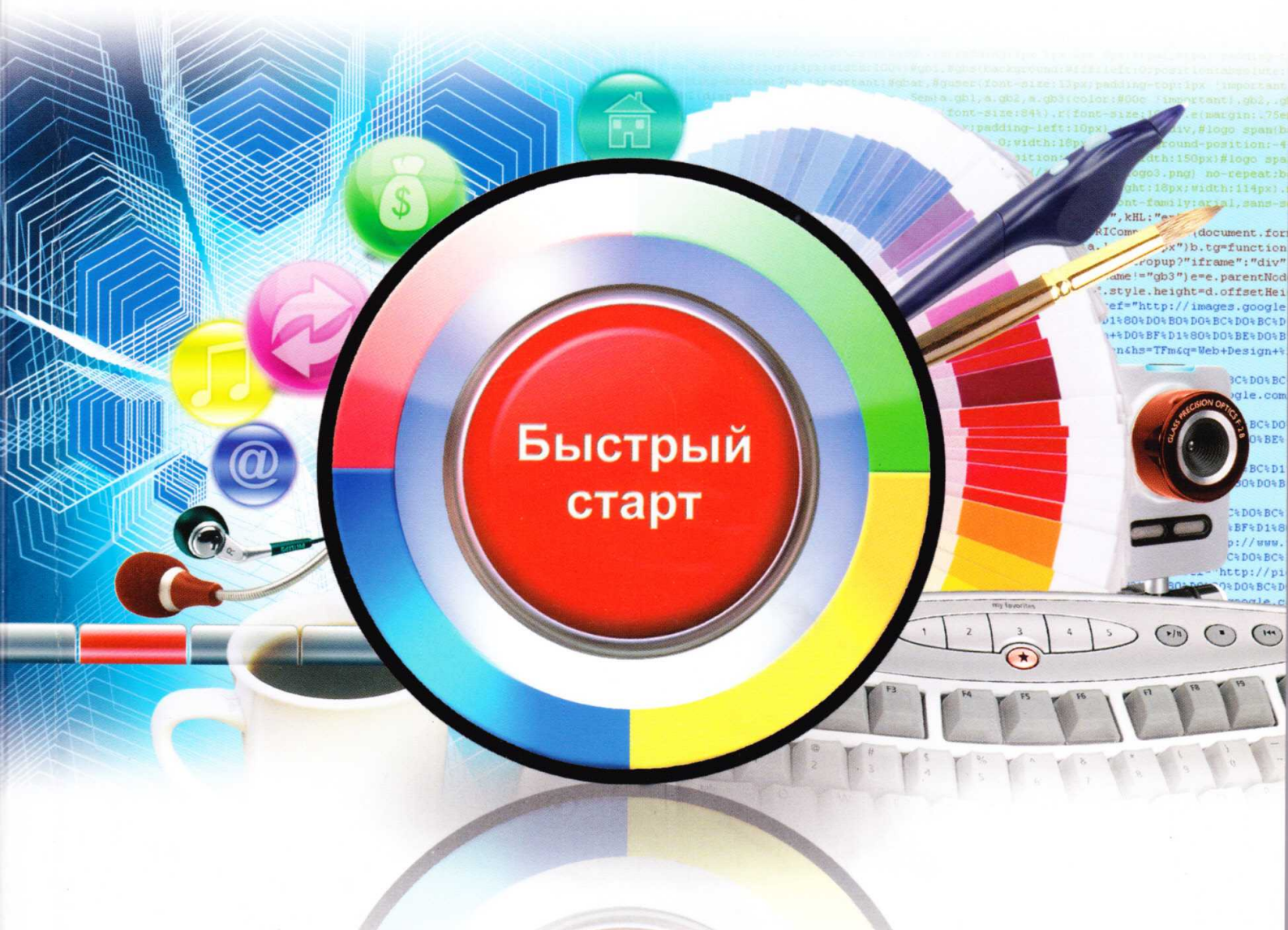


Е. Э. Самойлов

# Web-дизайн для начинающих



Быстрый  
старт

## Практическое руководство

Т Р И У М Ф

**Е. Э. Самойлов**

Практическое руководство

# Web-дизайн для начинающих

**Быстрый старт**

**«Издательство Триумф»**

**Москва**

**Самойлов, Евгений Эдуардович.**

С17 . Практическое руководство. Web-дизайн для начинающих : быстрый старт / Е. Э. Самойлов. — М. : Триумф, 2009. — 192 с. : ил. — (Быстрый старт). — ISBN 978-5-89392-430-5.

Агентство СІР РГБ

Эта книга содержит все необходимое для создания прекрасного web-сайта. Начав с создания простейших web-страниц, вы шаг за шагом построите собственный web-сайт, обладающий всеми качествами профессионального Интернет-портала. Вы познакомитесь с языком HTML и скриптами Java, украсите web-сайт графикой и наполните динамическим содержимым, а также научитесь работать в программе SharePoint Designer 2007 – профессиональном приложении для создания web-сайтов. Кроме того, вы узнаете, как разместить ваше творение в Интернете совершенно бесплатно.

После прочтения книги создание и размещение в Интернете собственного резюме, рекламного материала или галереи с вашими фотографиями станет просто увлекательным творческим занятием.

**ВАЖНАЯ ИНФОРМАЦИЯ ДЛЯ ТОВАРОВЕДОВ и ПОКУПАТЕЛЕЙ**

Обращаем ваше внимание на то, что настоящая книга издается в нескольких вариантах. Существуют версия издания с видеокурсом: «Самоучитель web-дизайна. Быстрый старт+Видеокурс», версия издания для более глубокого изучения: «Разработка, дизайн, программирование и раскрутка web-сайта» и версия издания для более глубокого изучения с видеокурсом: «Web-дизайн с нуля! Книга+Видеокурс». Выбирайте то издание, которое вам больше подходит.

Главный редактор издания	С. В. Черников
Ответственный редактор	Л. А. Каменская
Над текстом книги работал	Е. Э. Самойлов
Выпускающий редактор	И. Г. Колмыкова
Корректор	А. Н. Левина
Компьютерная верстка	Е. Н. Шахова
Дизайн обложки	Борис Ключко

ООО «Издательство Триумф»  
Россия, 125438, г. Москва, а/я 18.

Подписано в печать с оригинал-макета 22.09.2008 г.  
Формат 70×100<sup>1/16</sup>. Печать офсетная. Печ.л. 12. Заказ 5176.

Тираж 4000 экз.

Отпечатано в ОАО «Можайский полиграфический комбинат»  
143200, г. Можайск, ул. Мира, 93

# Содержание

## ГЛАВА 1.

<b>Устройство Всемирной паутины . . . . .</b>	<b>5</b>
Какие бывают web-сайты? . . . . .	6
Технологии создания web-сайтов . . . . .	12
Заключение . . . . .	14

## ГЛАВА 2.

<b>Первые web-страницы своими руками . . . . .</b>	<b>15</b>
Файлы HTML «изнутри» . . . . .	15
Создание первой web-страницы. . . . .	19
Основы языка HTML . . . . .	24
Ваш первый web-сайт . . . . .	42
Заключение . . . . .	64

## ГЛАВА 3.

<b>Приемы, без которых нельзя обойтись . . . . .</b>	<b>65</b>
Графика на web-страницах . . . . .	65
Использование звука и видео на web-страницах . . . . .	81
Карты ссылок . . . . .	87
Табличная верстка web-страниц . . . . .	90
Форматирование текстов . . . . .	99
Заключение . . . . .	105

## ГЛАВА 4.

<b>Web-дизайн и графика. . . . .</b>	<b>106</b>
Форматы графических файлов для web . . . . .	106
Источники изображений для web-страниц. . . . .	113
Обработка изображений в программе Adobe Photoshop . . . . .	119
Оптимизация кода HTML для web . . . . .	139
Заключение . . . . .	142

ГЛАВА 5.

**Web-сайт с динамическими страницами . . . . . 143**

Как работают статические и динамические web-сайты . . . . .	143
Создание клиентских сценариев на языке JavaScript . . . . .	145
Взаимодействие JavaScript и HTML . . . . .	166
Пример сценария JavaScript в HTML-документе . . . . .	170
Заключение . . . . .	179

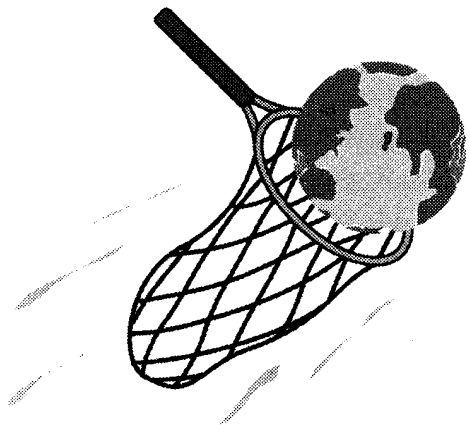
ГЛАВА 6.

**Размещение сайта в сети Интернет . . . . . 180**

Варианты размещения web-сайта в сети Интернет . . . . .	180
Загрузка сайта на сервер . . . . .	190

## ГЛАВА 1.

# Устройство Всемирной паутины



За неполные 17 лет своего существования Всемирная паутина очень сильно выросла. Настолько, что говоря об Интернете, люди часто подразумевают именно **WWW**, несмотря на то, что **web** является лишь одной из многочисленных услуг, предоставляемых Всемирной сетью. Чтобы получить доступ к **WWW**, даже не нужен персональный компьютер. Ресурсы **WWW** можно просматривать, хотя обычно и в несколько урезанном виде, с мобильных телефонов, карманных компьютеров, игровых приставок и даже при помощи бытовой техники, например некоторых телевизоров, микроволновых печей и холодильников.

Такие **web**-технологии, как **HTML** и **HTTP**, широко используются и отдельно от Интернета. Ярким примером этого являются, например, терминалы для проверки наличия свободных мест в поездах, стоящие на многих железнодорожных вокзалах.

На что же похожа современная система **WWW**? Больше всего — на всемирный «Информаторий», идея которого еще в середине XX века зародилась в умах многих писателей-фантастов. О такой системе упоминали в своих книгах братья Стругацкие, Роберт Хайнлайн и многие другие писатели, как известные, так и не очень. По сути, можно представить **web** как некую, невероятных размеров, базу данных, обращаясь к которой, пользователь может получить интересующую его информацию или, наоборот, передать свою информацию на тот или иной узел **web**.

Основой **WWW** являются **web**-серверы. В простейшем случае **web**-сервером называется компьютер, подключенный к Интернету и настроенный таким образом, чтобы Интернет-пользователи, подключающиеся к этому компьютеру с помощью **web**-браузера, могли просматривать **HTML**-файлы, на нем хранящиеся. На **web**-сервер устанавливают специальное программное обеспечение, называемое тоже **web**-сервером, которое обеспечивает работу этой системы. Чтобы избежать путаницы, компьютер, работающий **web**-сервером, мы будем называть просто сервером, а **web**-сервером будем именовать программное обеспечение. Отдельный **HTML**-документ на сервере называется страницей, а набор страниц, связанных друг с другом ссылками, объединенных общей тематикой и одним доменным именем, называется сайтом. На одном сервере могут одновременно работать сотни небольших сайтов, но бывает и наоборот: работу одного единственного сайта обеспечивают десятки и сотни серверов. Все зависит от размеров и посещаемости сайта.

## Какие бывают web-сайты?

Для каких целей создают сайты? Если сказать в двух словах, то для любых. Просто перечисление сфер человеческой деятельности и видов услуг, нашедших свое отражение в WWW, может занять не один десяток страниц, поэтому ограничимся кратким рассмотрением наиболее распространенных видов сайтов и web-сервисов.

- ✓ Web-представительства фирм (Рис. 1.1). Сейчас для практически любой компании, вне зависимости от области ее деятельности, наличие собственного web-сайта является правилом хорошего тона. Обычно такие сайты называют «визитками». На них размещают информацию об основных направлениях своей деятельности, делятся основными новостями компании, рассказывают о производимых товарах и новинках, а также предоставляют различные услуги.



Рис. 1.1. Web-представительство фирмы

- ✓ Промо-сайты. Сайты, создаваемые исключительно в рекламных целях. Для продвижения какого-нибудь товара, раскрутки новой торговой марки, рекламы нового фильма и т.д. Обычно отличаются минимальной информативностью и максимальной зрелищностью.
- ✓ Поисковые системы (Рис. 1.2). Хорошо известна поговорка, что «в Интернете есть все». Она, если и не полностью справедлива, то достаточно близка к истине. Информации в Интернете чрезвычайно много. Это порождает проблему поиска

в большом океане информации действительно нужной. Специально для этого были созданы поисковые системы, или просто «поисковики». Основой любой поисковой системы являются специальные поисковые «боты». Это программы, которые методично обследуют web-сайты Интернета, переползая по ссылкам с одного на другой, и составляют специальную «выжимку» содержания сайтов, называемую «индексом». Когда пользователь обращается к поисковой системе с запросом на поиск определенного слова или группы слов, «поисковик», на основе индексов, выдает пользователю список web-страниц, содержащих нужные слова. Положение страницы в списке определяется по ряду признаков: частоте употребления этого слова в документе, частоте посещений сайта, количестве ссылок, которые ведут на сайт с других web-сайтов, и т. д.

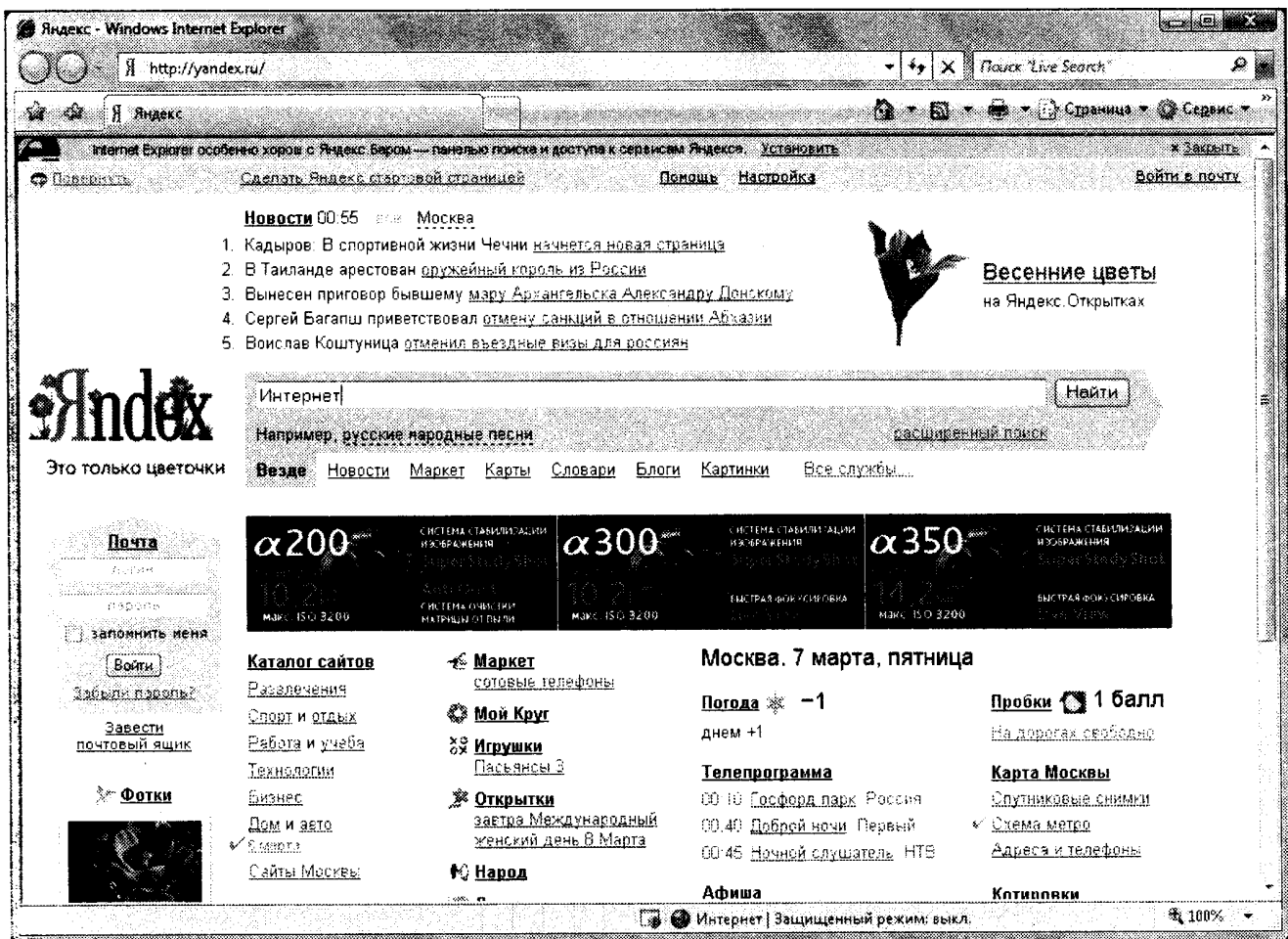


Рис. 1.2. Поисковая система Яндекс

- ✓ Порталы. Существует род сайтов, объединяющих в себе массу разнородной информации и услуг. Например, от службы знакомств до почтовой системы и энциклопедии. Сделано это для того, чтобы у пользователя не возникало необходимости обращаться к другим сайтам, а вместо этого он находился постоянно в пределах портала, принося владельцам доход от просмотренных им рекламных сообщений.
- ✓ Файловые хранилища. WWW заполнена всевозможными сборниками файлов: коллекции программ на все случаи жизни, сборники драйверов для компьютерного «железа», библиотеки музыки в формате mp3 и т. д.



- ✓ Архивы, библиотеки, энциклопедии, словари. Оправдывая звание всемирного Информатория, web содержит огромные залежи различных документов, частично сгруппированных на специальных сайтах. Часть этих архивов, в основном научных, доступна только за плату, но значительная часть информации находится в свободном доступе. Отдельного упоминания заслуживают web-энциклопедии, а точнее одна из них, носящая название **Wikipedia** (Рис. 1.3). Одна из ее главных особенностей в том, что в отличие от многих энциклопедий, составлением которых занимаются специально нанятые люди, содержимое **Wikipedia** пополняют сами пользователи Сети, причем весьма успешно.

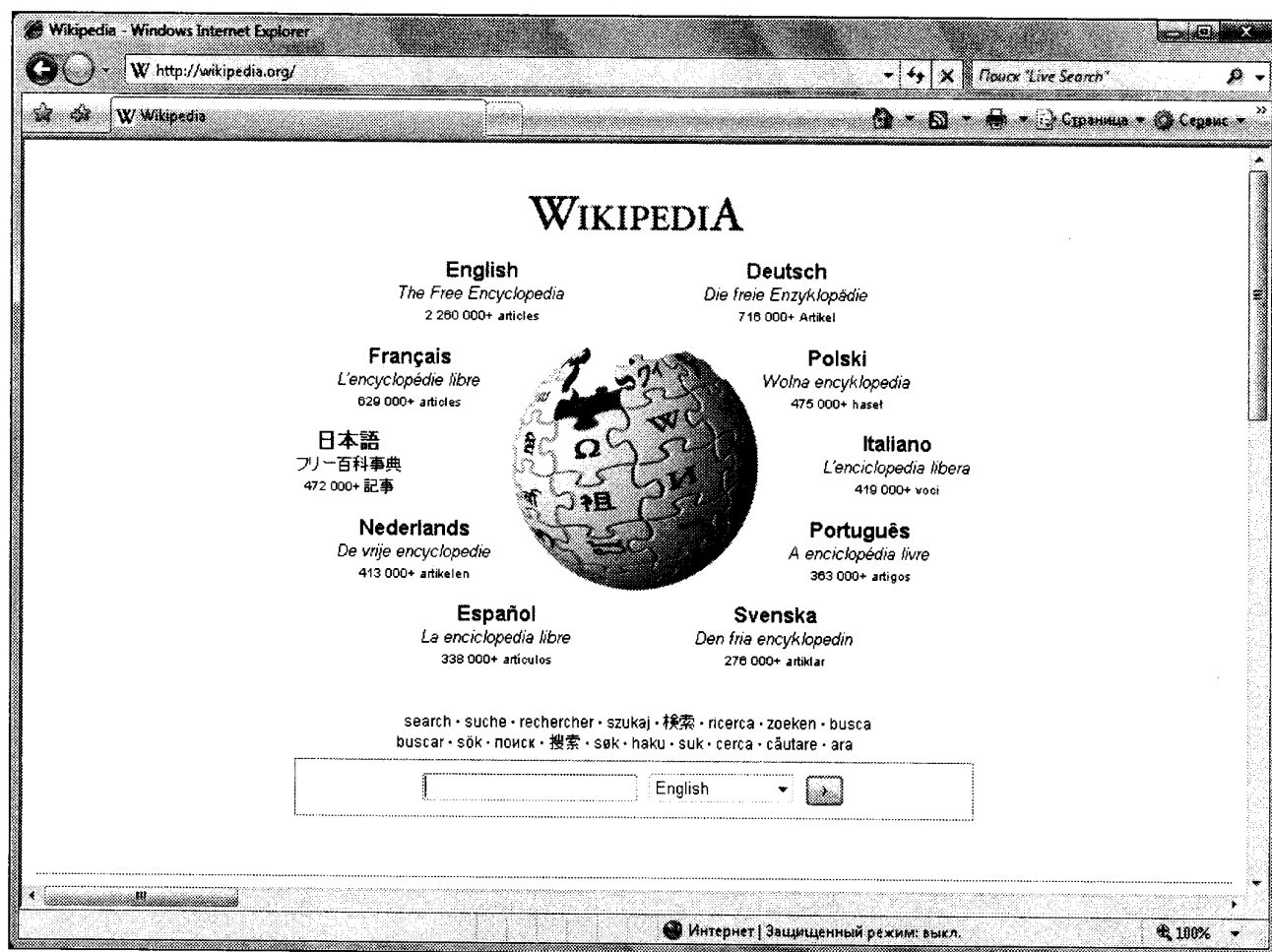


Рис. 1.3. Энциклопедия **Wikipedia**

- ✓ Электронные средства массовой информации (Рис. 1.4). С распространением web многие издательства стали создавать web-представительства своих журналов и газет, размещать в WWW новости, дайджесты новых номеров, а иногда и полностью тексты публикаций. Через некоторое время начали появляться и полностью «Электронные» СМИ, вообще не выпускающиеся в бумажном виде. Преимущество web-технологий для средств массовой информации очевидно – оперативность. В то время, как бумажные журналы и газеты выходят с определенной периодичностью: раз в месяц, раз в неделю, каждый день, информацию на сайте можно обновлять хоть каждые несколько минут. Кроме того, в Интернет-изданиях, в отличие от бумажных, практически нет проблем с размерами публикаций.



Рис. 1.4. Web-версия журнала

- ✓ Информационные сервисы. Доступность и простота обновления информации на web-сайтах привели к появлению во Всемирной паутине широкого спектра сайтов, специализирующихся на предоставлении часто изменяющейся информации. Например, прогнозов погоды, программ телепередач, расписаний сеансов в кино и театрах, и т. д.
- ✓ Техническая поддержка. Многие компании, особенно работающие в сфере высоких технологий, предоставляют своим пользователям ряд дополнительных услуг через свои web-сайты. Большинство компаний сотовой связи, например, предоставляют своим клиентам возможность проверить баланс телефона, просмотреть список последних операций, изменить список подключенных услуг, через персональную страничку пользователя на web-сайте компании.
- ✓ Домашние страницы (Рис. 1.5). Эта категория сайтов, возможно, одна из самых многочисленных. Многие пользователи web считают своим долгом оставить свой след в мире WWW и создают сайты, посвященные своему жизненному пути, своим увлечениям и интересам. Как правило, являются психологическим портретом автора домашней страницы.
- ✓ Фанатские ресурсы. Как и в реальном мире, в web постоянно возникают сообщества, посвященные культовым личностям, фильмам, сериалам или просто

кумирам молодежи. Как правило, такие сайты быстро наполняются совершенно исчерпывающей информацией о предмете поклонения, разнообразными мифами и сопутствующими деталями.

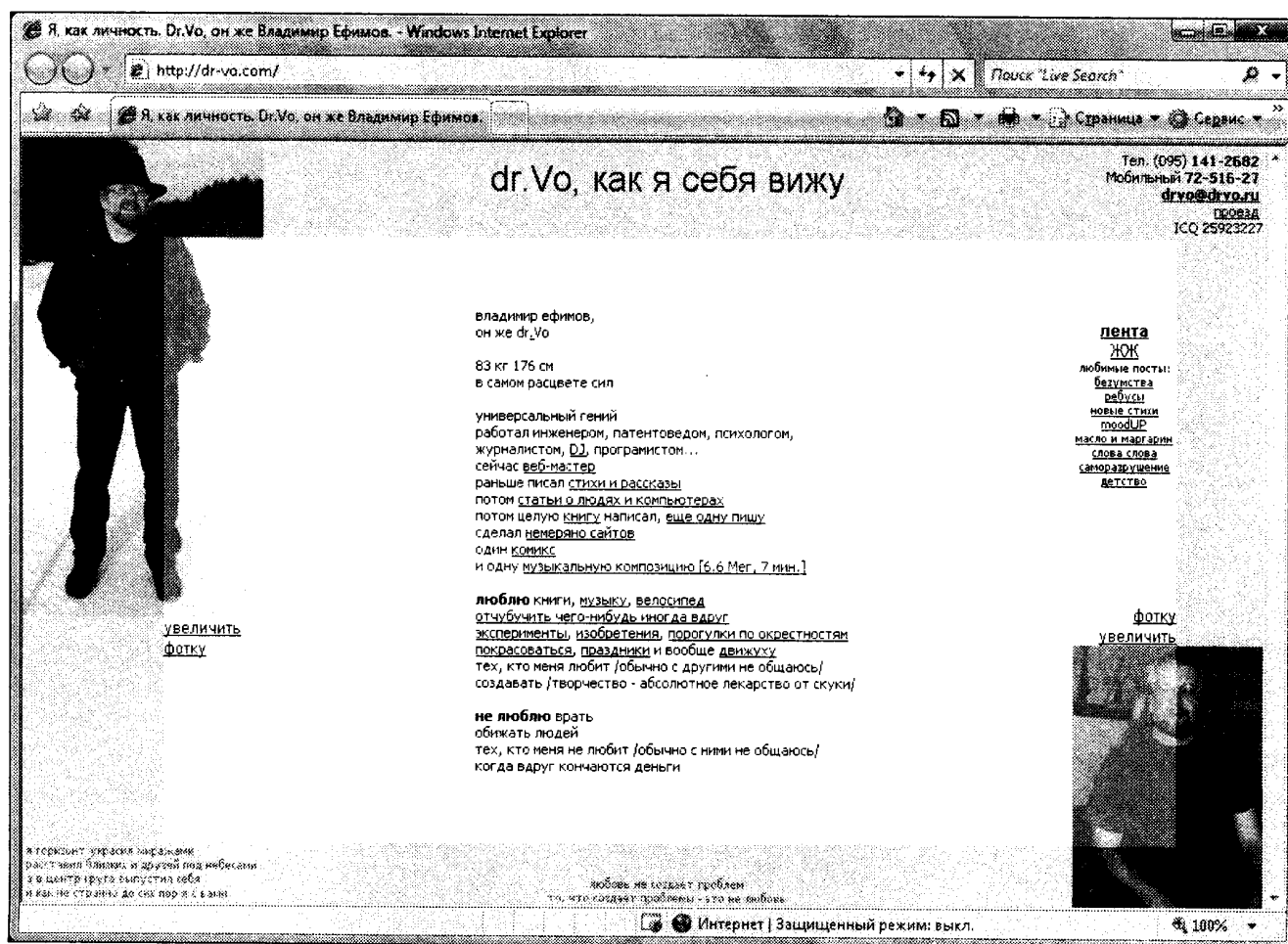


Рис. 1.5. Типичная домашняя страница

- ✓ Интернет-магазины (Рис. 1.6). Основой любой коммерческой деятельности является торговля. WWW также не исключение. Торгуют в Интернет-магазинах всем, чем угодно: от бытовой техники до хорошего настроения. Большой популярностью пользуются так называемые «онлайн-аукционы», самым известным из которых является аукцион **e-Buy**.
- ✓ Форумы (Рис. 1.7). Служат для организации общения посетителей сайта. Они представляют собой набор разделов, содержащих различные темы для обсуждения. Один из пользователей форума может задать вопрос или поместить сообщение по какой-то теме, остальные пользователи могут отвечать на этот вопрос и комментировать его. При необходимости можно просмотреть всю историю обсуждения того или иного вопроса.
- ✓ LiveJournal (Живой журнал, или просто ЖЖ) (Рис. 1.8). Средство, созданное специально для размещения в web личных дневников пользователей Сети. Члены сообщества LiveJournal и ему подобных могут писать свои дневники, читать дневники других участников этой системы, а также добавлять свои комментарии к чужим дневникам.

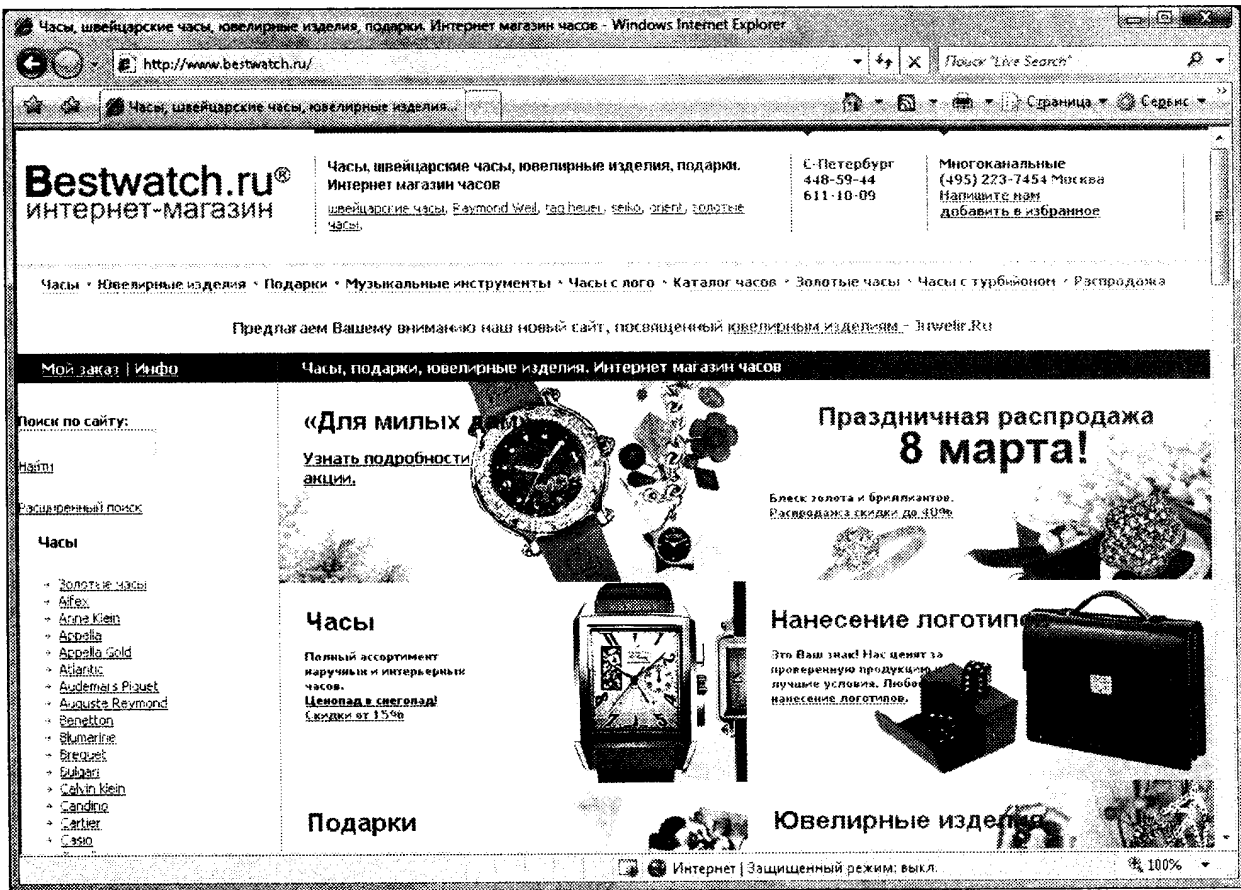


Рис. 1.6. Интернет-магазин

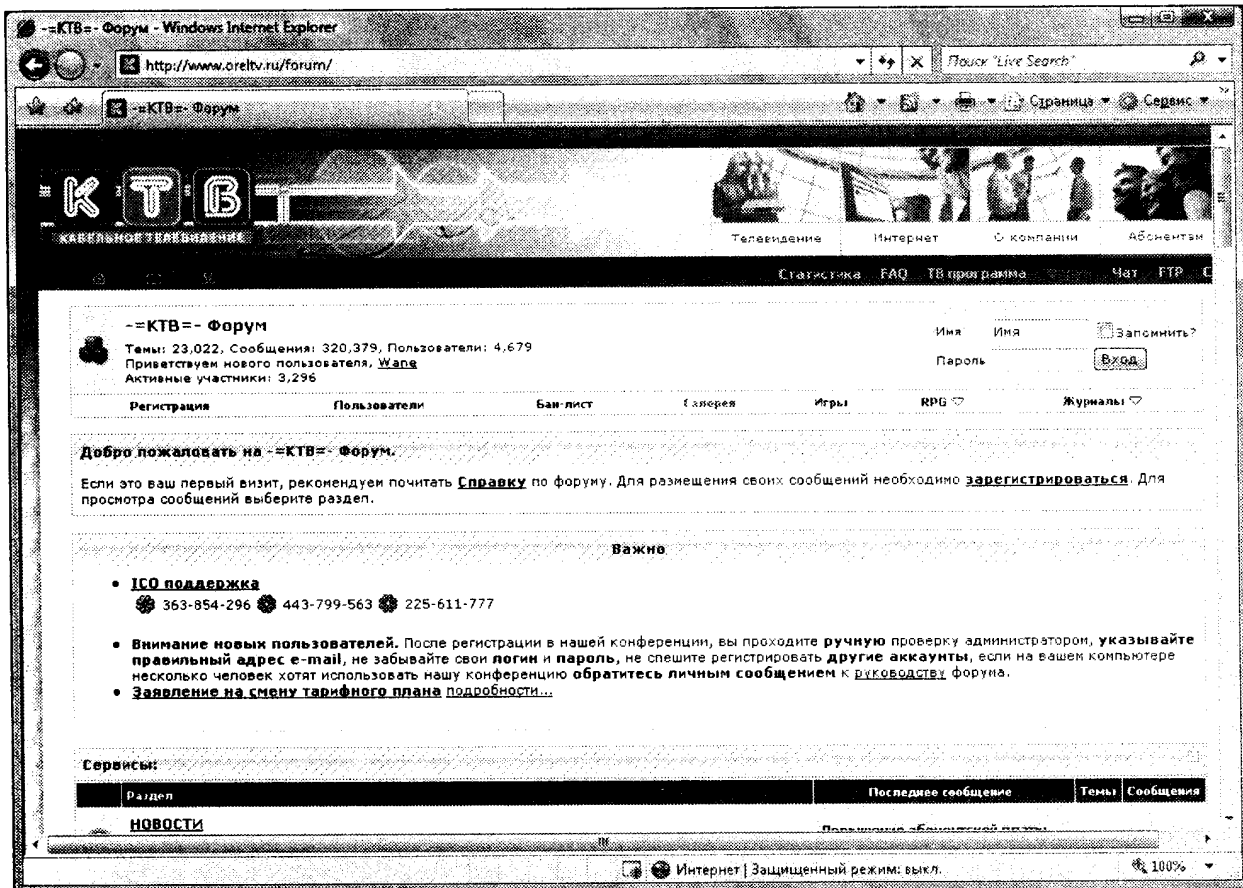


Рис. 1.7. Форум



Рис. 1.8. LiveJournal

## Технологии создания web-сайтов

Web-сайт в том виде, как было описано выше, т. е. состоящий просто из набора HTML файлов, ссылающихся друг на друга, создать просто, он неприхотлив в обслуживании, надежен и не требует от сервера большой вычислительной мощности. Такие web-сайты называют статическими.

Статический web-сайт всем хорош, пока его размеры невелики и информация на нем обновляется не слишком часто. Но если хоть одно из этих условий не выполняется, то все его плюсы быстро сходят на нет. Дело в том, что для внесения каких-либо изменений в статический сайт необходимо редактировать непосредственно сами HTML файлы. Это несложно, если вам нужно изменить пару страниц раз в месяц, но если это приходится делать каждый день, причем со всеми страницами сайта, особенно если их число переваливает за сотню, положение становится катастрофическим.

Кроме того, современные web-сайты все чаще делаются на основе некоторой базы данных, содержащей элементы его наполнения, и эти элементы необходимо показывать пользователю тогда, когда они ему требуются, и в том порядке, в котором ему нужно. Ярким примером таких систем являются Интернет-магазины. Где-то в недрах сервера хранится база данных, в которой содержится информация обо всех

товарах, которые продаются в этом магазине: названия товаров, цены, фотографии, различная дополнительная информация. Допустим, что этот магазин торгует часами. Пользователь может пожелать просмотреть все мужские часы определенных марок, обладающие нужными ему функциями и стоящие не дороже определенной суммы.

Третий недостаток статических web-сайтов виден из названия – они статичны, т. е. лишены движения и, в какой-то мере, скучны.

Преодолеть все эти недостатки статических web-сайтов можно, создавая так называемые «динамические» web-сайты. Динамический web-сайт в ответ на запросы web-браузера пользователя может не просто выдавать заранее подготовленные страницы HTML, а создавать страницы «динамически», на основе записей в базе данных. Такой подход к созданию сайта позволяет также отделить его содержимое от внешнего вида, т. е. появляется возможность быстро менять оформление сайта или даже предлагать пользователям несколько вариантов оформления на выбор.

Второй возможностью динамических web-сайтов является их «интерактивность», т. е. способность реагировать на действия пользователя. Например, сайт, торгующий экранами для ванн, может предлагать своим посетителям подобрать желательный декор и цвет экрана, выбрав их из предлагаемого ассортимента. Очень популярны игры, играть в которые можно, просто запустив браузер и посетив соответствующий web-сайт.

Технологии создания динамических web-сайтов делятся на две группы: серверные и клиентские. Под серверными технологиями подразумевают специальные программы, которые выполняются под руководством web-сервера и заняты обработкой запросов web-браузера. Чаще всего эти программы пишутся на специальных языках программирования, называемых языками сценариев. Самыми популярными языками сценариев являются PERL (Practical Extraction and Report Language – Практичный язык для создания выборок и отчетов) и PHP (расшифровывается, как самоповторяющаяся аббревиатура PHP: Hypertext Preprocessor – PHP: препроцессор гипертекста). Главное требование к языкам программирования динамических web-сайтов – это совместимость со стандартом CGI (Common Gateway Interface – общий шлюзовой интерфейс), который обеспечивает работу в «одной упряжке» web-браузера, web-сервера и программ, генерирующих содержимое сайта.

Еще одной достаточно популярной серверной технологией, разрабатываемой корпорацией Microsoft, является ASP (Active Server Pages – активные серверные страницы).

Серверные технологии создания сайтов не предъявляют никаких особенных требований к web-браузеру пользователя, браузер получает только итоговый результат обработки данных уже в конечном виде. Но на сервер ложится значительная вычислительная нагрузка, с которой он может просто не справиться при наплыве пользователей. А кроме того, если линии, соединяющие сервер и web-браузер пользователя, не слишком скоростные, в частности из-за низкой скорости соединения или большого расстояния между сервером и компьютером пользователя, время реакции на действия пользователя может быть достаточно большим, что далеко не всегда приемлемо. Решить эту проблему помогают «клиентские» технологии.

«Клиентом» в терминологии Всемирной паутины называют web-браузер конечного пользователя, а клиентскими технологиями, соответственно, технологии, с ним связанные. Основная задача этих технологий заключается в создании интерактивных элементов сайта, не требующих обмена данными с сервером. Есть две основные технологии этого типа: **JavaScript** и **Flash**. Технология **JavaScript** – это достаточно простой язык программирования, позволяющий манипулировать содержимым HTML-страниц, перемещать отдельные объекты по окну web-браузера и производить другие относительно несложные действия. Технология **Flash** – это система создания графических приложений, ориентированных на web. С ее помощью создаются красочные мультфильмы, музыкальные клипы, заставки, интерактивные сайты и полноценные игры. И все это можно сделать частью web-страницы. Один из недостатков **Flash** в том, что сложная **Flash**-анимация может полностью загрузить даже самый современный компьютер. Кроме того, такая графика требует большого количества скачиваемой информации, что может быть неприятно для пользователей, использующих тарифные планы оплаты Интернет-услуг с подсчетом скачиваемого объема информации, или при низкой скорости соединения – время загрузки таких сайтов может быть достаточно долгим.

Достоинства клиентских технологий очевидны – нет необходимости загружать web-сервер, но, как часто бывает, недостатки являются продолжениями достоинств. Дополнительные вычисления нагружают компьютер пользователя, особенно если он не слишком быстр, кроме того, сам браузер должен уметь работать с этими клиентскими технологиями. Причем, если с поддержкой **Flash** все обстоит просто, достаточно скачать модуль для работы с ним с сайта производителя этой системы (фирмы **Macromedia**), то при использовании **JavaScript** это не так. Браузер должен поддерживать соответствующие сценарии **JavaScript**, причем написанные в разных версиях языка, которые могут несколько отличаться друг от друга. Некоторые пользователи в целях безопасности сами отключают поддержку **JavaScript** в своем браузере.

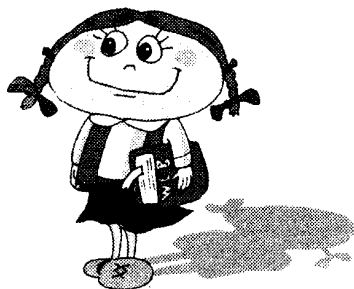
Серверные и клиентские технологии создания динамических сайтов не соперничают, а успешно дополняют друг друга при создании действительно современных динамических сайтов.

## Заключение

Вы узнали, как устроена Всемирная паутина и с помощью каких технологий создают web-сайты. Вероятно, после знакомства с таким богатством возможностей и количеством технологий вам покажется, что создавать web-сайты самостоятельно очень сложно, но на самом деле это не так. Разумеется, сразу вы не сможете сделать большой динамический сайт, но сделать небольшой статический сайт из нескольких страниц вам будет по силам и без особой подготовки. Как его сделать, вы узнаете из следующей главы. А набив руку в создании простых страниц, вы сможете перейти и к созданию более сложных сайтов и достаточно скоро вы сможете создавать по-настоящему профессиональные web-сайты. Продолжайте читать книгу!

## ГЛАВА 2.

# Первые web-страницы своими руками



Теперь, когда вы узнали, что такое Интернет, и познакомились с различными типами сайтов, пора узнать, как он выглядит «изнутри». В этой главе мы познакомимся с основами создания статических сайтов, с инструментами их разработки, и вы создадите свой первый статический сайт. Он будет небольшим, но вполне работоспособным.

## Файлы HTML «изнутри»

Любой статический сайт состоит из одного или нескольких файлов формата HTML, называемых еще «страницами». Как уже было рассказано в первой главе, HTML – это язык гипертекстовой разметки документов. Давайте сразу оговорим следующее: мы с вами будем следовать стандартам языка XHTML, версия которого 1.1 была одобрена в качестве Рекомендации Консорциума Всемирной паутины (W3C) 31 мая 2001 года. Однако, поскольку лишь немногие разработчики строго следуют его правилам, которые отличаются большей строгостью, чем стандарты HTML, а файлы XHTML также имеют расширение **html**, мы оставим общепринятое название языка разметки – HTML.

Основой языка HTML являются теги, специальные команды, заключаемые в угловые скобки, например **<body>**, **<img src=>** или **<p>**. С помощью тегов описывается структура документа, его внешний вид и ссылки на другие документы. Код простого XHTML-документа приведен в Листинге 2.1.

*Листинг 2.1. Простейший XHTML-документ*

---

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>
<html>
  <head>
    <meta http-equiv=»Content-Type» content=»text/html; charset=utf-
8» />
    <title>Это простейший документ HTML</title>
  </head>
  <body>
    <p>Содержимое документа HTML</p>
  </body>
</html>
```

---



Если вы откроете документ с этим кодом в web-браузере, то увидите результат, показанный на Рис. 2.1.

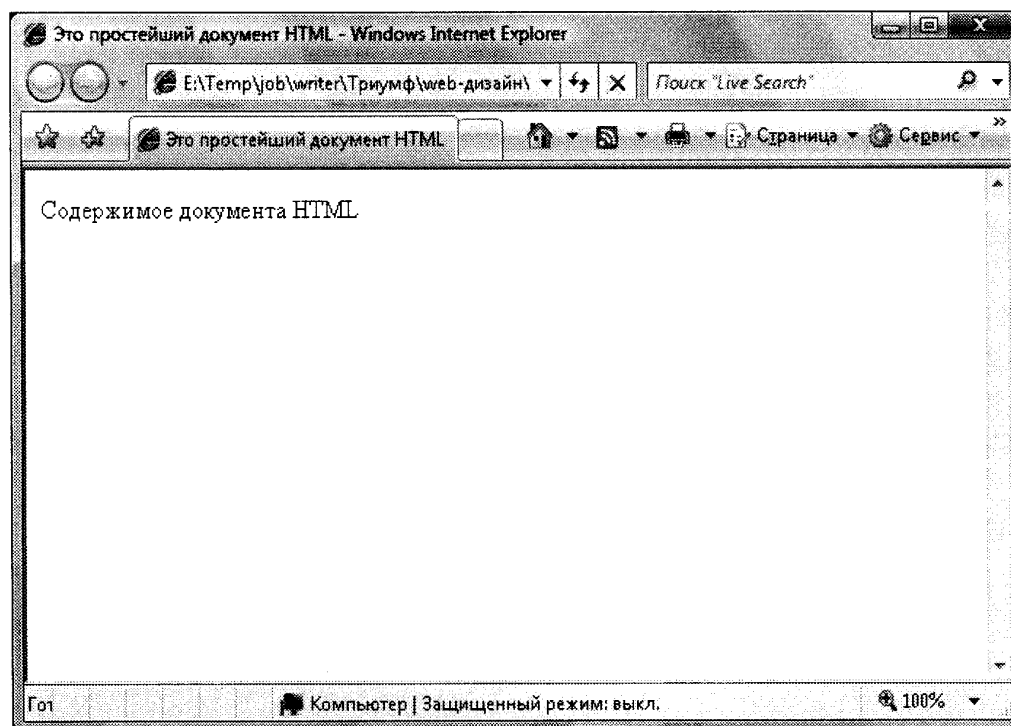


Рис. 2.1. Простейший документ HTML

Есть два основных способа создания HTML-документов. Первый – написание кода HTML вручную. Для этого вы должны изучить все основные теги HTML, хорошо знать, какое действие выполняет каждый из них, и хорошо представлять, как будет выглядеть итоговый результат после того или иного изменения кода. С одной стороны, такое «прямое» кодирование достаточно трудоемко и требует постоянно держать в памяти массу информации, но с другой – позволяет полностью контролировать все тонкости HTML-кода и в некоторых случаях добиваться результата, недостижимого другими способами. Еще одним плюсом такого способа разработки является то, что писать код HTML вы можете абсолютно в любом текстовом редакторе, например в Блокноте Windows. Хотя и существуют различные программы, облегчающие «ручное» кодирование HTML.

Другой способ – «визуальное» составление страниц в специальном HTML-редакторе. Создавать HTML-документы таким способом можно, просто вставляя на страницу нужные элементы оформления, ссылки на другие документы, изображения, тексты и размещая их так, как необходимо. Такой способ работы часто называют WYSISWYG (What You See is What You Get – что вы видите, то и получаете). Такая методика составления HTML-документов проста в освоении, позволяет быстро создавать достаточно сложные гипертекстовые документы и, кроме того, наглядна. Но недостатком такого способа работы является то, что HTML-редактор, как правило, не позволяет реализовать все тонкости языка HTML. Поэтому визуальные HTML-редакторы предусматривают и работу непосредственно с кодом страницы.

Одним из самых популярных визуальных HTML-редакторов является программа SharePoint Designer 2007, и в большей части этой книги рассматривается создание web-сайтов именно с помощью этой программы.

## Установка программы SharePoint Designer 2007

Начнем с установки программы SharePoint Designer 2007 на ваш компьютер.

Чтобы установить эту программу:

- ▶ Вставьте диск с программой SharePoint Designer 2007 в **DVD**-привод компьютера. Запустится программа автозапуска диска.

В зависимости от поставки способы запуска системы установки SharePoint Designer 2007 могут быть различны.

- ▶ Руководствуясь подсказками системы автозапуска, запустите систему установки программы SharePoint Designer 2007.

Вы должны увидеть диалог, сообщающий, что идет подготовка к установке программы. В следующем диалоге программа установки предложит ввести 25-значный ключ продукта (Рис. 2.2). Вы можете сделать это сразу или ввести позднее.

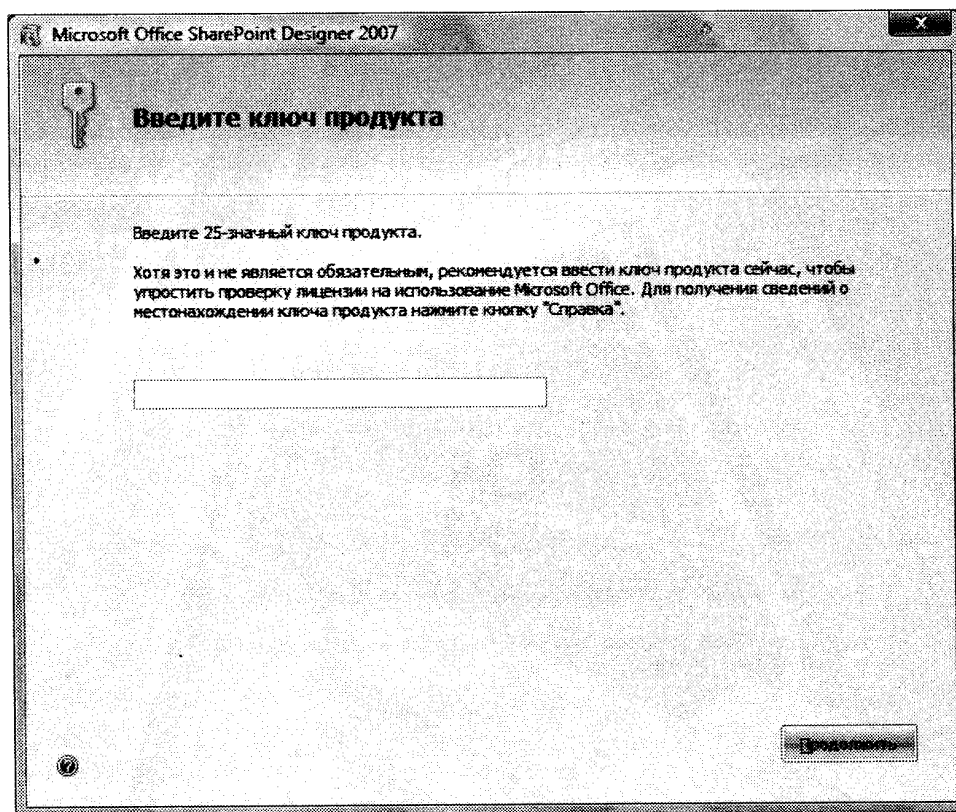


Рис. 2.2. Запрос ключа продукта

- ▶ Для продолжения установки щелкните мышью на кнопке **Продолжить** (Continue). На экране появится диалог выбора типа установки программы (Рис. 2.3).
- ▶ Щелкните мышью на кнопке **Установить** (Install), чтобы приступить к установке SharePoint Designer 2007 с наиболее часто используемыми компонентами. На экране отобразится диалог хода установки программы, в графическом виде отображающий процесс установки SharePoint Designer 2007.

Когда будут установлены все необходимые файлы и произведены требуемые настройки, появится диалог завершения установки программы (Рис. 2.4).

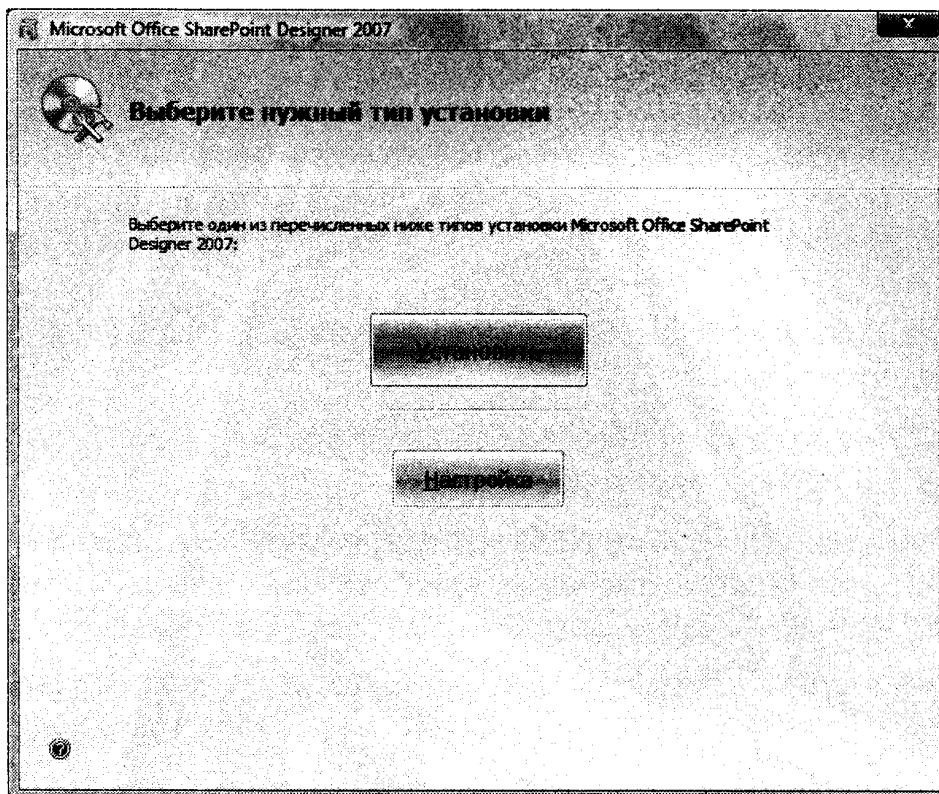


Рис. 2.3. Диалог выбора типа установки

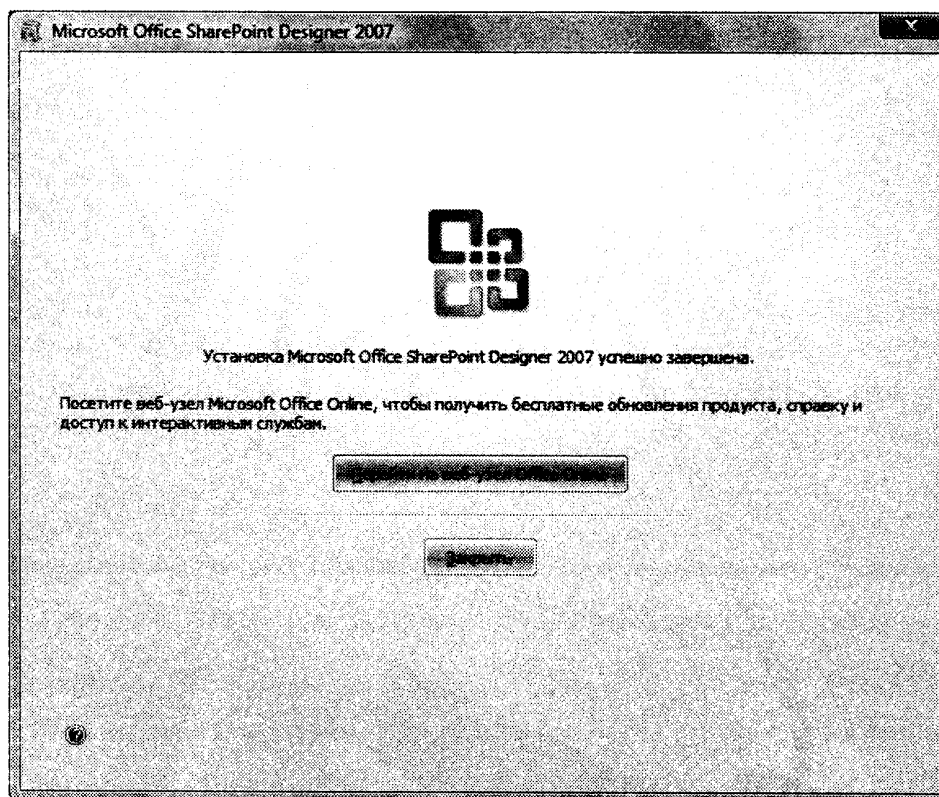


Рис. 2.4. Диалог завершения установки программы

- Щелкните мышью на кнопке **Готово** (Finish), чтобы завершить процесс установки программы.

Теперь программа SharePoint Designer 2007 успешно установлена.

## Создание первой web-страницы

Сейчас у вас есть все необходимые инструменты для создания вашей первой web-страницы. Для начала запустим программу SharePoint Designer 2007.

- Нажмите кнопку **Пуск (Start)** на **Панели задач (Taskbar)** операционной системы Windows. Откроется основное меню.
- Выберите команду **Все Программы ♦ Microsoft Office ♦ Microsoft Office SharePoint Designer 2007** (Programs ♦ Microsoft Office ♦ Microsoft Office SharePoint Designer 2007) из основного меню.

Программа будет запущена, вы увидите ее основное рабочее окно (Рис. 2.5). В центральной его части вы можете видеть HTML-код, подобный приведенному нами в Листинге 2.1. Этот код является необходимым для создания любого документа HTML. Окно программы напоминает рабочее окно текстового редактора. И это сходство не случайно. Создание web-страниц в программе SharePoint Designer 2007 во многом похоже на создание новых документов в текстовом редакторе, таком как Microsoft Word. Если вы знакомы с текстовыми редакторами, – это вам существенно поможет в освоении программы.

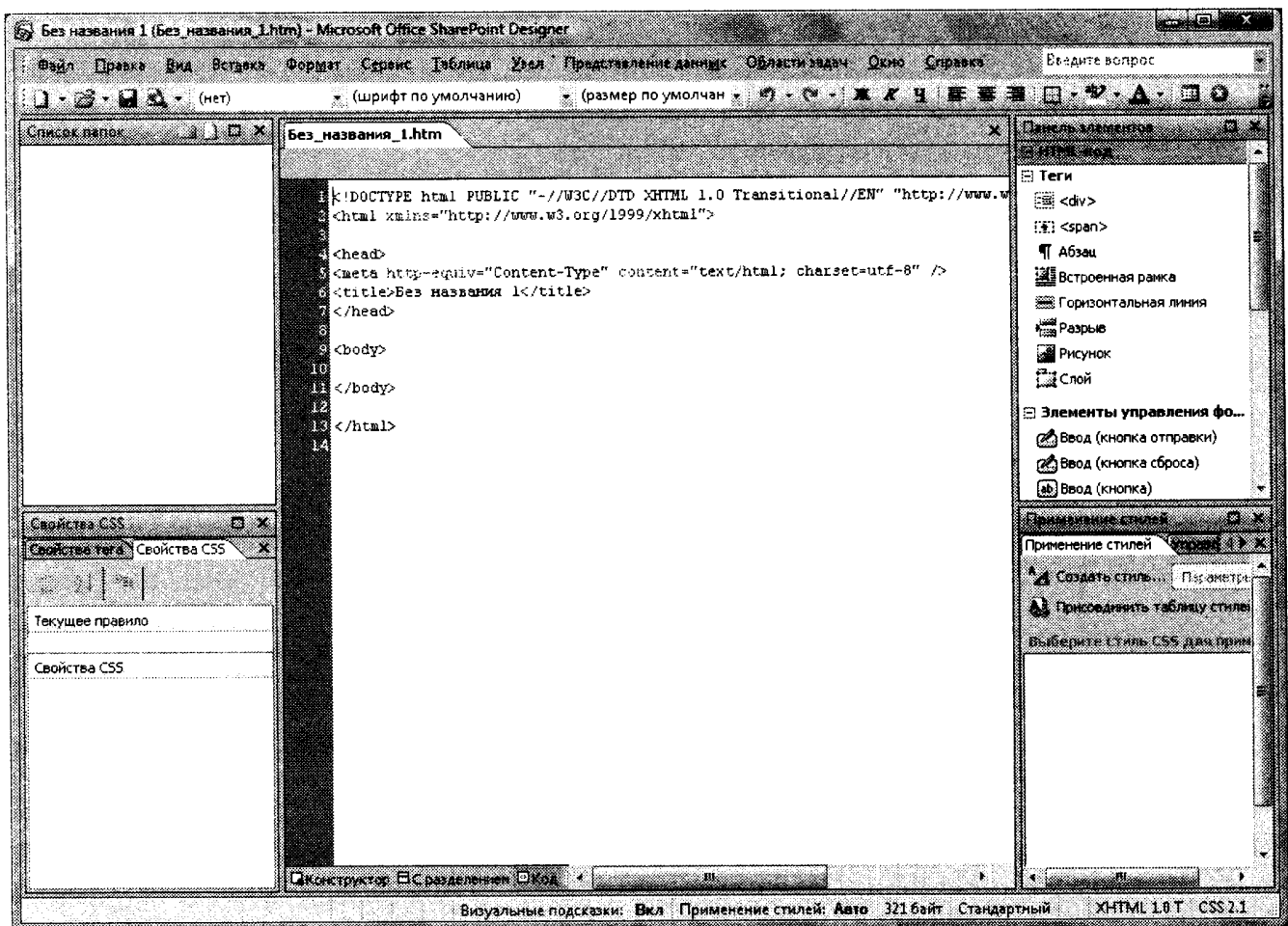


Рис. 2.5. Рабочее окно программы SharePoint Designer 2007

Давайте создадим вашу первую web-страницу.

## Создание web-страницы

Приступить к работе над web-страницей можно сразу после запуска программы. В дальнейшем, если вы захотите открыть новый документ для создания документа HTML, необходимо выполнить следующие действия:

- Выберите пункт меню **Файл ♦ Создать ♦ Страница** (File ♦ New ♦ Page). На экране отобразится диалог **Создание** (Create) (Рис. 2.6).

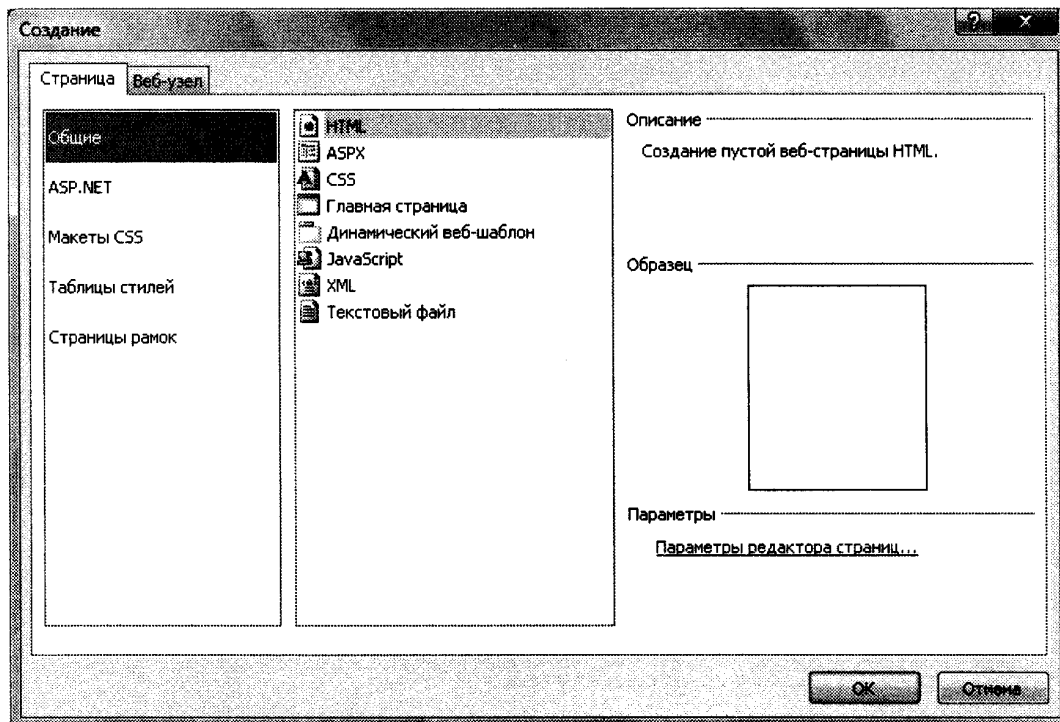


Рис. 2.6. Диалог **Создание** (Create)

- Убедитесь, что в левом списке диалога выбран пункт **Общие** (General), а в правом – **HTML**, и щелкните мышью на кнопке **ОК**. Программа **SharePoint Designer 2007** откроет для редактирования новый HTML-документ.

Вся центральная часть рабочего окна **SharePoint Designer 2007** служит для редактирования содержимого создаваемой страницы.

Непосредственно над рабочей областью программы располагается область заголовков документов. В ней отображаются названия всех документов, открытых для редактирования, а также некоторых вспомогательных модулей программы **SharePoint Designer 2007**. Новый документ, открытый при запуске программы, пока вы не дали ему название, озаглавлен как **Без\_названия\_1.htm**.

Ниже рабочей области расположены переключатели режимов работы редактора. Они служат для переключения между макетом страницы и HTML-кодом. Также возможна совместная работа этих режимов с разделением экрана.

Выше области заголовков расположены панели инструментов, внешний вид и назначение которых схожи с панелями инструментов многих текстовых редакторов. В правой верхней области программы представлены наиболее часто употребляемые

теги HTML. Для размещения нужного вам тега на странице просто перетащите его мышью. О назначении остальных областей рабочего окна вы узнаете ниже.

- Перейдите в режим конструктора, щелкнув мышью на переключателе **Конструктор** (Builder) ниже рабочей области окна программы.
- Щелкните мышью в области макета страницы и введите с клавиатуры текст: «Эта моя первая Web-страница».

Чтобы созданный вами документ везде корректно определялся как написанный на русском языке, а также для удобства просмотра исходного кода страницы, что понадобится нам чуть позже, сделайте следующее:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties) (Рис. 2.7).

- В текстовом поле ввода **Название** (Title) введите «Моя Web-страница». Это будет заголовком вашей страницы.


- Щелкните мышью на вкладке **Язык** (Language), откроется диалог настройки языка страницы (Рис. 2.8).

- В открывающемся списке **Пометить текущий документ, указав:** (Mark current document as) группы настроек **Язык страницы** (Page Language) выберите пункт **Русский** (Russian).

- В открывающемся списке **Сохранить документ, используя** (Save document using) группы настроек **Набор знаков** (Character set) выберите пункт **кириллица** (Cyrillic).

- Щелкните мышью на кнопке **ОК**, чтобы сохранить изменения и закрыть диалог настройки.

Сохраните полученную web-страницу:

- Щелкните мышью на кнопке  панели инструментов программы. Откроется диалог **Сохранить как** (Save as) (Рис. 2.9).
- Перейдите к папке, в которую вы хотите сохранить полученный файл.
- Щелкните мышью в поле ввода **Имя файла** (Filename) и введите название файла **index.html**.
- После этого щелкните мышью на кнопке **Сохранить** (Save), чтобы произвести сохранение и закрыть диалог.

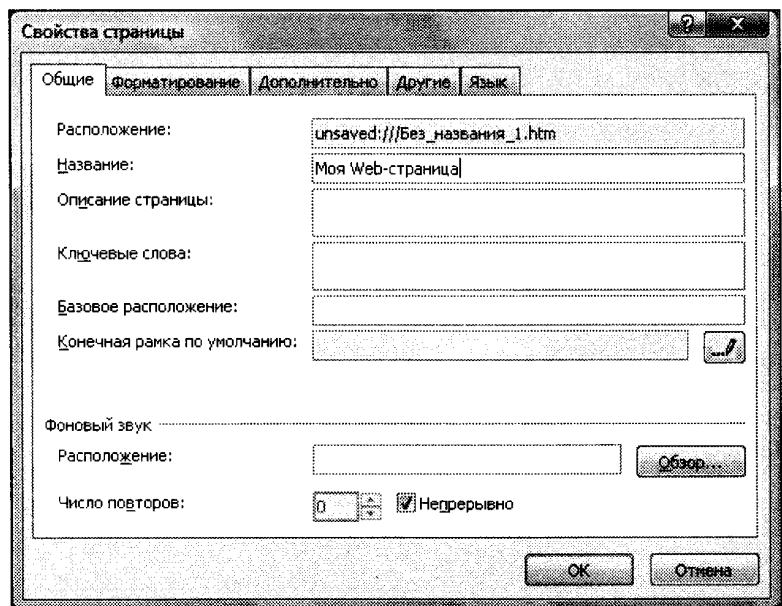


Рис. 2.7. Диалог **Свойства страницы** (Page properties)

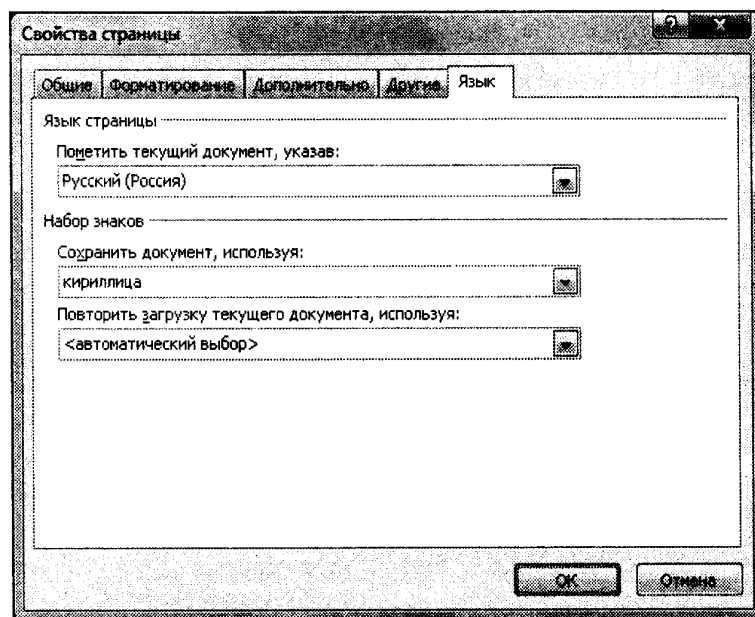


Рис. 2.8. Вкладка **Язык** (*Language*) диалога **Свойства страницы** (*Page properties*)

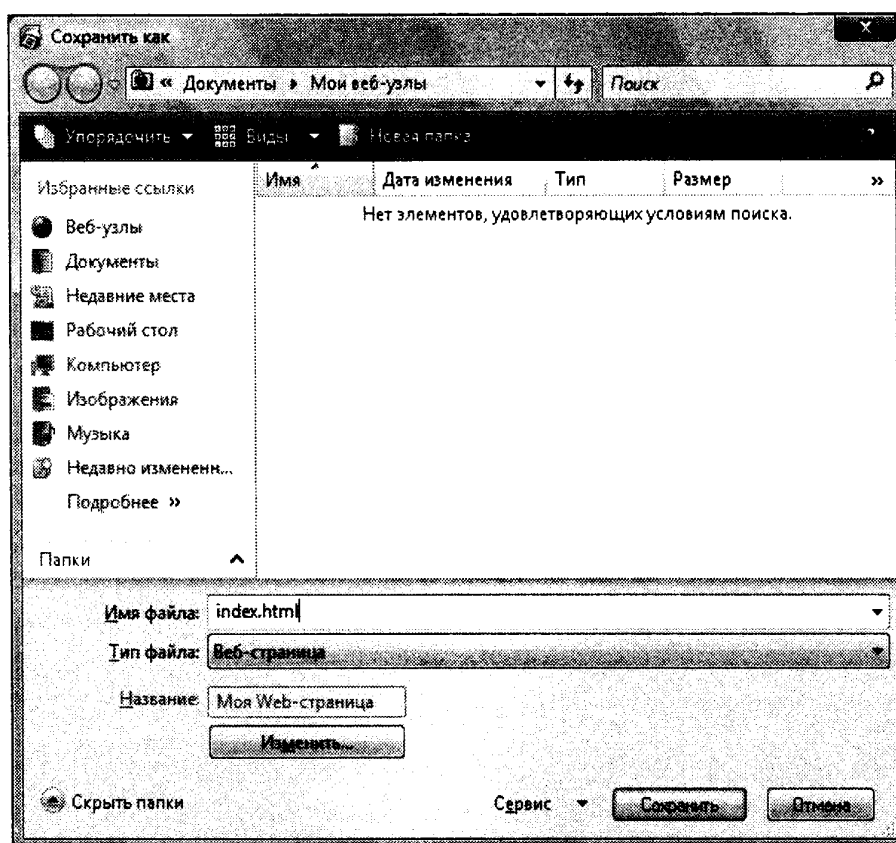



Рис. 2.9. Диалог **Сохранить как** (*Save as*)

В дальнейшем, чтобы повторно сохранить web-страницу, просто щелкайте мышью на кнопке  на панели инструментов программы, сохранение будет произведено без лишних вопросов. Рекомендуем вам проделывать эту процедуру почаще.

Вы можете сохранить документ под другим именем, например, чтобы создать еще одну web-страницу на базе уже существующей.

- Выберите команду меню **Файл ♦ Сохранить как** (File ♦ Save as). Откроется диалог сохранения файла (Рис. 2.9).
- Перейдите к папке, в которую вы хотите сохранить файл.
- Щелкните мышью в поле ввода **Имя файла** (Filename) и введите новое название файла.
- После этого щелкните мышью на кнопке **Сохранить** (Save), чтобы произвести сохранение и закрыть диалог.

Закончив работу с документом, вы можете закрыть его. Для этого:

- Щелкните мышью на кнопке закрытия документа, в области заголовков документов.

Если документ перед закрытием не был сохранен, появится диалог запроса о сохранении документа.

- Щелкните мышью на кнопке **Да** (Yes), чтобы сохранить изменения в документе и закрыть его.
- Щелкните мышью на кнопке **Нет** (No), чтобы закрыть документ, не сохраняя внесенные в него изменения.
- Щелкните мышью на кнопке **Отмена** (Cancel), чтобы отменить закрытие документа и вернуться к его редактированию.

Чтобы снова открыть для редактирования закрытый документ:

- Выберите команду меню **Файл ♦ Открыть** (File ♦ Open). На экране появится диалог **Открытие файла** (Open file), Рис. 2.10.

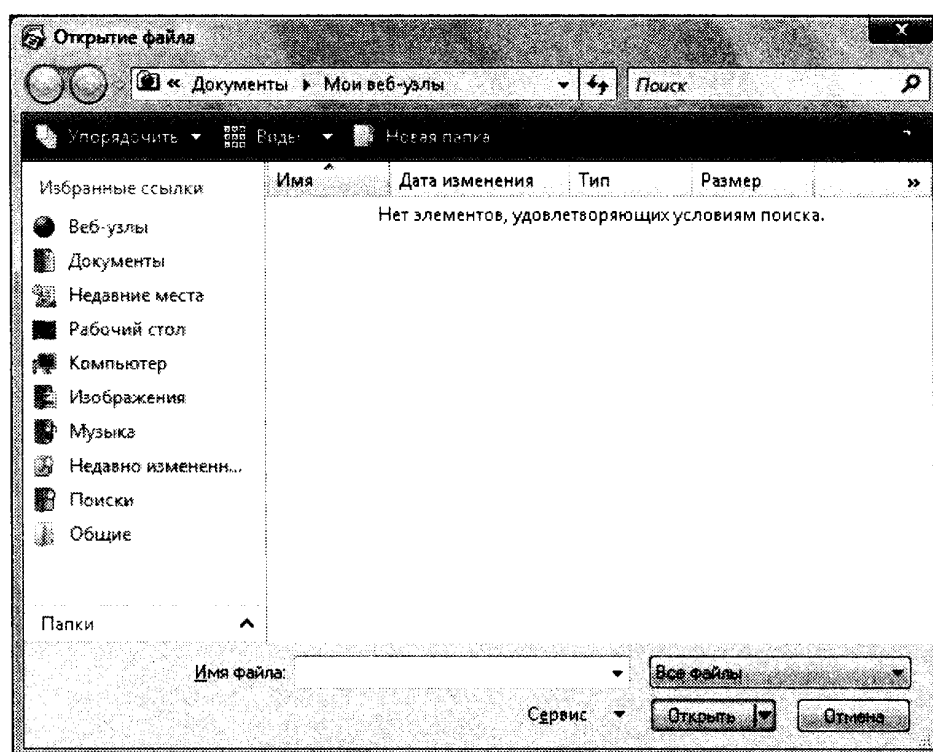


Рис. 2.10. Диалог **Открытие файла** (Open file)



- Перейдите к папке, в которой находится нужный вам файл, и щелкните мышью на его названии. Файл выделится цветом.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы открыть файл для редактирования и закрыть диалог.

Итак, вы создали свою самую первую web-страницу. Теперь можно посмотреть полученный результат.

## Просмотр созданных web-страниц в браузере

Программа SharePoint Designer 2007 предусматривает удобный способ открытия web-страниц в браузере. По умолчанию используется Internet Explorer, однако вы можете добавить любой браузер, установленный на вашем компьютере.

- Выберите команду меню **Файл** ♦ **Просмотреть в обозревателе** ♦ **Microsoft Internet Explorer 7.0** (File ♦ Preview in web-browser ♦ Microsoft Internet Explorer) или нажмите клавишу **F12**. Текущая HTML-страница откроется в web-браузере Internet Explorer (Рис. 2.11).

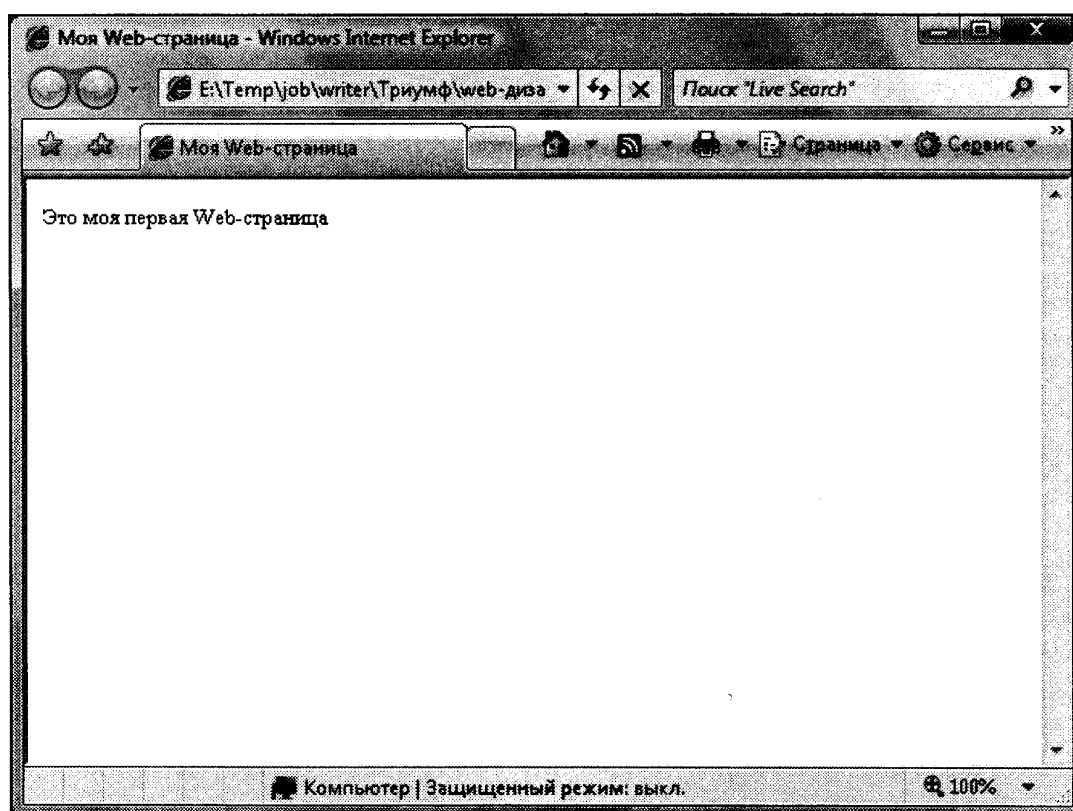


Рис. 2.11. Просмотр результатов работы в браузере Internet Explorer

## Основы языка HTML

Ваша первая web-страница создана. Но внешний вид ее пока ничем не отличается от обычного текстового файла, созданного, например, в Блокноте Windows. Давайте разберемся, какие возможности по оформлению текстов предоставляет HTML.

## Базовые понятия языка HTML

Как мы уже говорили выше, несмотря на то, что создавать web-страницы средствами редактора WYSIWYG несравненно быстрее и проще, чем напрямую вводить код HTML, но, тем не менее, знание азов HTML необходимо для разработки серьезных web-проектов. Поэтому сначала мы вкратце рассмотрим основы языка HTML, а в дальнейшем, рассказывая об оформлении web-страниц в программе SharePoint Designer 2007, будем приводить теги, отвечающие за тот или иной элемент оформления в коде страницы.

Как уже было сказано ранее, основу языка гипертекстовой разметки HTML составляют теги – текстовые команды, заключенные в угловые скобки. Теги бывают одиночными и парными. Одиночные теги, как и следует из их названия, используются поодиночке, обозначая самостоятельный элемент оформления. Например, тег **<br>** означает разрыв строки, а с помощью тега **<img>** в HTML-документ можно вставить изображение. В соответствии с правилами XHTML в тегах следует использовать строчные буквы, например, **<title>**.

Парные теги образуются двумя тегами, открывающим и закрывающим, которые отличаются друг от друга только значком «/» перед закрывающим тегом. Парные теги влияют на оформление всего кода HTML, находящегося между ними. Парные теги еще называют «контейнером», а код, находящийся между ними, «содержимым контейнера». Примером использования контейнера будет следующий код: **<b>Этот текст будет полужирным </b>**. Весь текст, помещающийся в контейнер **<b></b>** оформляется полужирным шрифтом и в web-браузере будет выглядеть примерно так: «**Этот текст будет полужирным**». Контейнеры могут вкладываться друг в друга, действуя вместе на содержимое вложенного контейнера. Например, если вложить в контейнер **<b></b>** другой контейнер, **<i></i>**, выделяющий текст внутри себя курсивом, примерно так: **<b>Этот текст будет полужирным <i>А этот еще и курсивным</i></b>**, то результат будет следующим: «**Этот текст будет полужирным. А этот еще и курсивным**».

Дополнительные параметры действия тегов задаются атрибутами. Атрибуты записываются вместе с названием тега в угловых скобках. Например, так: **<hr align=»center»>**. В этом примере атрибут **align** – тип выравнивания горизонтальной линии. В данном случае – по центру.

В качестве иллюстрации структуры языка HTML мы используем только что созданную вами web-страницу.

- Вернитесь к режиму просмотра кода web-страницы, щелкнув мышью на кнопке **Код** (Code) переключателя режимов работы редактора (Рис. 2.12).

Макет страницы в рабочей области программы SharePoint Designer 2007 заменен на его HTML-код. При желании можно создавать web-страницы непосредственно в этом режиме, напрямую вводя весь код.

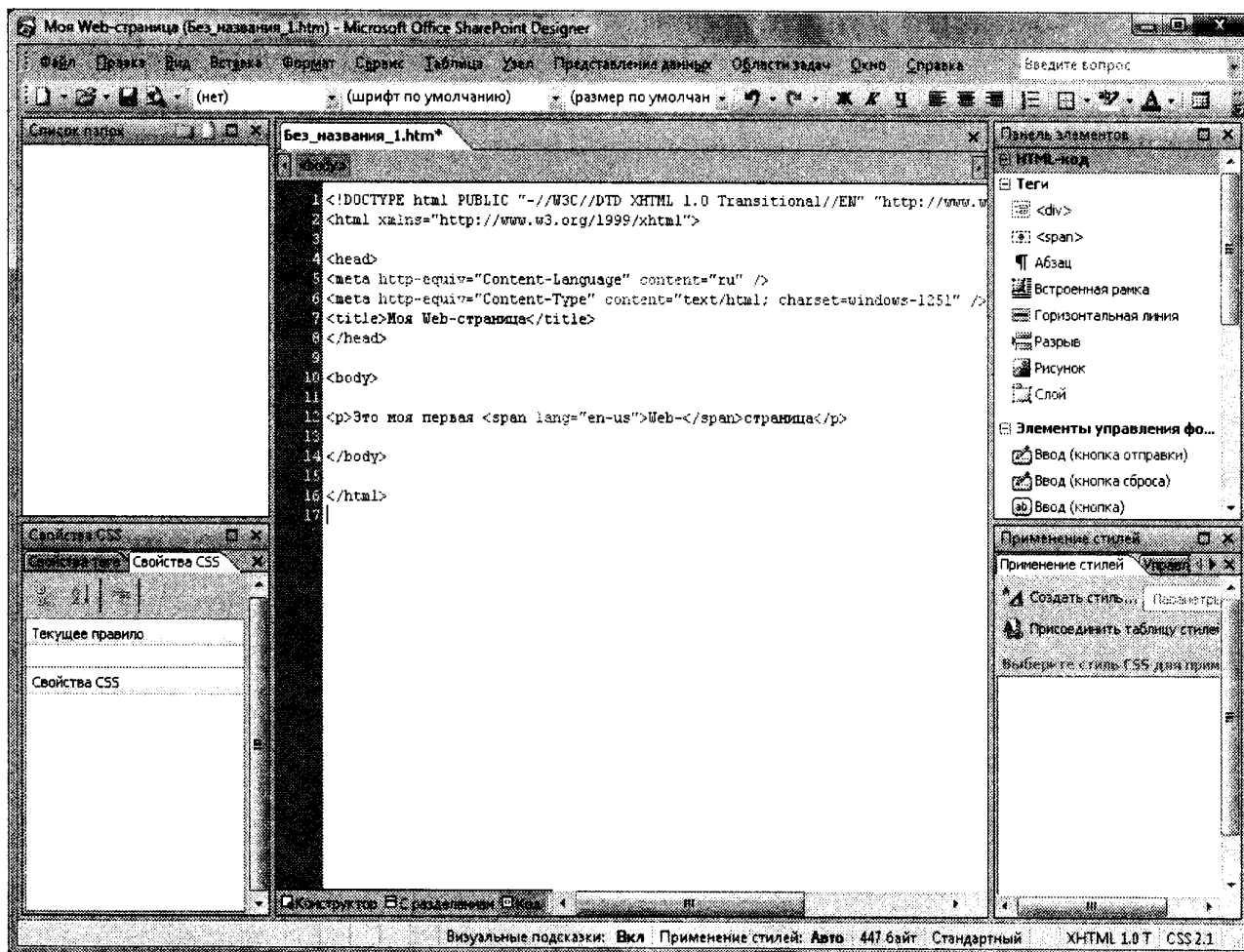


Рис. 2.12. Программа SharePoint Designer 2007  
в режиме работы с кодом страницы

Для понимания принципов работы HTML удобнее будет одновременно видеть как код страницы, так и ее макет, чтобы можно было заметить, какие теги как влияют на оформление документа, и наоборот – как изменение документа влияет на код страницы.

- ▶ Щелкните мышью на кнопке **С разделением** (Divide) переключателя режимов работы редактора.

Рабочая область SharePoint Designer 2007 разделится на две части, в нижней будет макет страницы, а в верхней части будет размещаться HTML-код вашего документа (Рис. 2.13).

Посмотрите внимательно на код созданной вами web-страницы (Листинг 2.2.). Для удобства понимания структуры HTML в листинге разные уровни вложенности контейнеров отмечаются отступами.

В языке HTML есть несколько тегов, которые отвечают не за оформление документа, а за его структуру. Это теги **<html>**, **<head>** и **<body>**. Весь документ HTML заключается в один большой контейнер **<html></html>**. Эта пара тегов сообщает web-браузеру о том, что документ написан на языке HTML, а не на каком-либо другом языке разметки. Этот контейнер в свою очередь содержит еще два контейнера. Первый,

**<head></head>**, называется заголовком, и в нем содержится разнообразная информация о содержимом всего документа: язык, на котором написаны тексты, версия языка HTML, на которую ориентировались создатели документа, основные ключевые слова для поисковых «ботов», общий заголовок HTML-документа и прочее. Второй контейнер, **<body></body>** включает в себя все основное содержимое документа, его «тело».

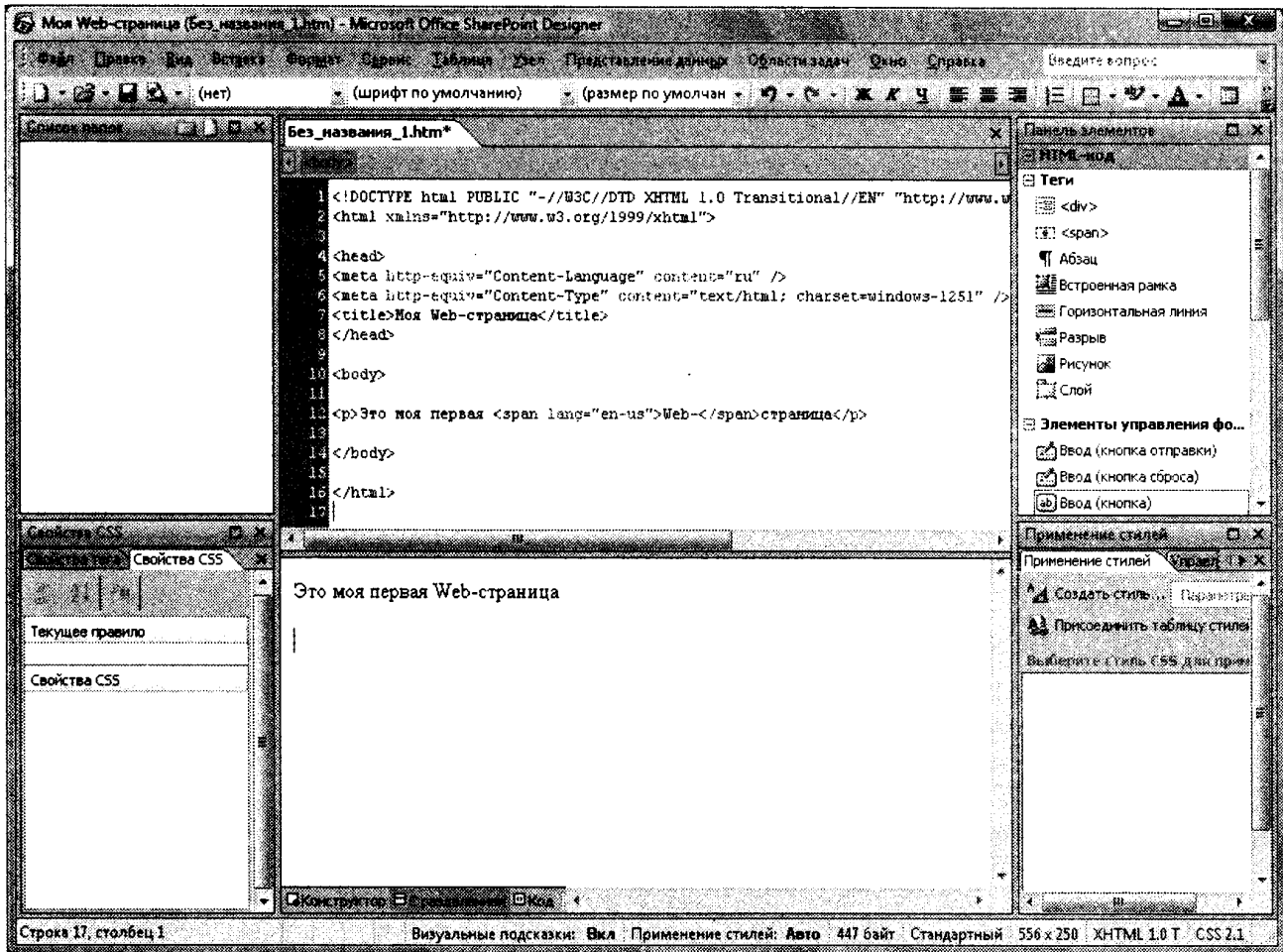


Рис. 2.13. Одновременная работа с макетом страницы и HTML-кодом в программе SharePoint Designer 2007

### Листинг 2.2. Код вашей первой web-страницы

```

<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
  <meta http-equiv=»Content-Language» content=»ru» />
  <meta http-equiv=»Content-Type» content=»text/html;
  charset=windows-1251» />
  <title>Моя Web-страница</title>
</head>

```

```
<body>
  <p>Это моя первая <span lang=»en-us»>Web-</span>страница</p>
</body>
</html>
```

Все остальные теги размещаются внутри одного из этих двух контейнеров.

## Основные элементы оформления HTML-страниц

В программе SharePoint Designer 2007 вы можете добавлять, удалять, редактировать тексты, изменять оформление отдельных слов и предложений, вставлять в него заголовки, списки, таблицы, так же, как и в любом текстовом редакторе. Познакомимся теперь с основными элементами оформления HTML-страниц, тегами, за это отвечающими, и способами работы с этими элементами в программе SharePoint Designer 2007.



### Название web-страницы

При просмотре web-сайтов в заголовке браузера отображается название текущей страницы. За название web-страницы отвечает парный тег **<title>**, размещаемый в заголовке HTML-документа. Название страницы размещается между открывающим и закрывающим тегами **<title>**. Например, так: **<title>Это заголовок Web-страницы</title>**. Это единственный тег в заголовочной части HTML-документа, действие которого отображается непосредственно в браузере. Помимо редактирования кода, заголовок web-страницы можно изменить, как мы это уже делали при помощи диалога **Свойства страницы** (Page properties)





Мы рекомендуем вам, чтобы лучше познакомиться с языком HTML, после каждого изменения web-страницы смотреть на изменения, которые произошли с ее кодом, и самостоятельно пробовать вносить изменения в HTML-теги.

### Форматирование абзацев текста

Удалим все содержимое текущей web-страницы.

- Переключитесь в режим макета страницы, щелкнув мышью на переключателе **Конструктор** (Builder).
- Щелкните мышью в рабочей области.
- Выберите пункт меню **Правка ♦ Выделить все** (Edit ♦ Select All).
- Нажмите клавишу . Все содержимое web-страницы будет удалено.
- Чтобы начать новый абзац текста, просто нажмите клавишу . В языке HTML абзац заключается в контейнер парным тегом **<p>**.

Содержимое абзаца может быть выровнено относительно окна просмотра (Рис. 2.14). В HTML выравнивание абзаца задается атрибутом **align** тега **<p>**. Он может прини-

мать четыре значения: **left** – выравнивание по левому краю окна, **right** – по правому, **center** – по центру и **justify** – выравнивание текста по ширине окна. Задать выравнивание текущего абзаца в программе **SharePoint Designer 2007** можно, щелкнув мышью на соответствующей кнопке в панели инструментов. Щелкнув мышью на кнопке , вы выровняете текст абзаца по левому краю окна. Щелкнув на кнопке  – по правому. Кнопка  отвечает за выравнивание текста по центру, а  – за выравнивание по ширине текста. Выравнивание по ширине доступно на панели инструментов **Форматирование (Format)**, отобразить которую можно, щелкнув правой кнопкой мыши на панели инструментов и выбрав в открывшемся меню одноименный пункт.

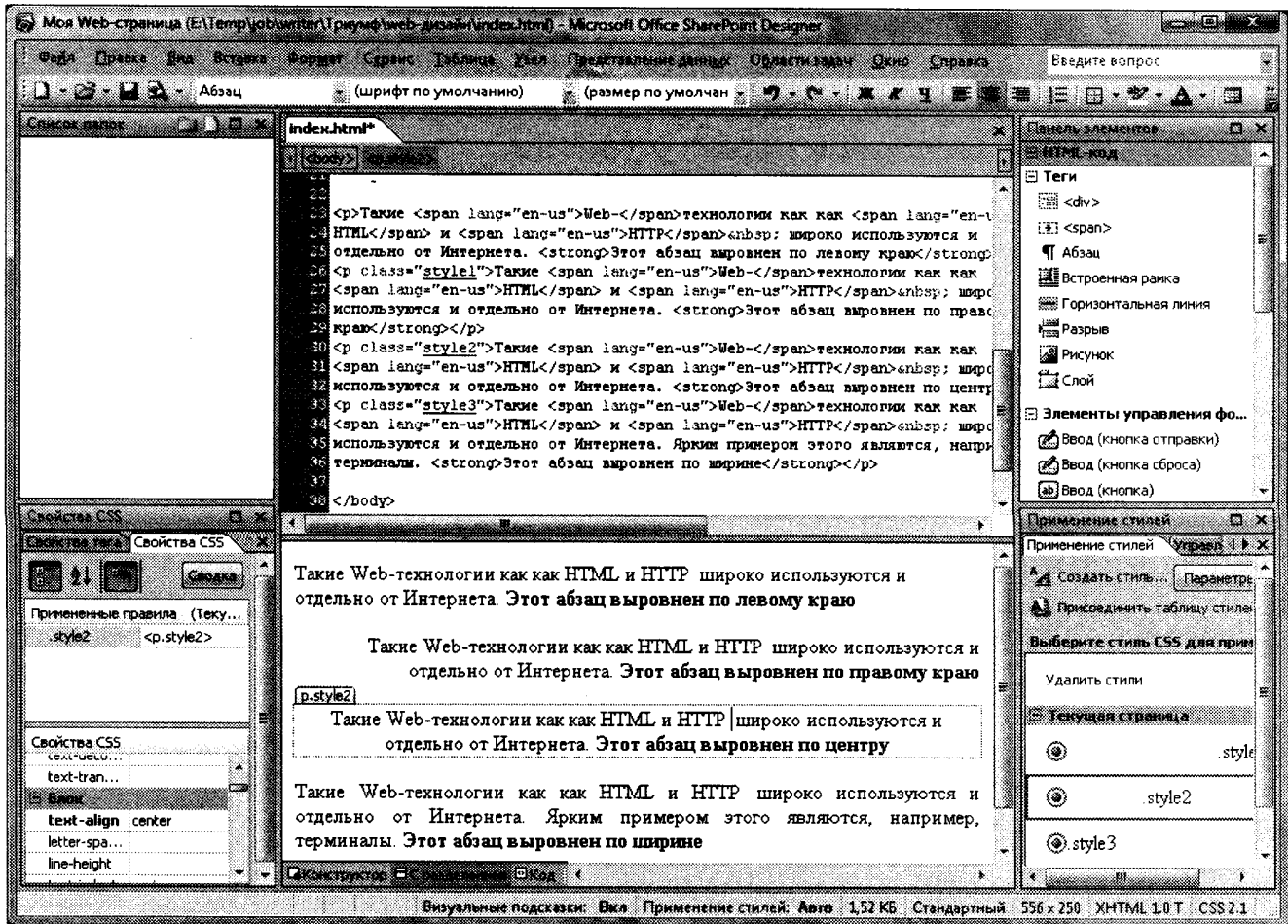


Рис. 2.14. Выравнивание абзацев текста в HTML

## Заголовки

Основой структурирования текста и разбиения его на отдельные смысловые фрагменты в HTML являются заголовки (Рис. 2.15). Язык HTML поддерживает шесть уровней заголовков. Они создаются с помощью контейнеров `<h1></h1>`, `<h2></h2>` и до `<h6></h6>`, где цифра, стоящая после буквы «Н», означает уровень заголовка.

Заголовки первого уровня обычно содержат название всего документа. Заголовки второго уровня – названия основных разделов, третьего – названия подразделов и т. д.

Правильно расставленные заголовки позволяют проще ориентироваться в документе и легко находить нужную информацию.

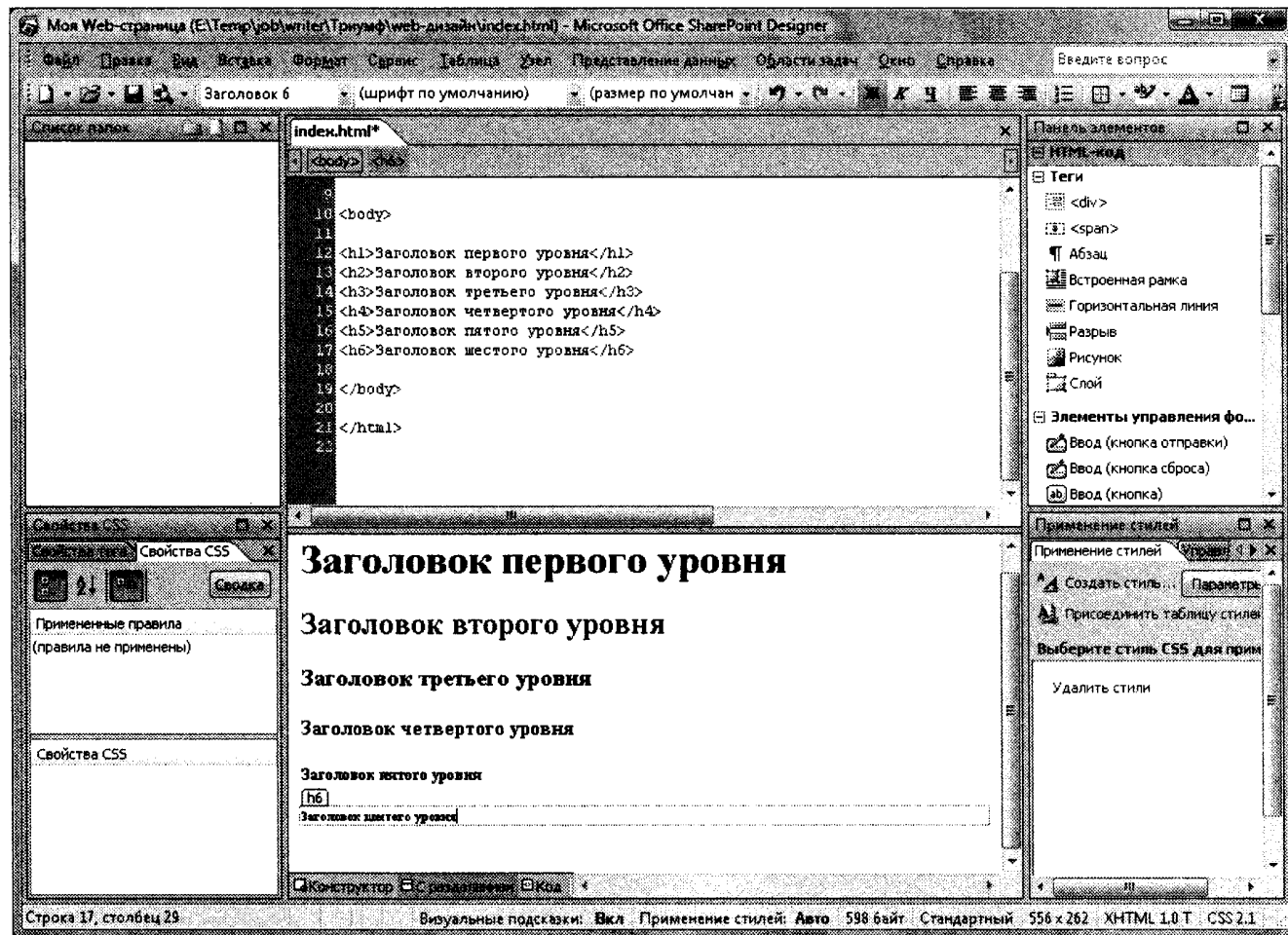






Рис. 2.15. Заголовки в HTML

Удалим все содержимое текущей web-страницы:

- Щелкните мышью в области макета страницы.
- Выберите пункт меню **Правка** ♦ **Выделить все** (Edit ♦ Select All).
- Нажмите клавишу **Delete**. Все содержимое web-страницы будет удалено.

Чтобы создать заголовок в программе SharePoint Designer 2007:

- Щелкните мышью на открывающемся меню **Стиль** (Style) – **Обычный**, панели инструментов.
- В открывшемся меню щелкните мышью на одной из строк, подписанных **Заголовок 1**, ... **Заголовок 6** (Header 1 ... Header 6), в зависимости от того, заголовок какого уровня вы хотите создать.
- Введите текст заголовка с клавиатуры.
- Чтобы закончить ввод заголовка, нажмите клавишу **Enter**. Вы начнете новый абзац обычного текста.

Как и обычные абзацы, заголовки можно выравнивать относительно окна просмотра кнопками , ,  и .

## Горизонтальные линии

Еще одним способом разделения HTML-документа на отдельные части является разбиение его горизонтальными линиями. Они позволяют четко отделять разделы страницы друг от друга, визуально отсекая один раздел от другого.

Горизонтальная линия задается одиночным тегом `<hr>` (Рис. 2.16).

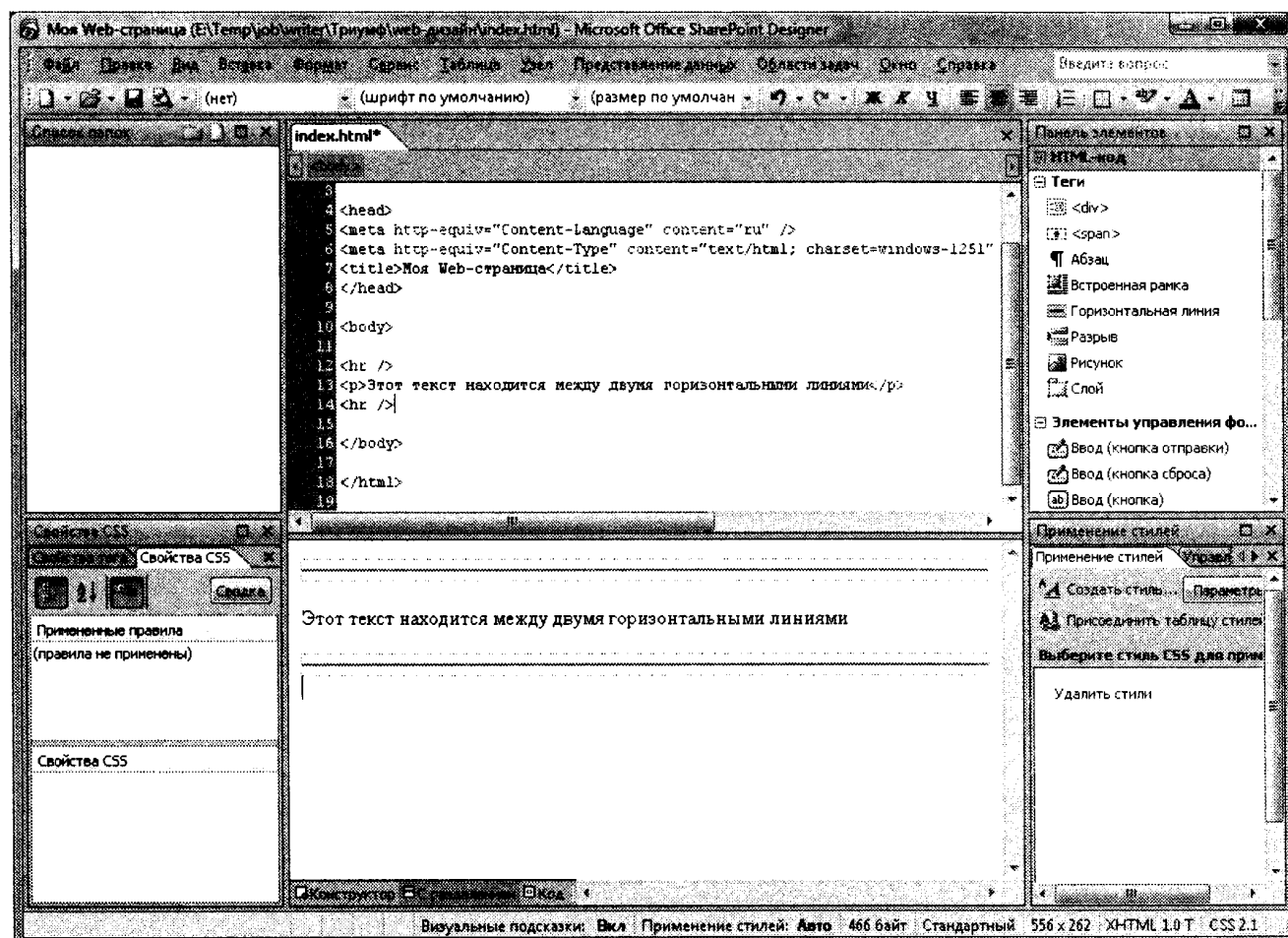


Рис. 2.16. Горизонтальные линии в HTML

Удалим все содержимое текущей web-страницы:


- Щелкните мышью в области макета страницы.
- Выберите пункт меню **Правка** ♦ **Выделить все** (Edit ♦ Select All).
- Нажмите клавишу **Delete**. Все содержимое web-страницы будет удалено.

Чтобы в программе SharePoint Designer 2007 вставить на web-страницу горизонтальную линию:

- Из расположенной в правой верхней части окна области **HTML-код** (HTML-code) перетащите мышью элемент **Горизонтальная линия** (Horizontal ray) в нужное вам место. Линия будет вставлена.

Чтобы удалить горизонтальную линию со страницы:



- Щелкните мышью на линии, которую хотите удалить. Линия будет выделена рамкой черного цвета.
- Нажмите на клавишу . Линия будет удалена.


## Списки


Одним из часто используемых способов упорядочивания текста являются списки. Они бывают нумерованными, маркированными, в виде меню и в виде списка определений.

Для оформления ряда похожих элементов лучше всего подходят маркированные списки. Каждый элемент такого списка выделяется специальным значком, маркером.



Нумерованный список задается контейнером `<ul></ul>`. Каждый отдельный элемент списка содержится в контейнере `<li></li>`.

Чтобы создать маркированный список:

- Щелкните мышью на кнопке панели инструментов **Маркеры (Markers)** .

Можете приступить к вводу содержимого списка. Каждый новый элемент добавляется нажатием клавиши .

Чтобы закончить создание списка:


- Мышью или, нажав клавишу , переместите текстовый курсор вниз.
- Дважды нажмите клавишу .


Для оформления различных рецептов, пошаговых инструкций и различных перечислений, в которых порядок элементов имеет значение, целесообразно использовать нумерованные списки (Рис. 2.17).

Нумерованный список – это такой список, элементы которого маркируются числами либо буквами в порядке возрастания.



Нумерованный список задается контейнером `<ol></ol>`. Каждый отдельный элемент списка содержится в контейнере `<li></li>`.

Чтобы создать нумерованный список:

- Щелкните мышью на кнопке панели инструментов **Нумерация (Numeration)** .

Можете приступить к вводу содержимого списка. Каждый новый элемент добавляется нажатием клавиши .

Чтобы закончить создание списка:

- Мышью или, нажав клавишу , переместите текстовый курсор вниз.
- Дважды нажмите клавишу .

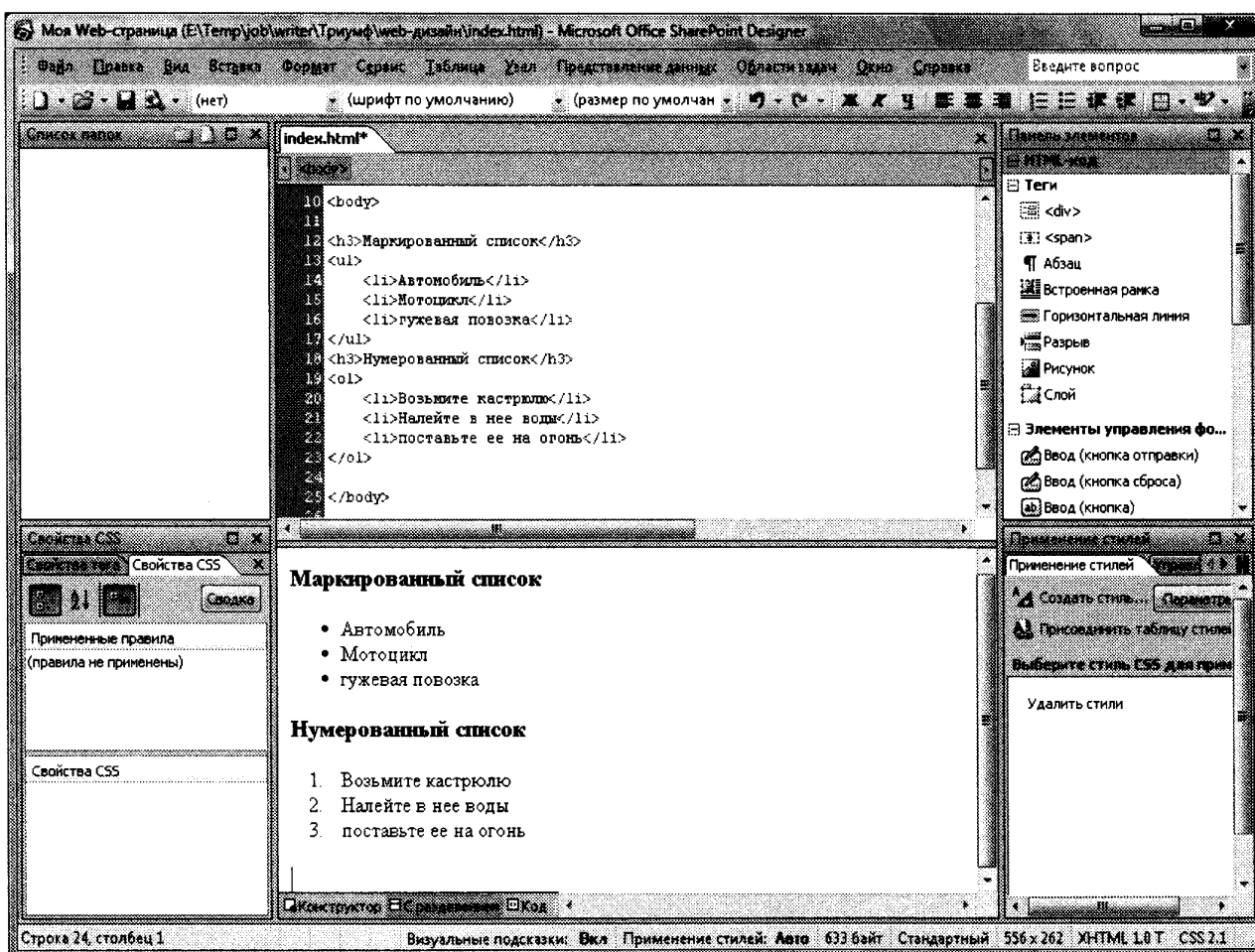


Рис. 2.17. Маркированный и нумерованный списки

Для составления списков терминов или понятий предназначены списки определений (Рис. 2.18). Это такие списки, каждый элемент которых состоит, в свою очередь, из двух частей: термина и его описания.

Список определений задается контейнером `<dl></dl>`. Термины задаются контейнерами `<dt></dt>`, а их описания контейнерами `<dd></dd>`.

Чтобы создать список определений:

- Щелчком мыши откройте раскрывающийся список **Стиль (Style)** () панели инструментов.
- В открывшемся списке щелкните мышью на строчке, подписанной **Определенный термин (Determined term)**.

Можете приступить к вводу содержимого списка. Каждый новый элемент добавляется нажатием клавиши . Термины и их описания будут чередоваться.

Чтобы закончить создание списка:

- Мышью или, нажав клавишу , переместите текстовый курсор вниз.
- Дважды нажмите клавишу .

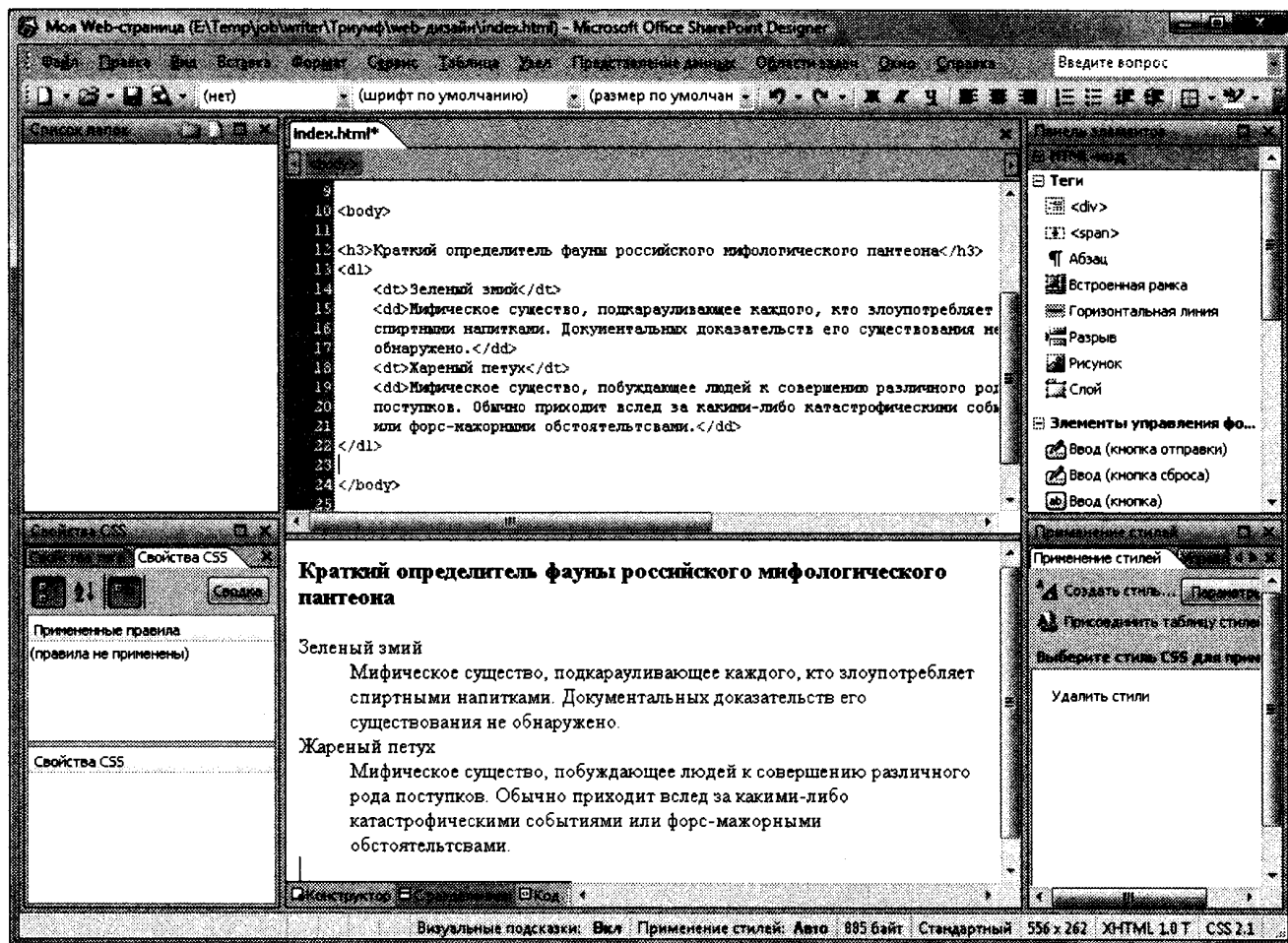


Рис. 2.18. Список определений


## Таблицы

Часто встречается необходимость располагать какие-либо данные в таблицах (Рис. 2.19). Более того, использование таблиц является одним из популярнейших средств оформления web-страниц. Подробно использование таблиц для оформления внешнего вида HTML-документов мы рассмотрим в третьей главе, а сейчас лишь кратко ознакомимся с основными элементами таблиц.

Основой HTML-таблиц является парный тег **<table>**, создающий контейнер, внутри которого размещается вся таблица.

Вся таблица, в свою очередь, состоит из столбцов и строк. Отдельные строки определяются контейнерами **<tr></tr>**. Каждая строка состоит из нескольких, по числу столбцов, ячеек, обозначаемых парными тегами **<td></td>**.

Удалим все содержимое текущей web-страницы:

- Щелкните мышью в области макета страницы.
- Выберите пункт меню **Правка ♦ Выделить все** (Edit ♦ Select All).
- Нажмите клавишу . Все содержимое web-страницы будет удалено.

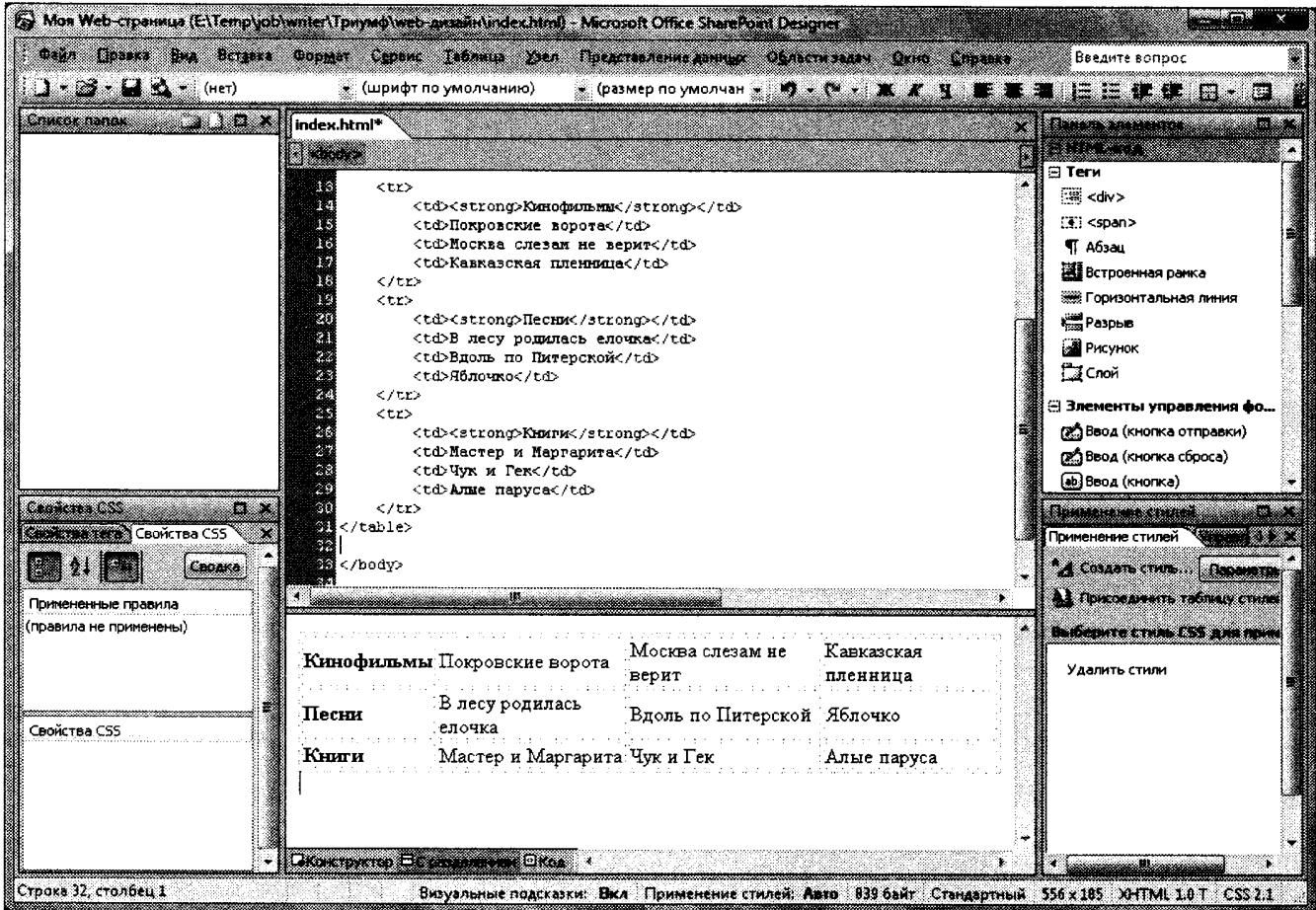



Рис. 2.19. Таблица

Теперь создадим новую таблицу в программе SharePoint Designer 2007.

- Щелкните мышью на кнопке **Добавить таблицу** (New Table) () панели инструментов программы. Появится меню выбора конфигурации таблицы.

В меню выбора конфигурации таблицы в схематическом виде представлена создаваемая вами таблица, каждый квадратик меню соответствует одной ячейке.

- Перемещая курсор мыши, добейтесь, чтобы в меню темным цветом было выделено столько строк и столбцов, сколько вам необходимо в создаваемой таблице.
- Выделив нужное количество квадратиков в меню, щелкните левой кнопкой мыши. Таблица будет создана.

Щелкнув мышью внутри любой ячейки созданной таблицы, вы получите возможность ввести туда текст.

При редактировании таблицы, нажимая клавишу **Tab**, вы будете перемещать текстовый курсор последовательно по ячейкам таблицы. С каждым нажатием клавиши **Tab** курсор будет смещаться на ячейку вправо, пока не достигнет правой границы таблицы. После этого нажатие на клавишу **Tab** приведет к переходу в левую ячейку следующей строки таблицы. Если текстовый курсор находился в правой нижней ячейке таблицы, то, нажав на клавишу **Tab**, вы создадите в таблице новую строку.

Если вас не устраивают высота, либо ширина столбцов и строк, их можно изменить.

- Подведите указатель мыши к любой границе таблицы, курсор примет вид стрелочки, вертикальной (↕) либо горизонтальной (↔).
- Нажмите левую кнопку мыши и, не отпуская ее, перетащите границу в другое положение.
- Отпустите кнопку мыши, при этом размеры таблицы изменятся.

## Гиперссылки

Одной из самых важных возможностей языка HTML является возможность ссылаться на другие документы и услуги сети Интернет (Рис. 2.20). В языке HTML ссылки, называемые еще гиперссылками, создаются с помощью парного тега `<a>`, атрибут `href` которого указывает адрес документа, на который ведет ссылка. Текст, помещающийся в контейнере `<a></a>`, помечается особым образом, и, щелкнув по нему, пользователь перейдет по ссылке. Например, контейнер `<a></a>` может содержать название какого-нибудь научного термина, а ссылка вести на энциклопедическую статью, описывающую этот термин. В таком случае, щелкнув по названию термина в HTML-документе, пользователь перейдет к документу, содержащему описание этого термина. Содержимое контейнера `<a></a>` называется элементом привязки.

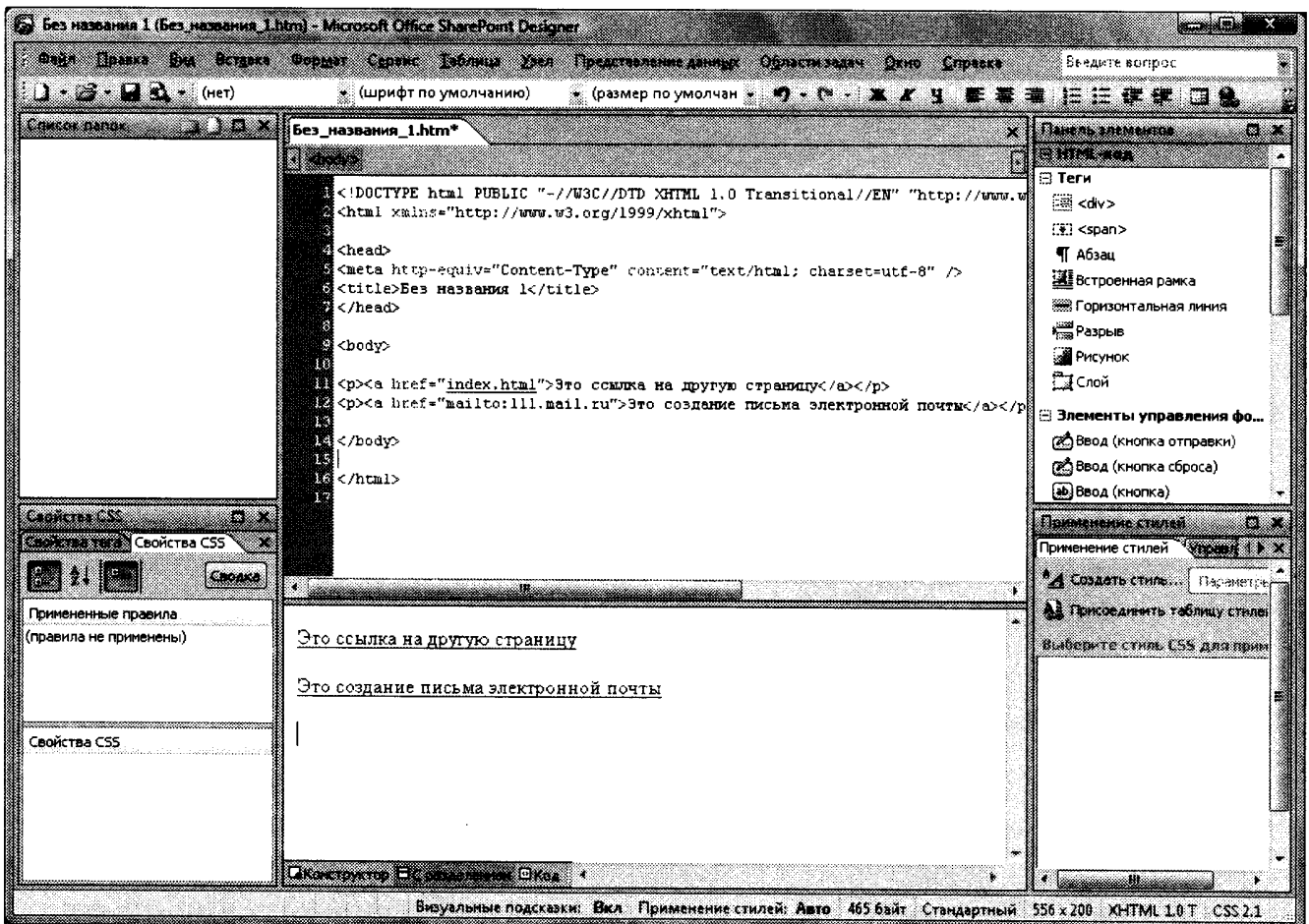




Рис. 2.20. Ссылка в HTML-документе

Ссылка может вести на другой документ, на определенную часть документа или вызывать некоторое действие, например создание письма электронной почты на определенный адрес или скачивание на компьютер пользователя какого-нибудь файла.

Удалите все содержимое текущей web-страницы.

- Щелкните мышью в области макета страницы.
- Выполните пункт меню **Правка ♦ Выделить все** (Edit ♦ Select All).
- Нажмите клавишу . Все содержимое web-страницы будет удалено.

Теперь создадим гиперссылку в программе SharePoint Designer 2007:

- Выберите команду меню **Вставка ♦ Гиперссылка** (Insert ♦ Hyperlink), либо
- Нажмите кнопку  на панели инструментов программы.

Появится диалог **Добавление гиперссылки** (Add hyperlink), Рис. 2.21. С помощью этого диалога можно создать несколько типов гиперссылок, но мы пока ограничимся созданием ссылки на новый документ, который будет создан после создания ссылки на него. В следующей главе мы рассмотрим работу с механизмом гиперссылок более подробно.

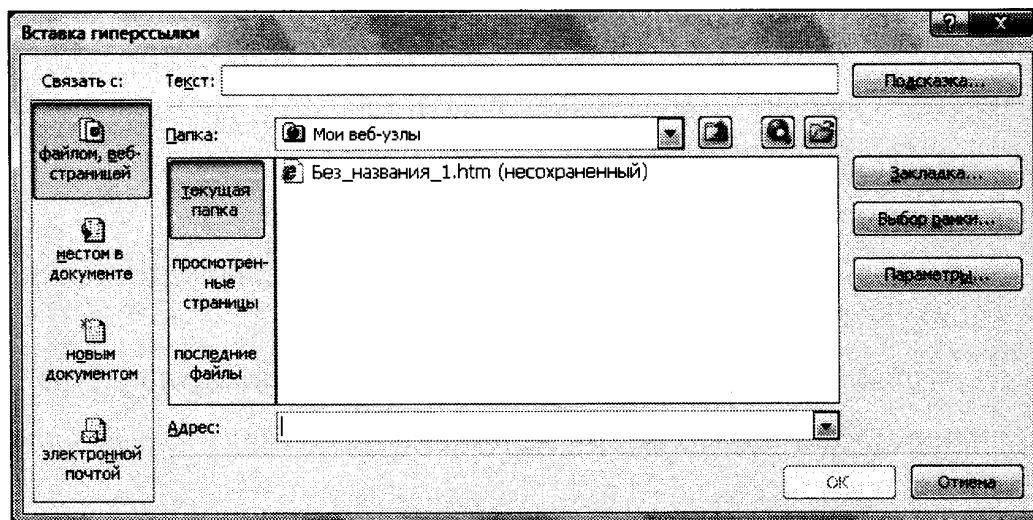


Рис. 2.21. Диалог **Добавление гиперссылки** (Add hyperlink)

- Нажмите кнопку **новым документом** (new document) на панели кнопок **Связать с** (Link with); диалог примет вид, представленный на Рис. 2.22.
- Щелкните мышью в поле ввода **Текст** (Text) и введите с клавиатуры слово «ссылка». Этот текст будет элементом привязки создаваемой ссылки.
- Щелкните мышью в поле ввода **Имя нового документа** (Document name) и введите название **link.htm**. Это будет название нового документа, на который создается ссылка.

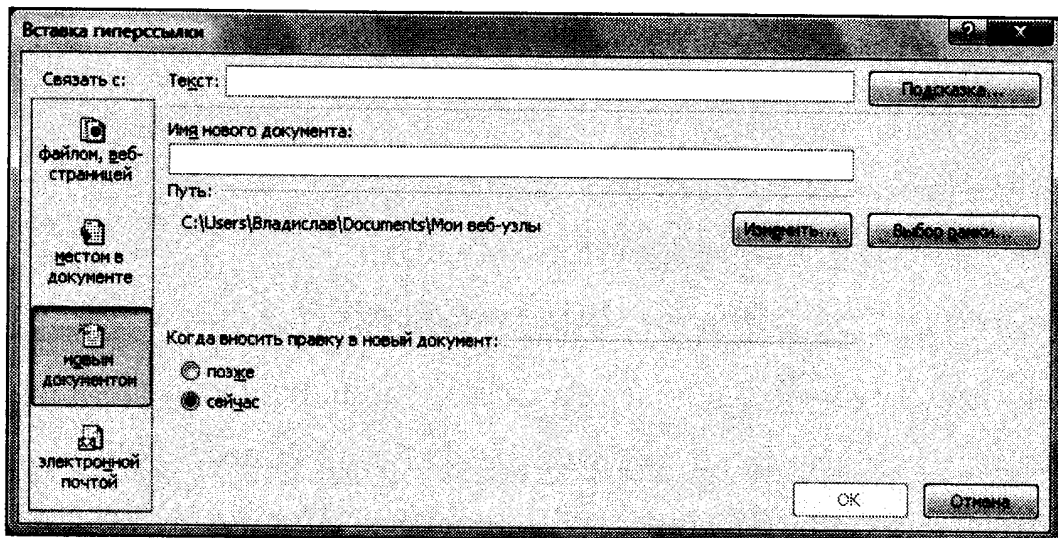


Рис. 2.22. Раздел **новым документом** (*new document*), диалога **Добавление гиперссылки** (*Add hyperlink*)

- Щелкните мышью на кнопке **ОК**. Будет создана гиперссылка в текущем HTML-документе, а также будет создан и открыт для редактирования новый HTML-документ.
- Еще одним способом создания гиперссылки является использование контекстного меню программы SharePoint Designer 2007.
- Введите текст, который будет элементом привязки гиперссылки.
- Выделите этот текст. Для этого переместите указатель мыши в начало выделяемого текста, нажмите левую клавишу мыши и, не отпуская ее, переместите указатель мыши к концу выделяемого текста.
- Отпустите левую клавишу мыши, текст будет выделен.
- После этого переместите указатель мыши в область выделенного текста и нажмите правую кнопку мыши. Появится контекстное меню.
- Щелкните мышью на пункте контекстного меню **Гиперссылка** (*Hyperlink*).


Появится диалог **Добавление гиперссылки** (*Add hyperlink*), представленный на Рис. 2.21.

- Нажмите кнопку **новым документом** (*new document*) панели кнопок **Связать с** (*Link with*), диалог примет вид, показанный на Рис. 2.22.

При использовании данного способа поле **Текст** (*Text*) диалога будет заполнено текстом, взятым из области выделения.

- Щелкните мышью в поле ввода **Имя нового документа** (*Document name*) и введите название «**link.htm**». Это будет название нового документа, на который создается ссылка.
- Щелкните мышью на кнопке **ОК**. Будет создана гиперссылка в текущем HTML-документе, а также будет создан и открыт для редактирования новый HTML-документ.

Итак, нами созданы гиперссылка и документ, на который она ссылается.

- Щелкните мышью в рабочей области созданного документа и напечатайте текст: «**сюда ведет ссылка**».
- Сохраните документ, щелкнув мышью на кнопке  панели инструментов программы.
- Закройте документ, щелкнув мышью на кнопке закрытия документа в области заголовков документов.

Теперь посмотрим полученный результат в web-браузере.

- Сделайте активным документ **index.html**, щелкнув мышью на его названии в области заголовков программы **SharePoint Designer 2007**.
- Выберите команду меню **Файл ♦ Просмотреть в обозревателе ♦ Microsoft Internet Explorer** (File ♦ Preview in web-browser ♦ Microsoft Internet Explorer). На экране появится окно браузера (Рис. 2.23). Страница откроется в браузере Internet Explorer.

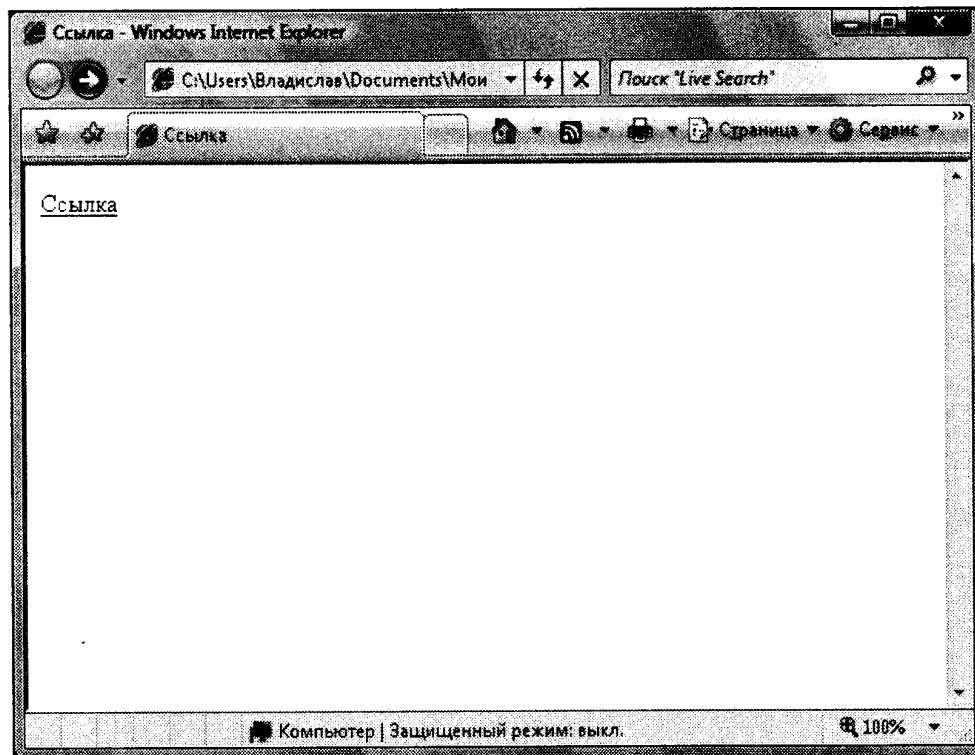
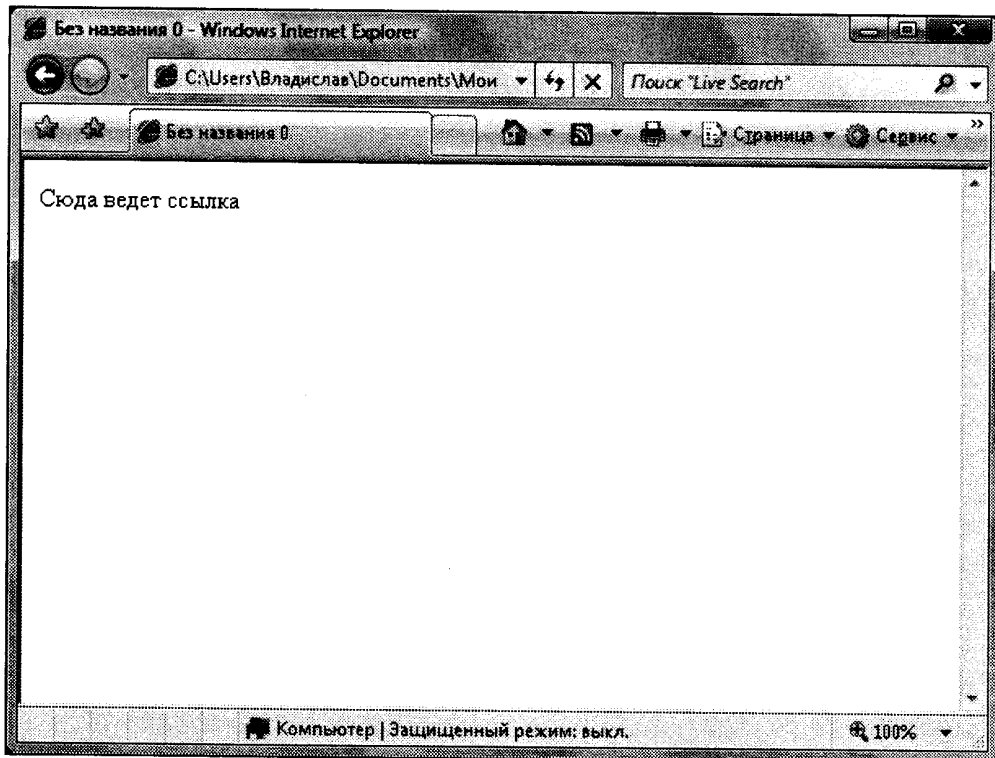


Рис. 2.23. Гиперссылка в web-браузере Microsoft Internet Explorer

- Щелкните мышью на гиперссылке «ссылка». В браузере откроется документ **link.htm**, в окне браузера отобразится его содержимое (Рис. 2.24).

Теперь вы знакомы со всеми основными элементами оформления web-страниц. Вам теперь по силам создать полноценный web-сайт, состоящий из нескольких связанных друг с другом страниц, с системой заголовков, списками и простыми таблицами. Созданию вашего первого web-сайта посвящен следующий раздел этой главы.





*Рис. 2.24. Результат перехода по гиперссылке в браузере Microsoft Internet Explorer*


## ОСНОВЫ ИСПОЛЬЗОВАНИЯ КАСКАДНЫХ ТАБЛИЦ СТИЛЕЙ

Теперь, когда вы познакомились с основами языка разметки HTML, пришло время познакомиться с технологией CSS. Представьте себе такую ситуацию: у вас на странице имеется несколько рисунков с одинаковыми размерами ширины и высоты. Забегая немного вперед, я скажу, что хорошим тоном считается указание этих параметров для изображений. Используя лишь средства HTML, вам пришлось бы указывать эти свойства для каждого отдельного изображения при помощи атрибутов **width** и **height**. А изменить шрифт с Times New Roman на какой-либо другой при помощи языка разметки вообще невозможно. Для устранения подобных недостатков HTML 17 декабря 1996 года была принята технология CSS – Cascading Style Sheets (Каскадные таблицы стилей). Актуальная на сегодняшний день версия была принята в 1998 году и откорректирована 6 ноября 2006 – CSS 2.1.

Технология CSS позволяет объединять общие свойства различных элементов страницы в именованные стили, или классы. В дальнейшем вам достаточно указать для элемента лишь название класса при помощи атрибута **class**, избегая ввода нескольких атрибутов, описывающих свойства объекта. Это позволяет уменьшить размер страницы, время, затрачиваемое на ее создание, а также сделать исходный код более наглядным.

Работая в программе SharePoint Designer 2007, для использования достоинств технологии CSS необязательно изучать ее синтаксис и стандарты, хотя, несомненно, подобные знания помогли бы в оптимизации вашего сайта. Программа сама создает стили, и для их применения к элементам достаточно лишь щелкнуть мышью.

Удалим все содержимое текущей web-страницы:

- Щелкните мышью в области макета страницы.
- Выберите пункт меню **Правка ♦ Выделить все** (Edit ♦ Select All).
- Нажмите клавишу . Все содержимое web-страницы будет удалено.

Теперь создадим стиль для абзаца текста.

- Введите какой либо текст, например «**Технология CSS прекрасно дополняет возможности HTML**».
- При помощи мыши выделите данный абзац.
- На панели инструментов щелкните мышью на кнопке **Цвет текста** (Text Color) и выберите темно-зеленый.
- На панели инструментов щелкните мышью на открывающемся списке **Размер** (Size) и выберите **large (18 pt)** (Большой (18 pt)).
- На панели инструментов щелкните мышью на открывающемся списке **Шрифт** (Font) и выберите **Arial, Helvetica, sans-serif**.

Обратите внимание, что на расположенной справа внизу области рабочего окна программы **Применение стилей** (Style Using) появилось название стиля – **style1**.

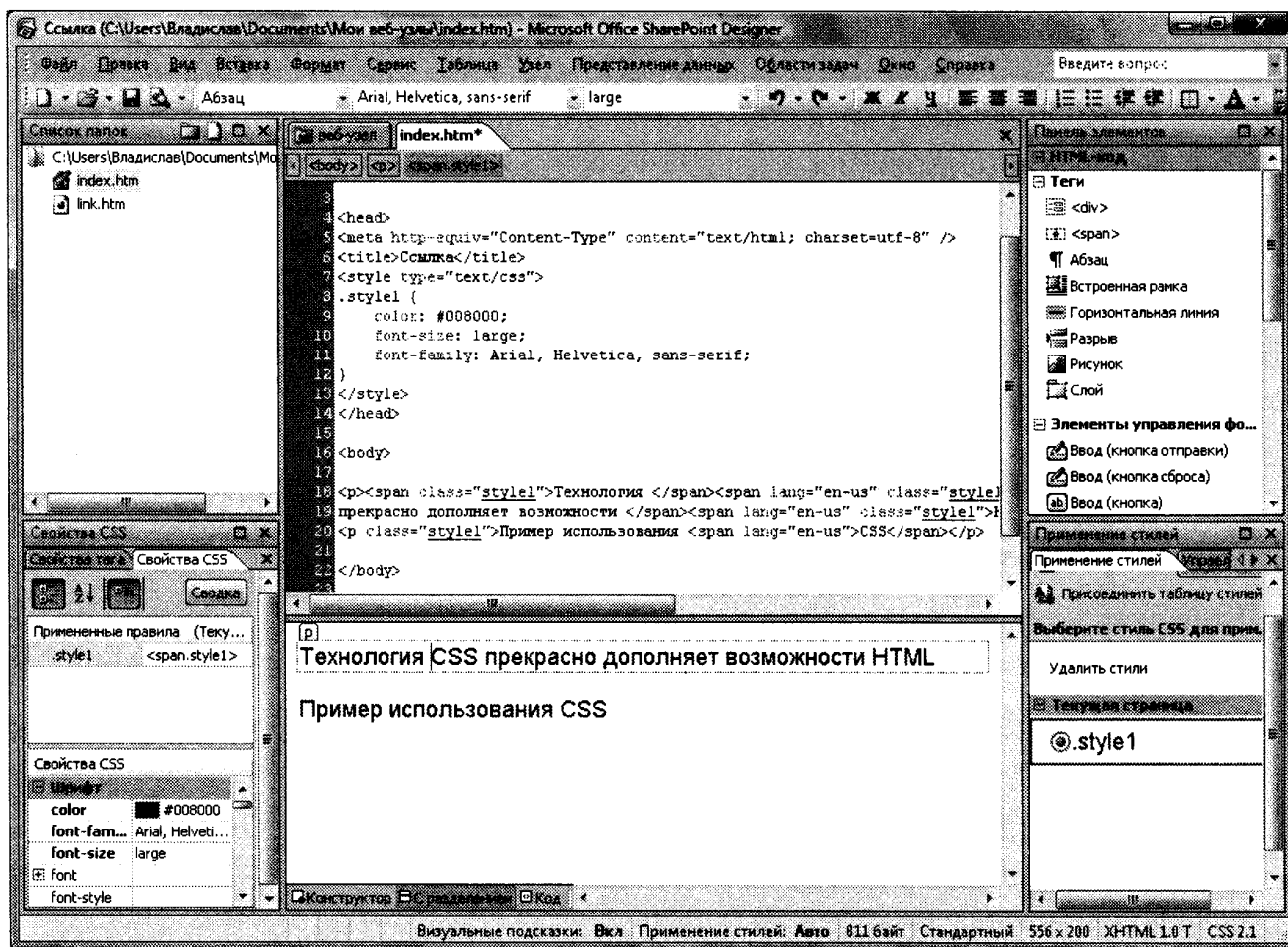



Рис. 2.25. Применение каскадных таблиц стилей

- ▶ Нажмите на клавиатуре клавишу  для создания нового абзаца и введите «**Пример использования CSS**».
- ▶ При помощи мыши выделите данный абзац.
- ▶ Щелкните мышью на названии стиля **style 1**.

Теперь оба абзацы выглядят совершенно одинаково. Переключившись в режим исходного кода, вы можете увидеть расположенное в заголовке страницы описание стиля, заключенное в фигурные скобки (Рис. 2.25).

Как видите, применение стилей может сильно облегчить вашу работу и раскрывает дополнительные возможности. Более подробно о **CSS** мы поговорим в следующей главе.

## Ваш первый web-сайт

Как правило, первым сайтом, который делает человек, приходя в мир **WWW**, является домашняя страничка, его визитная карточка. Не будем отступать от этого правила и применим только что полученные знания для создания несложной домашней страницы. Не стоит воспринимать ее как шаблон для создания вашей собственной домашней страницы, поскольку главной целью ее создания является не она сама, а практическое применение способов оформления **HTML**-страниц и основ строения **web**-сайтов.

Как уже говорилось ранее, статический **web**-сайт представляет собой набор **HTML**-файлов, расположенных в нескольких каталогах **web**-сервера. Один из этих **HTML**-файлов, который автоматически открывается браузером пользователя при посещении такого **web**-сайта, называется основной страницей сайта. Обычно на ней размещают общую информацию о сайте и ссылки на остальные разделы сайта. Страницы, связанные с основной страницей сайта, называют второстепенными.

Чтобы сделать **HTML**-документ основной страницей сайта, ему достаточно присвоить название **index.html**.

## Создание нового web-сайта

Создадим новый **web**-сайт в программе **SharePoint Designer 2007**.

- ▶ Выберите команду меню **Файл ♦ Создать ♦ Веб-узел** (**File ♦ New ♦ Web-host**). Откроется диалог **Создание** (**Creation**) (Рис. 2.26).
- ▶ Щелкните мышью на кнопке **Обзор** (**Browse**) группы настроек **Параметры** (**Properties**). Откроется диалог **Место для нового веб-узла** (**Place for new Web-site**), представленный на Рис. 2.27.
- ▶ Выберите или создайте папку, в которой будут размещаться файлы **web**-сайта, и щелкните мышью на кнопке **Открыть** (**Open**), перейдите к выбранной папке и закройте диалог **Место для нового веб-узла** (**Place for new Web-site**), щелкнув мышью на кнопке **Открыть** (**Open**).

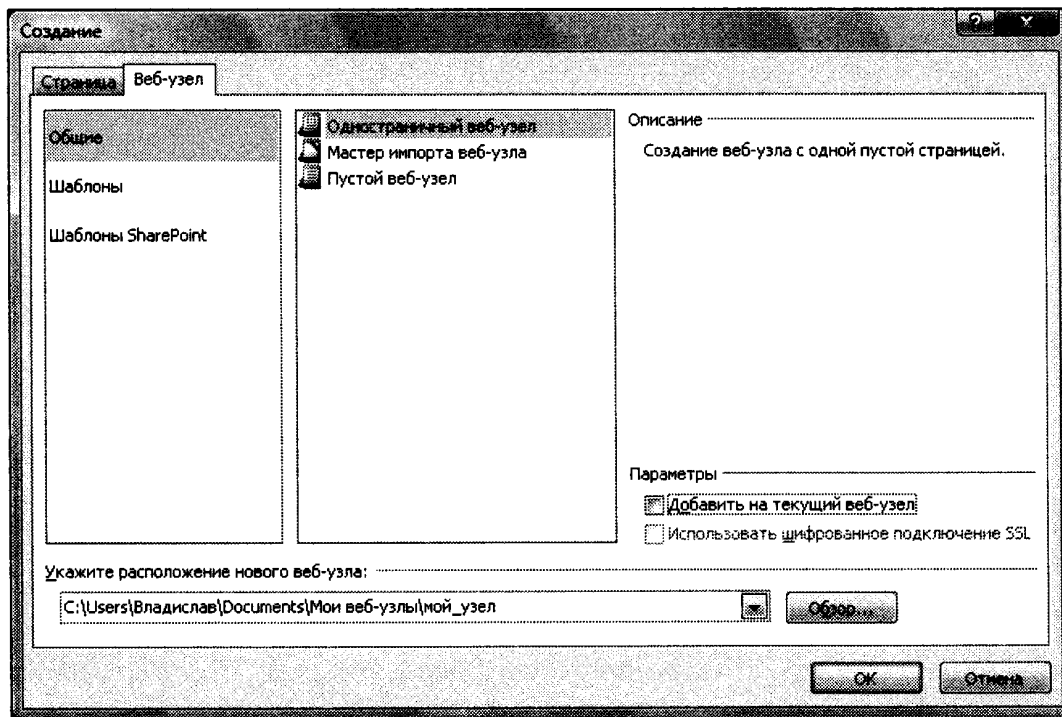


Рис. 2.26. **Создание** (Creation)

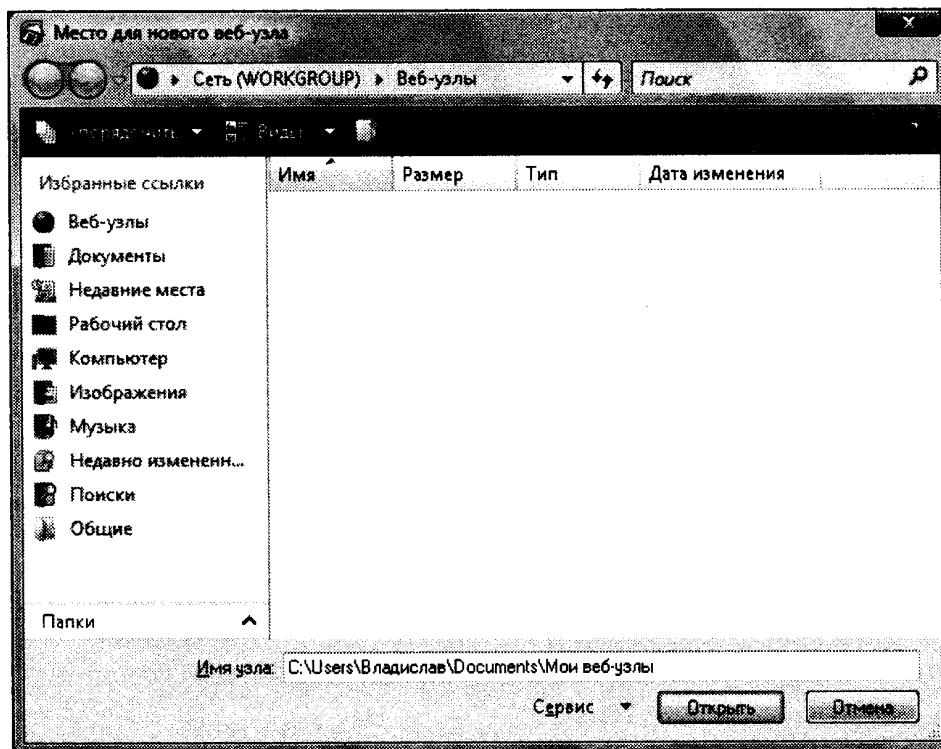


Рис. 2.27. Диалог **Место для нового веб-узла**  
(Place for new Web-site)

- Щелкните мышью на ярлыке **Одностраничный веб-узел** (Single page Web-site).
- Щелкните мышью на кнопке **ОК**. Диалог закроется, будут созданы структура каталогов web-сайта и его главная страница (Рис. 2.28).

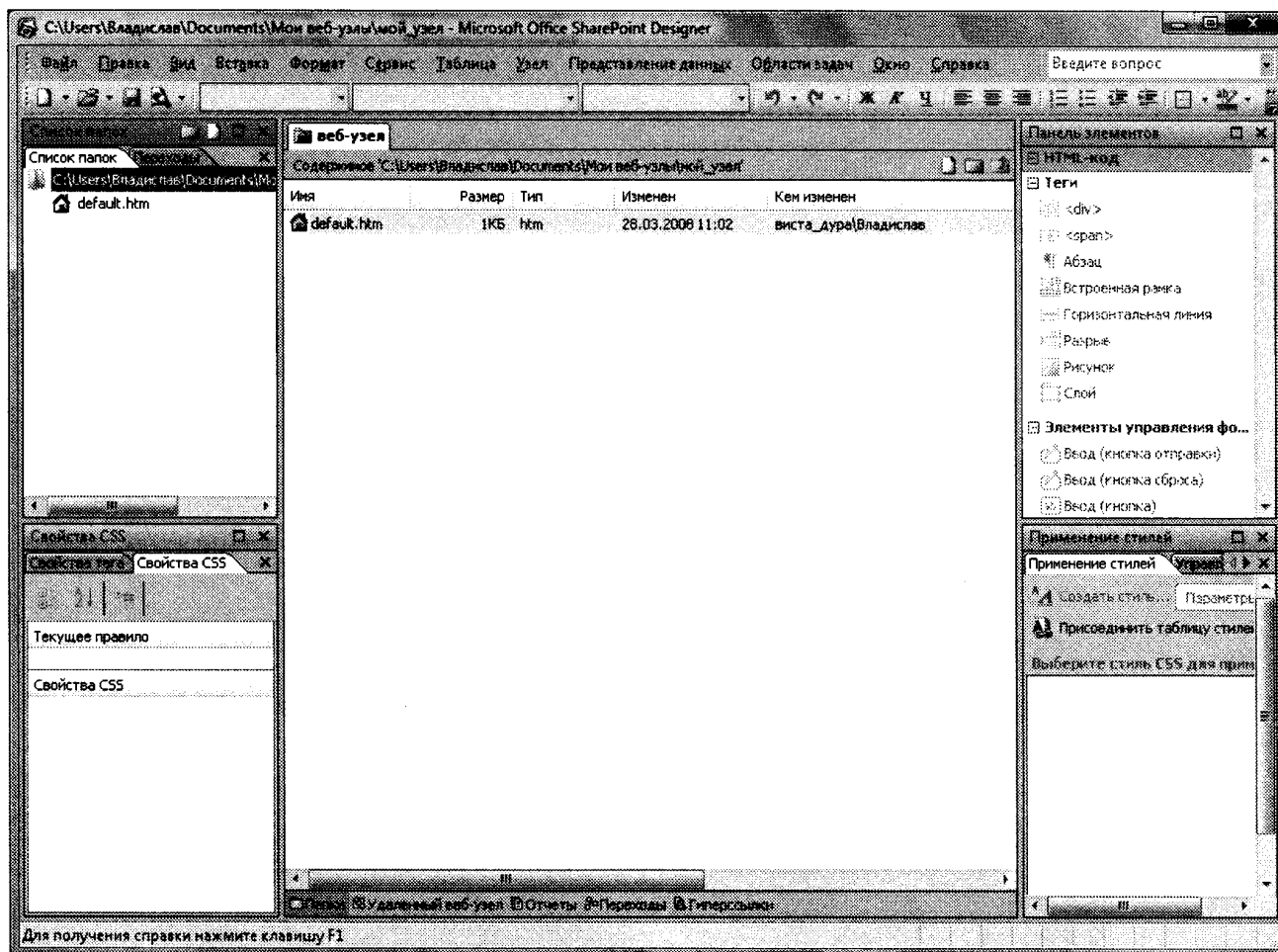


Рис. 2.28. Одностраничный web-узел

## Создание главной страницы

Приступим к созданию содержимого главной страницы вашего первого web-сайта.

Единственная страница вашего сайта имеет название **default.htm**. Чтобы эта страница являлась главной страницей сайта, ей необходимо присвоить название **index.htm**.

- Щелкните правой кнопкой мыши на названии файла **default.htm**, выберите в открывшемся меню пункт **Переименовать** (Rename) и введите новое название **index.htm**.
- В открывшемся диалоге **Подтверждение переименования** (Renaming confirmation) щелкните мышью на кнопке **ОК**.
- Дважды щелкните левой кнопкой мыши на имени главной страницы сайта – **index.htm**. Страница откроется в режиме редактирования.

Чтобы созданный вами документ везде корректно определялся как написанный на русском языке, проделайте следующее:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties), представленный на Рис. 2.29.

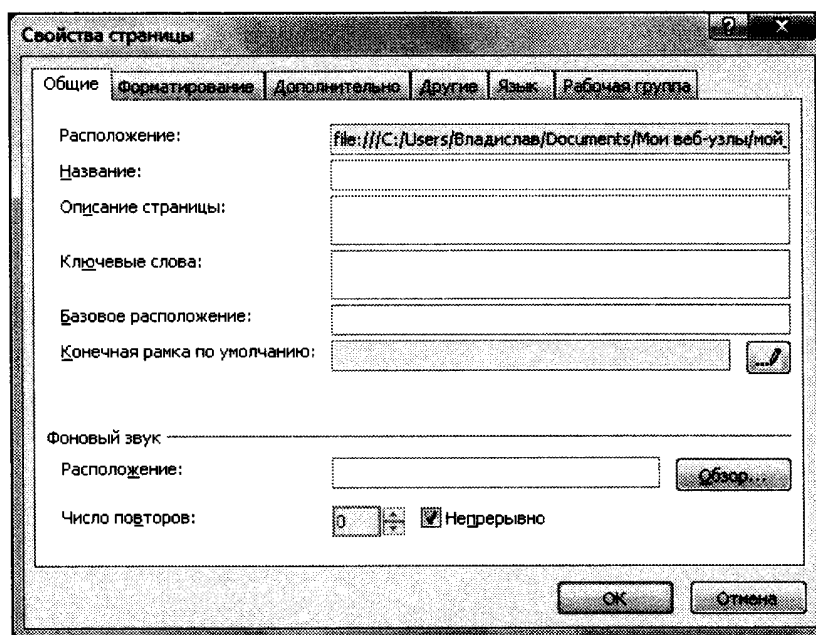


Рис. 2.29. Диалог **Свойства страницы** (Page properties)

- Щелкните мышью на вкладке **Язык** (Language), откроется диалог настройки языка страницы (Рис. 2.30).

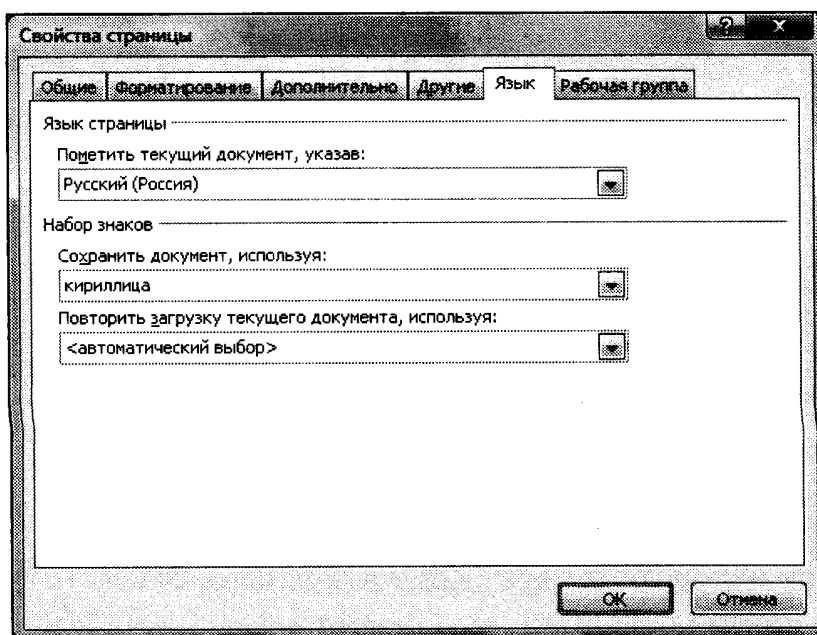


Рис. 2.30. Вкладка **Язык** (Language) диалога **Свойства Страницы** (Page properties)

- В открывающемся меню **Пометить текущий документ, указав:** (Mark current document as) группы настроек **Язык страницы** (Page Language) выберите пункт **русский** (russian).
- В открывающемся меню **Сохранить документ, используя** (Save document using) группы настроек **Набор знаков** (Character set) выберите пункт **кириллица** (Cyrillic).

- Щелкните мышью на кнопке **ОК**, чтобы сохранить изменения и закрыть диалог настройки.

Назначьте название главной странице сайта:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties).
- Щелкните мышью на поле ввода **Название** (Title) и введите название: «**Главная страница**» (Рис. 2.31).

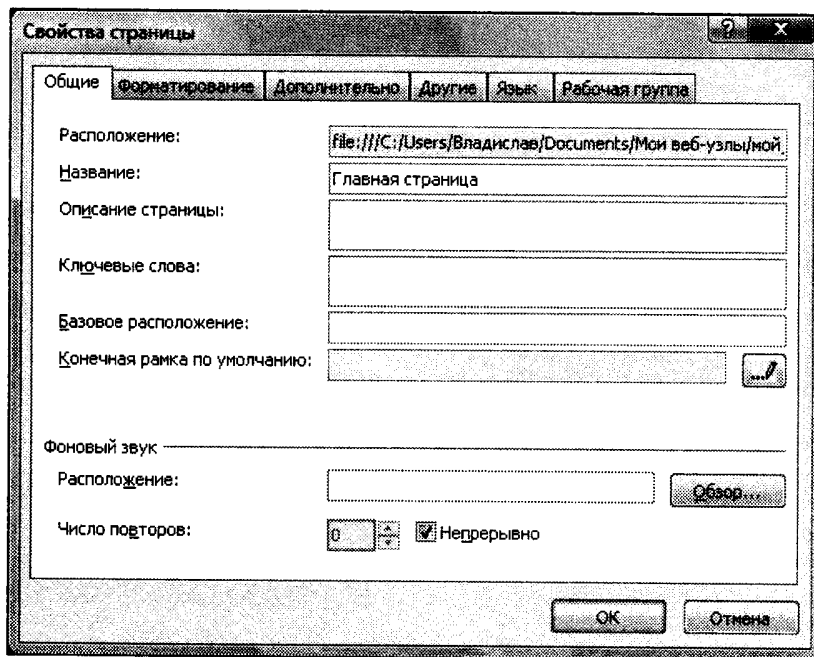


Рис. 2.31. Ввод названия страницы

- Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Главная страница вашего первого web-сайта будет состоять из одного заголовка, некоторого количества текста и нескольких ссылок на другие страницы сайта. Приступим к наполнению страницы.

Создайте заголовок главной страницы.

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ()
- В открывшемся списке щелкните мышью на строчке, подписанной **Заголовок 1**, чтобы создать заголовок первого уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: «**Моя домашняя страница**».
- Нажмите клавишу **Enter**, чтобы завершить ввод заголовка.

Пришел черед основного текста страницы.


- Введите с клавиатуры следующий текст:

Домашняя страница является визитной карточкой человека в WWW, слепком его «Я», поэтому я постараюсь по возможности полно отразить свой внутренний мир на этих нескольких страничках. Чтобы познакомиться со мною поближе, побродите по разделам сайта, ссылки на которые представлены чуть ниже.

Теперь отделите горизонтальной линией раздел текста от раздела ссылок:

- Из расположенной справа области **HTML-код** (HTML-code) перетащите мышью элемент **Горизонтальная линия** (Horizontal ray) и расположите его ниже введенного текста.

Теперь необходимо вставить ссылки на другие страницы сайта и небольшие комментарии к этим ссылкам. Поскольку остальные страницы еще не созданы, вместе с созданием ссылок мы будем создавать и сами страницы.

- Нажмите кнопку  на панели инструментов. Появится диалог **Добавление гиперссылки** (Add hyperlink), представленный на Рис. 2.32.

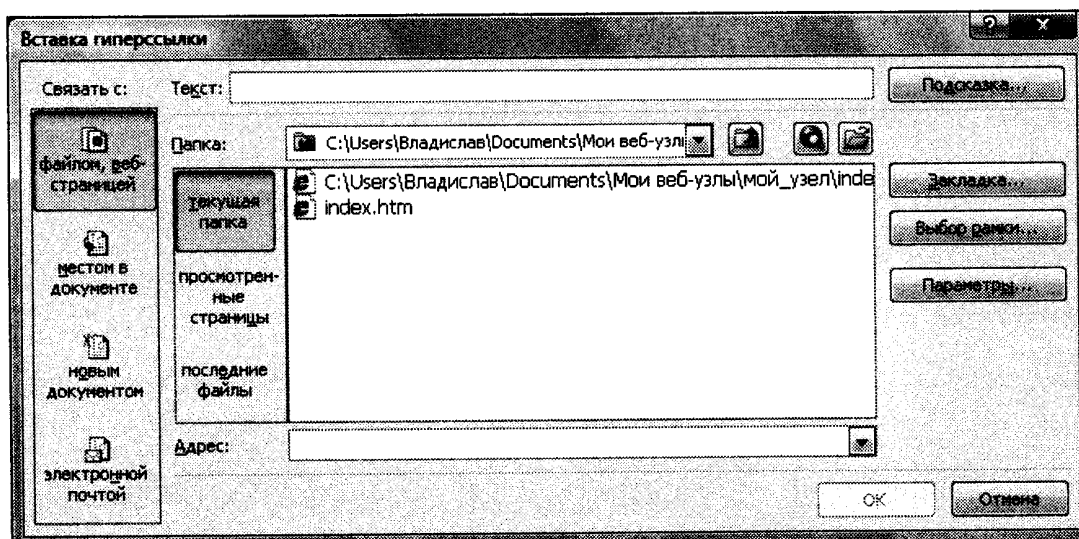


Рис. 2.32. Диалог **Добавление гиперссылки** (Add hyperlink)

- Нажмите кнопку **новым документом** (new document) панели кнопок **Связать с** (Link with), диалог примет вид, указанный на Рис. 2.33.
- Щелкните мышью в поле ввода **Текст** (Text) и введите с клавиатуры «Обо мне». Этот текст будет элементом привязки создаваемой ссылки.
- Щелкните мышью в поле ввода **Имя нового документа** (Document name) и введите название **about.htm**. Это будет название нового документа, на который создается ссылка.
- Щелкните мышью на кнопке **ОК**. Будет создана гиперссылка в текущем HTML-документе, и будет создан и открыт для редактирования новый HTML-документ под названием **about.htm**.



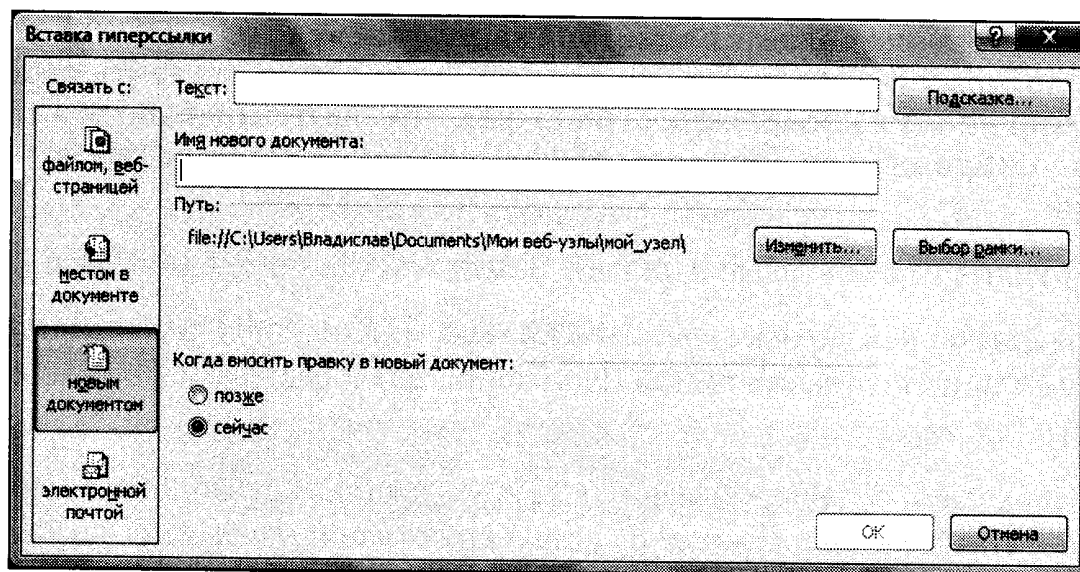


Рис. 2.33. Раздел **новым документом** (*new document*), диалога **Добавление гиперссылки** (*Add hyperlink*)

К редактированию файла **about.htm** мы вернемся чуть позже, а сейчас завершим создание главной страницы сайта.

- Щелкните мышью на названии **index.htm** в области заголовков открытых документов. Вы вернетесь к редактированию главной страницы.
- Щелкните мышью справа от только что созданной гиперссылки и введите следующий текст: «- **Некоторая личная информация, моя краткая биография и основные этапы жизненного пути.**»

Вы создали первую ссылку на другую страницу и небольшой комментарий к этой ссылке.

- Нажмите клавишу **Enter**, чтобы начать новый абзац, и точно таким же образом, как первую, создайте вторую ссылку с комментарием.

Ссылка должна вести на страницу **hobby.htm**, текст ссылки: «**Интересы**», а текст комментария: «- **мои стремления, увлечения, хобби. То, что мне интересно.**»

- После этого опять начните новый абзац и создайте третью, последнюю на этой странице, ссылку.

Эта ссылка будет вести на страницу **links.htm**, текст ее: «**Интересные ссылки**», а комментарий к ней: «- **ссылки на сайты, которые показались мне интересными.**»

Завершим создание главной страницы еще одной горизонтальной линией:

- Из расположенной справа области **HTML-код** (HTML-code) перетащите мышью элемент **Горизонтальная линия** (Horizontal ray) и расположите его ниже последнего абзаца.

Полученная в результате страница должна выглядеть, как на Рис. 2.34.

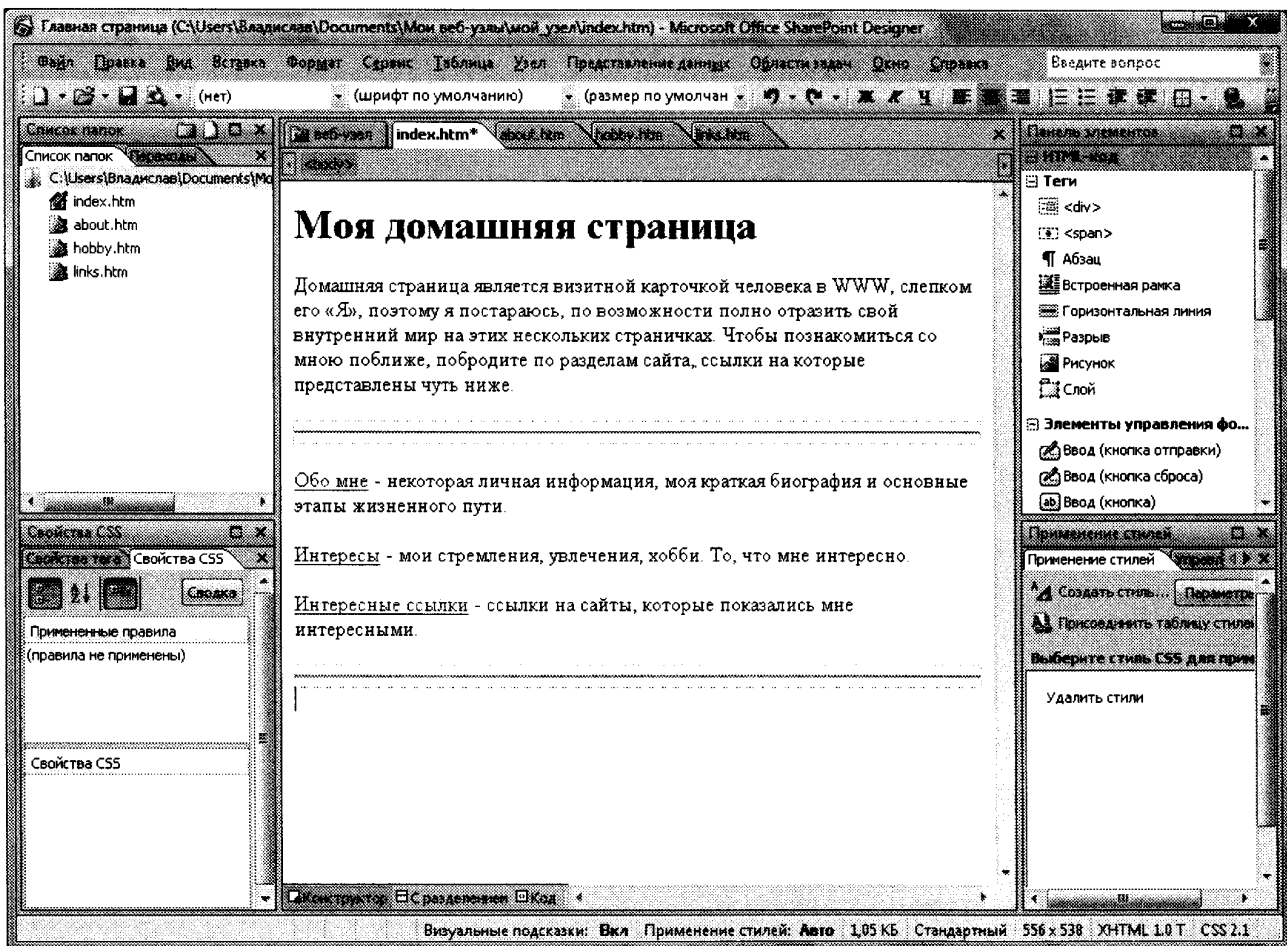



Рис. 2.34. Главная страница сайта

Не забудьте сохранить главную страницу, щелкнув мышью на кнопке  панели инструментов программы, и приступайте к редактированию страницы сайта «Обо мне».

## Создание раздела сайта «Обо мне»

- Щелкните мышью на названии страницы **about.htm** в области заголовков документов. Файл **about.htm**, содержащий раздел «Обо мне», станет активным и доступным для редактирования.

Чтобы созданный вами документ везде корректно определялся, как написанный на русском языке, сделайте следующее:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства ♦ страницы** (Page properties), представленный на Рис. 2.29.
- Щелкните мышью на вкладке **Язык** (Language), откроется диалог настройки языка страницы (Рис. 2.30).
- В открывающемся меню **Пометить текущий документ, указав:** (Mark current document as) группы настроек **Язык страницы** (Page Language) выберите пункт **русский** (russian).

- В открываемом меню **Сохранить документ, используя** (Save document using) группы настроек **Набор знаков** (Character set) выберите пункт **кириллица** (Cyrillic).
- Щелкните мышью на кнопке **ОК**, чтобы сохранить изменения и закрыть диалог настройки.

Назначим название этой странице:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties).
- Щелкните мышью на поле ввода **Название** (Title) и введите название: **«Обо мне»**.
- Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Раздел сайта **«Обо мне»** будет содержать один заголовок 1-го уровня, два заголовка 2-го уровня, краткую информацию об авторе страницы, упорядоченную в виде списка определений, и несколько слов о жизненном пути автора.




Создадим главный заголовок страницы:

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ( ▾) панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 1**, чтобы создать заголовок первого уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: **«Немного о себе»**.
- Нажмите клавишу , чтобы завершить ввод заголовка.

Создадим подзаголовок:

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ( ▾) панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 2**, чтобы создать заголовок второго уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: **«Личная информация»**
- Нажмите клавишу , чтобы завершить ввод заголовка.



Теперь создайте список определений, в который будет сведена информация личного характера. В качестве терминов будут различные личные параметры, а в качестве описания терминов – описание этих параметров.

- Щелчком мыши откройте раскрывающийся список **Стиль (Style)** () панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Определенный термин (Determined term)**.
- Введите первый личный параметр: «**Фамилия Имя Отчество**»
- Нажмите клавишу , чтобы перейти к написанию значения этого параметра. Введите текст: «**Трамвай Иванович Вагонов**».
- Нажмите клавишу , чтобы перейти ко второму элементу списка.
- Введите остальные элементы списка. Второй элемент: «**Дата рождения**», его значение – «**21 марта 2002 года**». Третий элемент: «**Образование**», его значение – «**Три класса церковно-приходской школы**». Четвертый элемент: «**Иностранные языки**», его значение – «**Зулу, хинди, суахили, немного наречие браминов**». Пятый и последний элемент: «**Семейное положение**», его значение – «**Холост**».

Введя все данные, закончим создание списка:

- Дважды нажмите клавишу .


Создадим следующий подзаголовок:

- Щелчком мыши откройте раскрывающийся список **Стиль (Style)** () панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 2**, чтобы создать заголовок второго уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: «**Биография**».
- Нажмите клавишу , чтобы завершить ввод заголовка.

Завершим создание раздела краткой биографией:

- Введите текст биографии: «**Родился. Вырос. Пошел в школу. Закончил, пошел работать. Этим занимаюсь и по сей день.**»

Вот вы и создали вторую страницу сайта, она должна выглядеть так, как показано на Рис. 2.35.

- Не забудьте сохранить проделанную работу, щелкнув мышью на кнопке  панели инструментов программы, и приступайте к редактированию страницы сайта «**Интересы**».

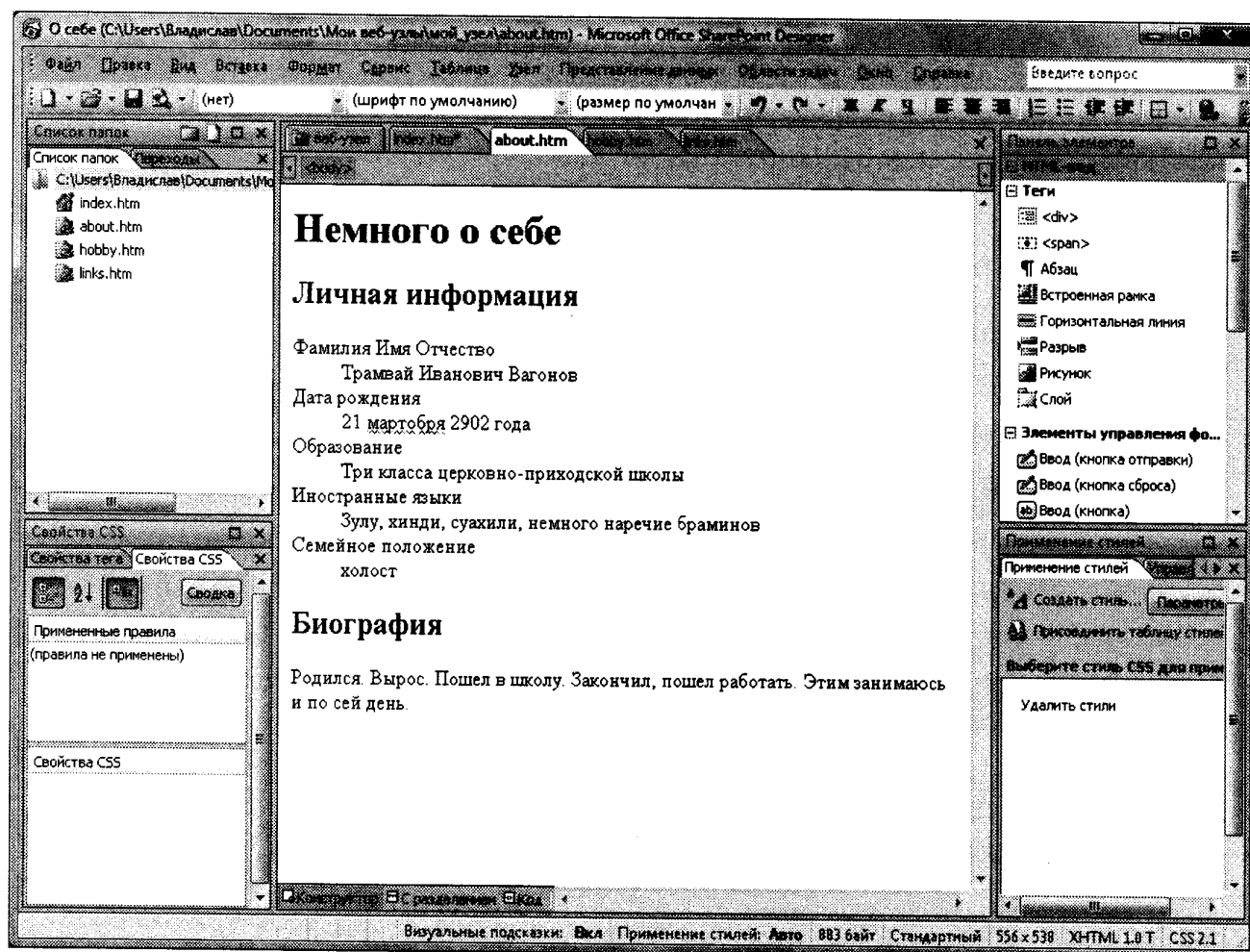


Рис. 2.35. Раздел домашней страницы «Обо мне»

## Создание раздела сайта «Интересы»

- Щелкните мышью на названии страницы **hobby.htm** в области заголовков документов. Файл **hobby.htm**, содержащий раздел «Интересы», станет активным для редактирования.

Чтобы созданный вами документ везде корректно определялся как написанный на русском языке, проделайте следующее:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties), Рис. 2.29.
- Щелкните мышью на вкладке **Язык** (Language), откроется диалог настройки языка страницы (Рис. 2.30).
- В открывающемся списке **Пометить текущий документ, указав:** (Mark current document as) группы настроек **Язык страницы** (Page Language) выберите пункт **русский** (russian).
- В открывающемся списке **Сохранить документ, используя** (Save document using) группы настроек **Набор знаков** (Character set) выберите пункт **кириллица** (Cyrillic).

- Щелкните мышью на кнопке **ОК**, чтобы сохранить изменения и закрыть диалог настройки.

Назначим название этой странице:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties).
- Щелкните мышью на поле ввода **Название** (Title) и введите название: **«Мои интересы»**.
- Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Раздел сайта «Интересы» будет содержать один заголовок 1-го уровня, два заголовка 2-го уровня и краткое перечисление интересов автора, разбитых по двум категориями: «музыка» и «спорт». Категория «музыка» будет упорядочена в виде маркированного списка, а категория «спорт» в виде нумерованного списка.

Создадим главный заголовок страницы:

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ( ▾) панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 1**, чтобы создать заголовок первого уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: **«Мои пристрастия»**.
- Нажмите клавишу , чтобы завершить ввод заголовка.

Добавим небольшое описание содержимого этой страницы:


- Введите текст: **«Здесь приведены мои основные интересы, разделенные на несколько категорий.»**

Создадим подзаголовок:

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ( ▾) панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 2**, чтобы создать заголовок второго уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: **«Музыка»**.
- Нажмите клавишу , чтобы завершить ввод заголовка.

Теперь создадим маркированный список, в котором будут перечислены все музыкальные направления, которыми увлекается автор страницы.



- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) ( ▾) панели инструментов.

- В открывшемся меню щелкните мышью на строке, подписанной **Маркированный список** (Unordered list).
- Введите первый пункт списка, «**Rock'n'Roll**».
- Нажмите клавишу , чтобы перейти к вводу второго элемента списка.
- Введите остальные элементы списка. Пункты такие: «**Art-Rock**», «**Acid Jazz**», «**Drum'n'Bass**», «**Trip-Hop**», «**New Age**».



Введя все данные, закончим создание списка:

- Дважды нажмите клавишу .

Создадим следующий подзаголовок:

- Щелчком мыши откройте раскрывающийся список **Стиль** (Style) () панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 2**, чтобы создать заголовок второго уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: «**Спорт**».
- Нажмите клавишу , чтобы завершить ввод заголовка.


Теперь создадим нумерованный список, в котором будут перечислены в порядке убывания значимости виды спорта.

- Щелчком мыши откройте открывающийся список **Стиль** (Style) () панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Нумерованный список** (Ordered list).
- Введите первый пункт списка «**Роликовые коньки**».
- Нажмите клавишу , чтобы перейти к вводу второго элемента списка.
- Введите остальные элементы списка. Пункты такие: «**Ледовые коньки**», «**Лыжи**», «**Плавание**», «**Рыбалка**».

Введя все данные, закончим создание списка:

- Дважды нажмите клавишу .

Еще один раздел успешно создан, полученная страница должна выглядеть так, как показано на Рис. 2.36.

- Не забудьте сохранить проделанную работу, щелкнув мышью на кнопке  панели инструментов программы, и приступайте к редактированию последней страницы сайта «**Интересные ссылки**».

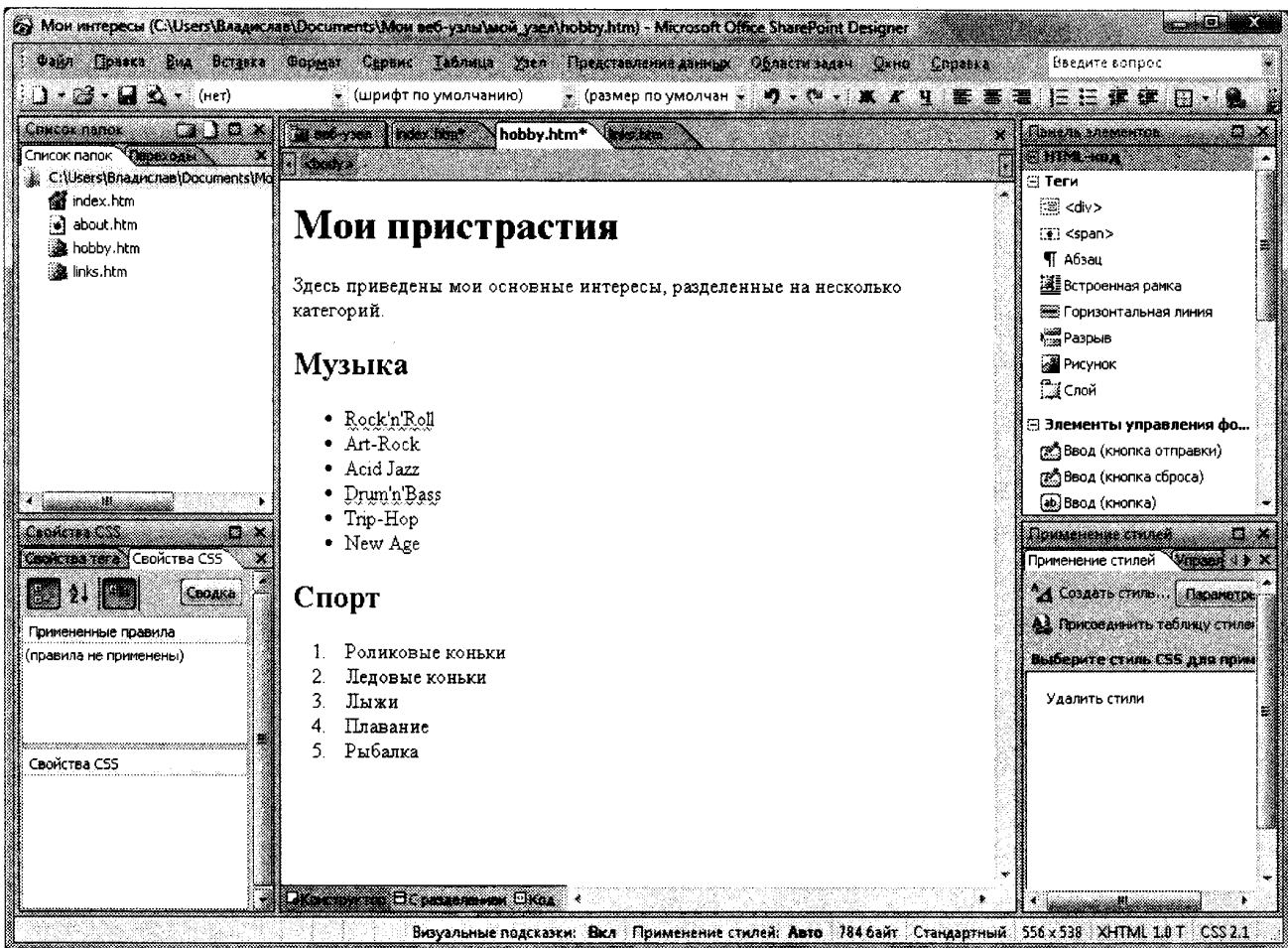


Рис. 2.36. Раздел домашней страницы «Интересы»

## Создание раздела сайта «Интересные ссылки»

- Щелкните мышью на названии страницы **links.htm** в области заголовков документов. Файл **links.htm**, содержащий раздел «Интересные ссылки», станет активным для редактирования.

Чтобы созданный вами документ везде корректно определялся как написанный на русском языке, сделайте следующее:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties), Рис. 2.29.
- Щелкните мышью на вкладке **Язык** (Language), откроется диалог настройки языка страницы (Рис. 2.30).
- В открывающемся меню **Пометить текущий документ, указав:** (Mark current document as) из группы настроек **Язык страницы** (Page Language) выберите пункт **русский** (russian).
- В открывающемся списке **Сохранить документ, используя** (Save document using) группы настроек **Набор знаков** (Character set) выберите пункт **кириллица** (Cyrillic).





- Щелкните мышью на кнопке **ОК**, чтобы сохранить изменения и закрыть диалог настройки.

Назначим название этой странице:

- Выберите команду меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties).
- Щелкните мышью на поле ввода **Название** (Title) и введите название: «**Интересные ссылки**».
- Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Раздел сайта «**Интересы**» будет содержать один заголовок 1-го уровня и небольшую таблицу, в которую будут сведены ссылки на разные web-ресурсы и их краткое описание.


Создадим главный заголовок страницы:

- Щелчком мыши откройте открывающийся список **Стиль** (Style) () панели инструментов.
- В открывшемся списке щелкните мышью на строке, подписанной **Заголовок 1**, чтобы создать заголовок первого уровня.
- Щелкните мышью в области макета страницы и введите текст заголовка: «**Интересные ссылки**».
- Нажмите клавишу , чтобы завершить ввод заголовка.

Добавим небольшое описание содержимого этой страницы:

- Введите текст: «**В этом разделе собраны ссылки на различные Web-страницы, которые мне показались интересными и полезными**».

Теперь создадим таблицу, в которой будет содержаться описание Интернет-ссылок. Таблица будет состоять из трех колонок. В первой колонке будут идти сами ссылки, во второй колонке – краткие описания страниц, а в третьей – различные комментарии.

- Щелкните мышью на кнопке **Добавить таблицу** (New Table) () панели инструментов программы. Появится меню выбора конфигурации таблицы.

В меню выбора конфигурации таблицы в схематическом виде представлена создаваемая вами таблица, каждый квадратик меню соответствует одной ячейке.

- Перемещая курсор мыши, добейтесь, чтобы в меню темным цветом было выделено три столбца и две строки.
- Щелкните левой кнопкой мыши. Будет создана таблица из трех столбцов и двух строк.

Теперь заполним таблицу содержимым:

- Щелкните мышью в левой верхней ячейке таблицы. Текстовый курсор переместится в эту ячейку.
- Введите заголовок первого столбца таблицы: «**Ссылки**».
- Перемещая нажатиями клавиши **Tab** текстовый курсор по ячейкам первой строки, введите остальные два заголовка таблицы: «**Краткое описание**» и «**Комментарии**».
- Нажмите еще раз клавишу **Tab**, текстовый курсор переместится в первую ячейку второй строки.

Вставим гиперссылку на Интернет-сайт:

- Выберите команду меню **Вставка** ♦ **Гиперссылка** (Insert ♦ Hyperlink). Появится диалог **Добавление гиперссылки** (Add hyperlink), Рис. 2.37.

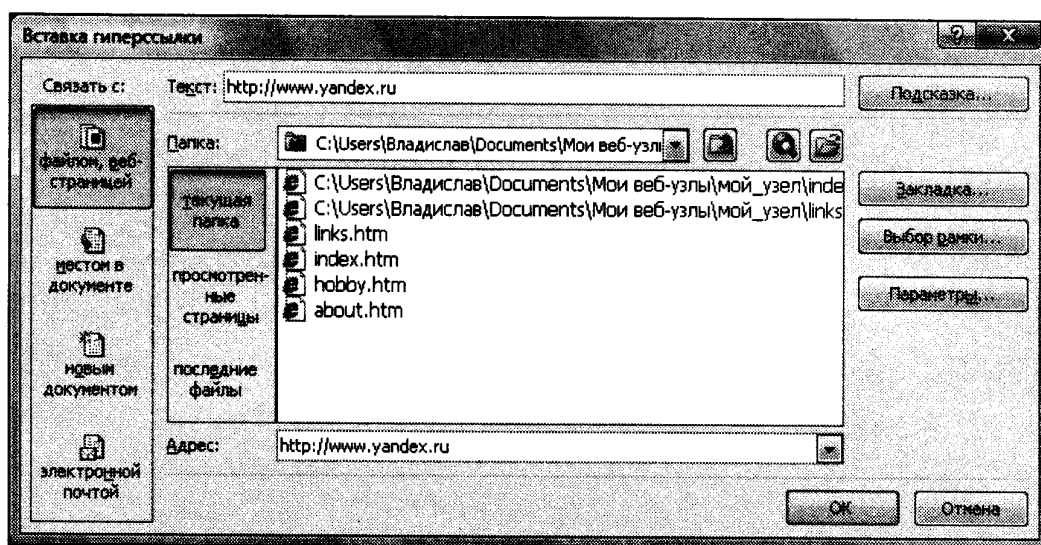


Рис. 2.37. Создание ссылки на удаленный web-узел

До этого вы создавали гиперссылки вместе с созданием самих документов, на которые ссылались. Теперь вам предстоит создать несколько ссылок на документы, которые не только существуют, но и располагаются вне вашего компьютера. Создадим первую такую ссылку.

- Щелкните мышью в поле ввода **Адрес** (Address). Введите в поле адрес сайта: **http://www.yandex.ru**.
- Щелкните мышью в поле ввода **Текст** (Text) и введите в него текст элемента привязки ссылки, **www.yandex.ru**.
- Щелкните мышью на кнопке **OK**, чтобы закрыть диалог и создать гиперссылку.

Заполним остальные два столбца строки таблицы:

- Нажмите клавишу **Tab**, текстовый курсор перейдет на вторую ячейку второй строки.

- ▶ Введите текст описания ссылки: **«Самая популярная российская поисковая система»**.
- ▶ Еще раз нажмите клавишу **Tab**, текстовый курсор перейдет на третью ячейку второй строки.
- ▶ Введите текст комментария: **«Может и не самая лучшая в мире, но отнюдь не самая плохая поисковая система»**.
- ▶ Снова нажмите клавишу **Tab**. В таблице будет создана третья строка. Текстовый курсор перейдет на первую ячейку третьей строки.

Аналогично первой гиперссылке следует ввести в таблицу ссылки на оставшиеся сайты. Первый: адрес – **http://www.mult.ru**, текст элемента привязки – **«www.mult.ru»**, краткое описание – **«Мультипликационная студия Олега Куваева»**, комментарий – **«Студия создателя Масяни.»**. И второй: адрес – **http://www.microsoft.com**, текст элемента привязки – **«www.microsoft.com»**, краткое описание – **«Сайт корпорации Microsoft.»**, комментарий – **«А это создатели Windows!»**.

Вот вы закончили создание и последней страницы вашего первого web-сайта. Полученная страница должна выглядеть, как на Рис. 2.38.

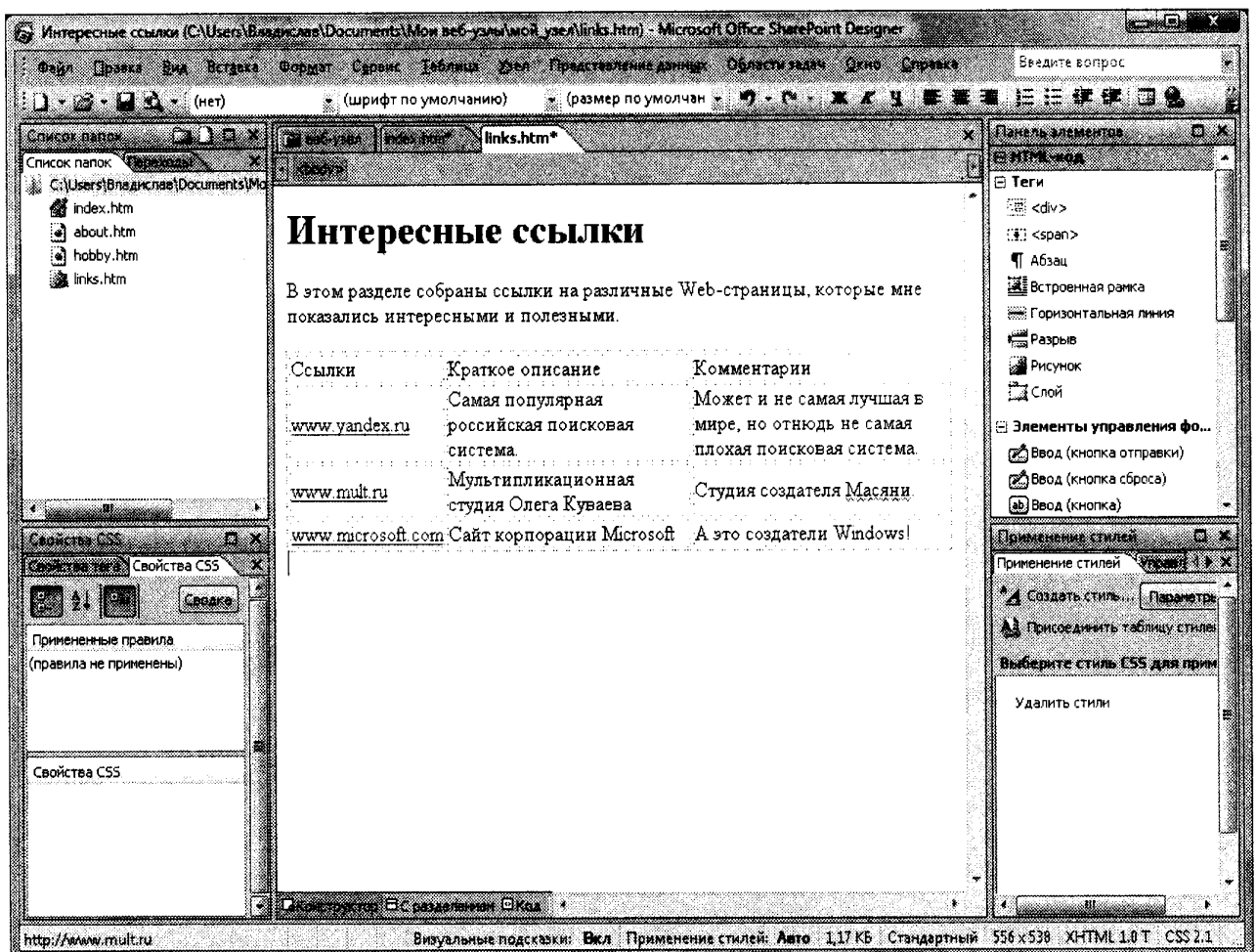



Рис. 2.38. Раздел домашней страницы «Интересные ссылки»

- Не забудьте сохранить проделанную работу, щелкнув мышью на кнопке  панели инструментов программы.

Теперь осталось добавить несколько финальных штрихов – и ваш web-сайт готов.

## Добавление навигационного меню на второстепенные страницы web-сайта

Некоторым недостатком структуры только что созданного вами сайта является то, что вы можете открывать второстепенные страницы сайта, только находясь на главной странице. Эту оплошность можно исправить, добавив на второстепенные страницы навигационное меню, в котором будут ссылки на все остальные разделы сайта.

Чтобы навигационное меню было отделено от основной части раздела, отделим его горизонтальной линией.

- Щелкните мышью в нижней части макета страницы.
- Выберите пункт меню **Вставка ♦ HTML-код ♦ Горизонтальная линия** (Insert ♦ HTML-code ♦ Horizontal line). Линия будет вставлена.

Теперь последовательно вставим гиперссылки на все разделы сайта.

- Выберите команду меню **Вставка ♦ Гиперссылка** (Insert ♦ Hyperlink). Появится диалог **Добавление гиперссылки** (Add hyperlink), представленный на Рис. 2.39.

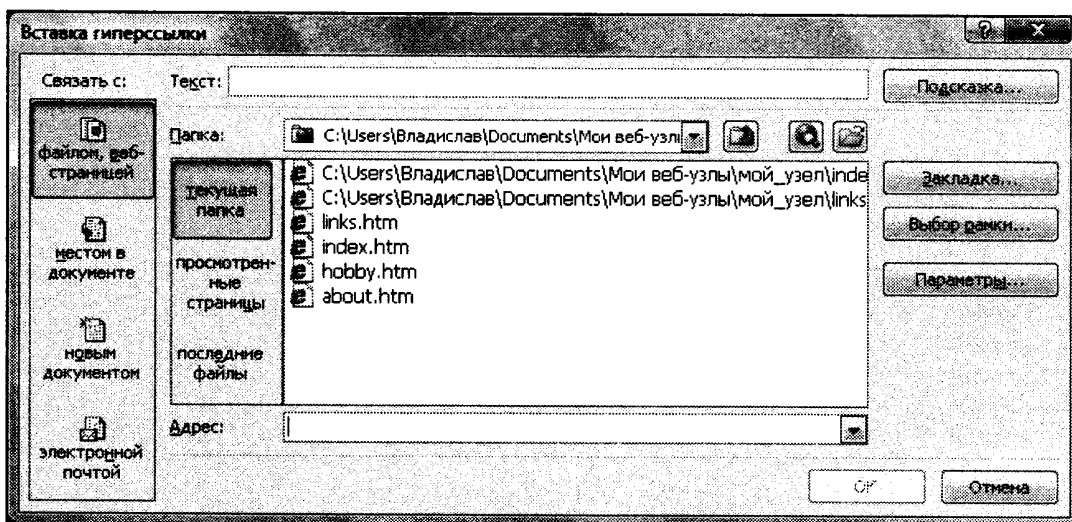


Рис. 2.39. Добавление ссылок на страницы вашего сайта

- Щелкните мышью на названии файла **index.html** в списке файлов **Папка** (Folder). В поле ввода **Адрес** (address) подставится название файла, а в поле ввода **Текст** (Text) автоматически подставится название главной страницы сайта.
- Щелкните мышью на кнопке **ОК**, чтобы создать гиперссылку и закрыть диалог.

Чтобы ссылки на разные страницы сайта не сливались друг с другом, отделим их вертикальными черточками.

- Нажмите клавишу **Пробел**, затем введите значок «|» и еще раз нажмите клавишу **Пробел**.

Таким же образом добавим гиперссылки на остальные страницы, тоже разделяя их вертикальными черточками. Напоминаем названия файлов остальных разделов: **about.htm**, **hobby.htm**, **links.htm**.

В результате макет страницы приобретет вид, показанный на Рис. 2.40.

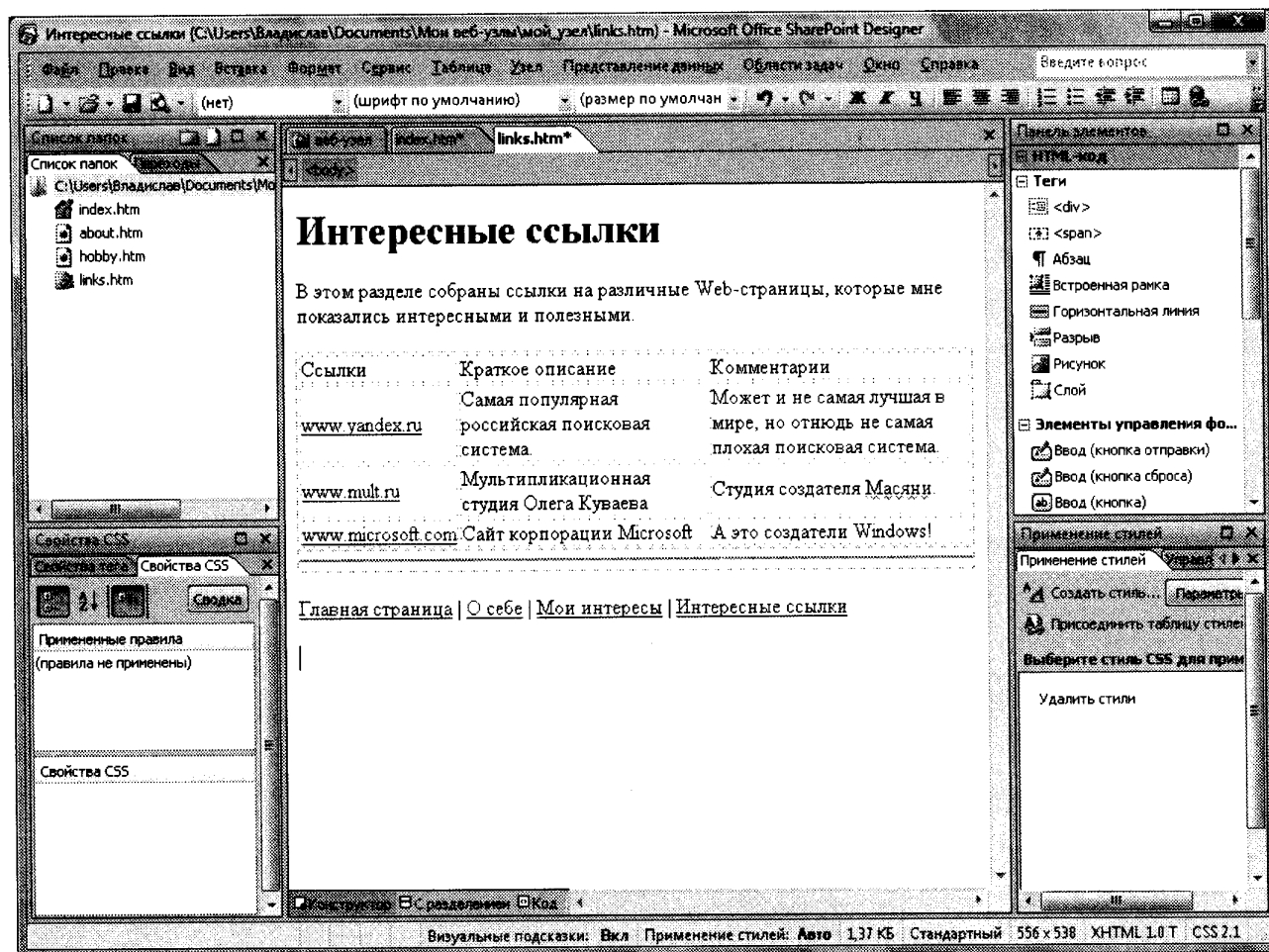


Рис. 2.40. Макет страницы сайта с добавленным навигационным меню

Скопируйте полученное навигационное меню в буфер обмена Windows.

- Переместите указатель мыши так, чтобы он находился чуть левее горизонтальной линии над навигационным меню.
- Нажмите левую кнопку мыши и, не отпуская ее, перемещайте указатель мыши так, чтобы горизонтальная линия и навигационное меню оказались выделенными.
- Отпустите левую кнопку мыши, навигационное меню выделено.
- Выберите команду меню **Правка ♦ Копировать** (Edit ♦ Copy). Меню будет скопировано в буфер обмена.

Теперь последовательно вставим навигационное меню на все второстепенные страницы сайта.

- Сделайте активной страницу «Интересы», щелкнув мышью на названии файла страницы **hobby.htm** в области заголовков документов.
- Щелкните мышью в нижней части макета страницы.
- Выберите команду меню **Правка ♦ Вставить** (Edit ♦ Insert). Меню будет вставлено на страницу.
- Сделайте активной страницу «Обо мне», щелкнув мышью на названии файла страницы **about.htm** в области заголовков документов.
- Щелкните мышью в нижней части макета страницы.
- Выберите команду меню **Правка ♦ Вставить** (Edit ♦ Insert). Меню будет вставлено на страницу.

Теперь навигационное меню есть на всех второстепенных страницах. Поскольку в меню стоят ссылки на все страницы сайта, то получается, что в меню каждого раздела одна гиперссылка ссылается на сам этот раздел. Эту несуразность желательно исправить. Можно просто удалить лишние ссылки вместе с элементами привязки, но лучше удалить ссылку, оставив текст ссылки.

- Сделайте активной страницу «Интересные ссылки», щелкнув мышью на названии файла страницы **links.htm** в области заголовков документов.
- Щелкните правой кнопкой мыши на гиперссылке **Интересные ссылки** навигационного меню. Откроется контекстное меню.
- Щелкните мышью на строке меню **Свойства гиперссылки** (Hyperlink properties). Откроется диалог **Изменение гиперссылки** (Change hyperlink) (Рис. 2.41).

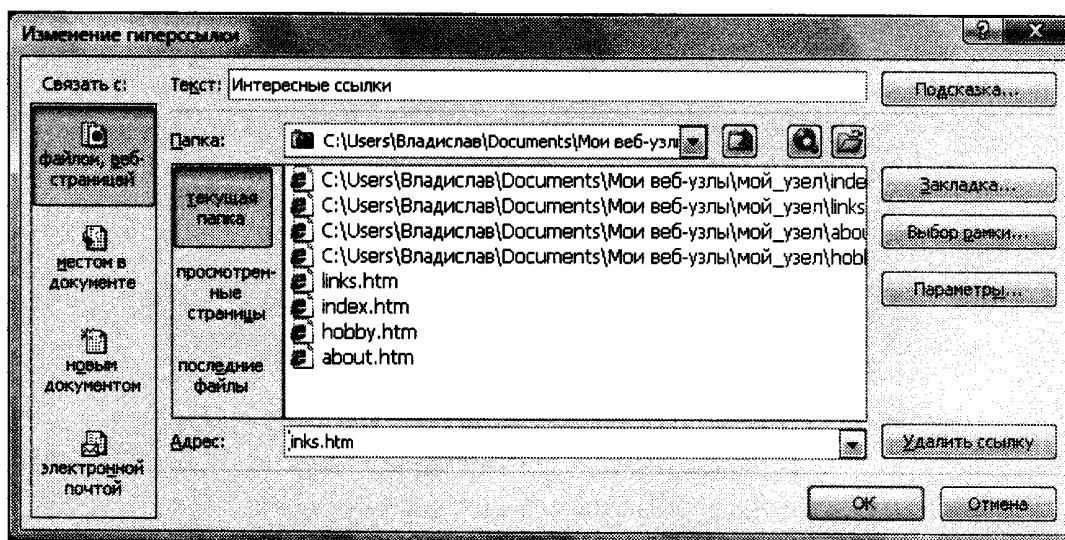


Рис. 2.41. Диалог **Изменение гиперссылки** (Change hyperlink)

- Щелкните мышью на кнопке **Удалить ссылку** (Delete link), чтобы удалить ссылку и закрыть диалог. В результате ссылка заменится на простой текст (Рис. 2.42).

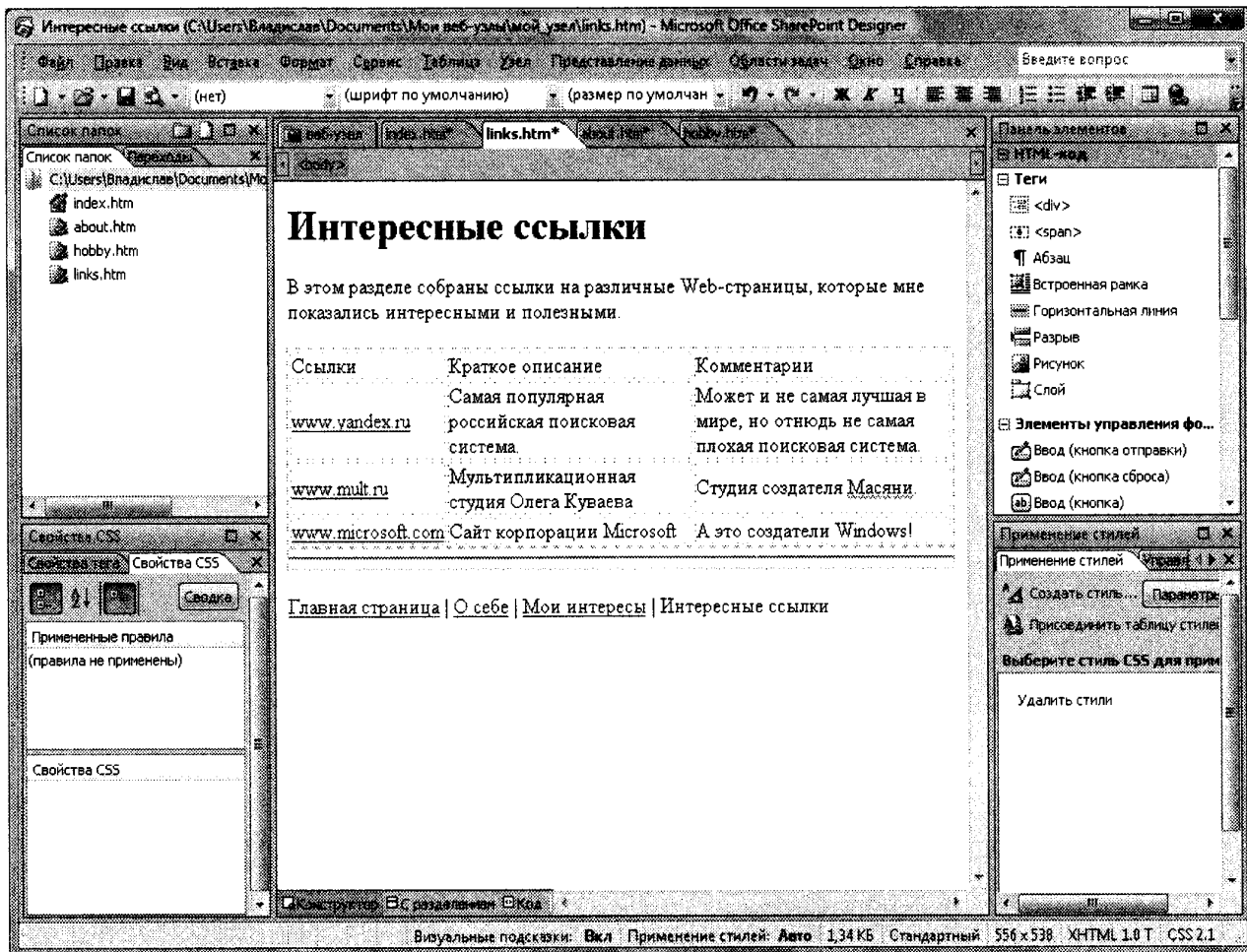


Рис. 2.42. Замена гиперссылки на обычный текст

Проделайте то же самое на других второстепенных страницах. Удалите из файла **about.htm** ссылку на раздел «**Обо мне**», а из файла **hobby.htm** ссылку на раздел «**Интересы**».

После этого ваш web-сайт будет полностью готов к использованию. Сохраним все изменения на страницах. Чтобы сохранить все измененные страницы разом:

- Выберите пункт меню **Файл ♦ Сохранить все** (File ♦ Save all), все измененные страницы будут сохранены.

Теперь можете приступить к просмотру полученного web-сайта.

## Просмотр web-сайта в браузере Internet Explorer

Откроем только что созданный web-сайт в браузере.

- Сделайте активной главную страницу сайта, щелкнув мышью на названии файла главной страницы **index.htm** в области заголовков документов.
- Выберите команду меню **Файл ♦ Просмотреть в обозревателе ♦ Microsoft Internet Explorer** (File ♦ Preview in web-browser ♦ Microsoft Internet Explorer). Главная страница сайта откроется в web-браузере Internet Explorer (Рис. 2.43).

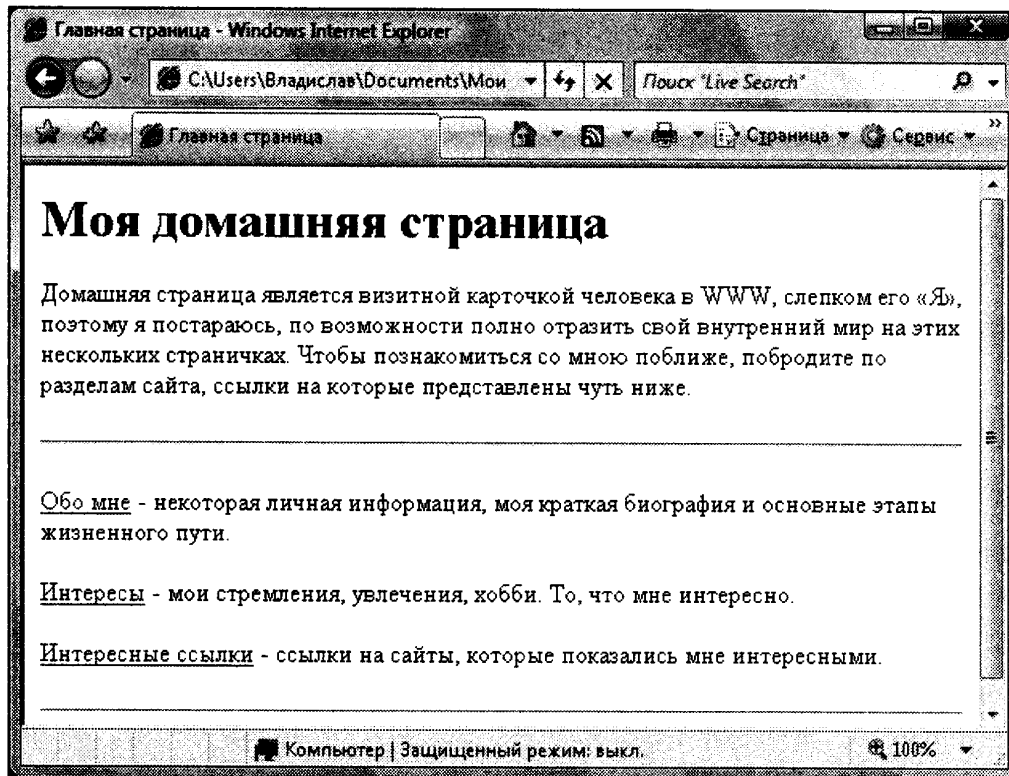


Рис. 2.43. Главная страница сайта в web-браузере Microsoft Internet Explorer

Щелкая мышью на ссылках, вы можете перейти к другим страницам сайта. Перейдем к разделу сайта «**Интересные ссылки**».

- Щелкните мышью на гиперссылке **Интересные ссылки**. Браузер откроет раздел сайта «**Интересные ссылки**» (Рис. 2.44).

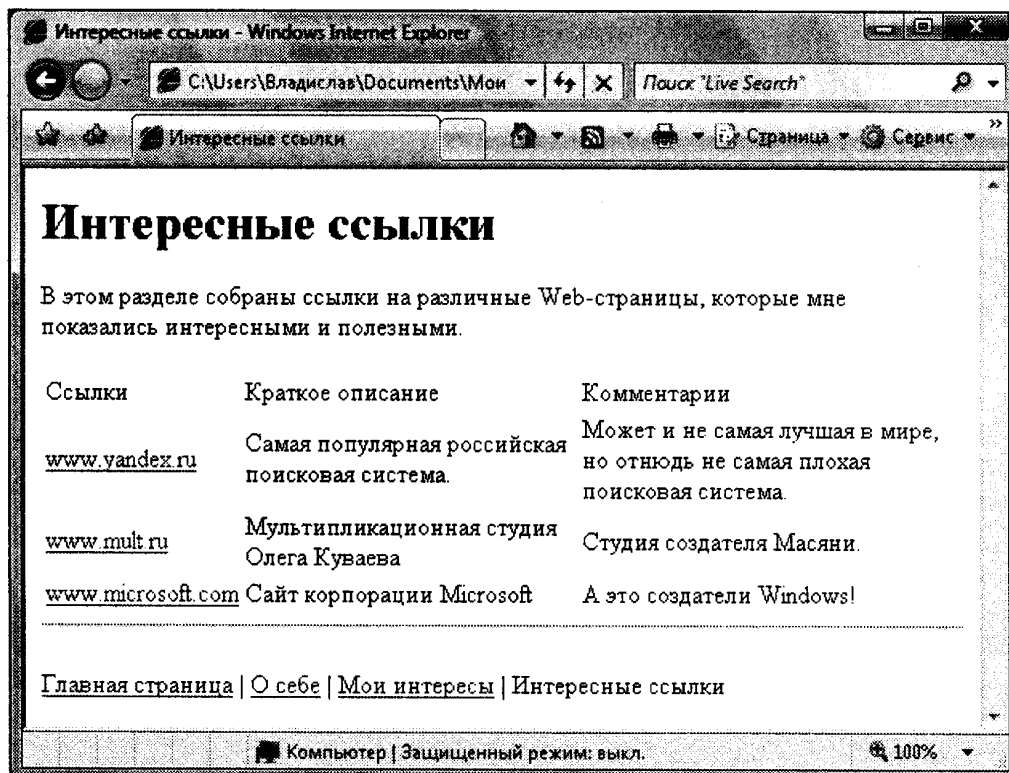


Рис. 2.44. Страница «Интересные ссылки» сайта



Если ваш компьютер подключен к сети Интернет, то, щелкнув по одной из гиперссылок в таблице на этой странице, можно открыть в web-браузере описанные в этой таблице сайты.

Чтобы открыть другой раздел сайта или перейти на главную страницу, вам достаточно щелкнуть мышью по одной из ссылок навигационного меню внизу web-страницы.

## Заключение

Поздравляем вас! Вы только что преодолели важнейший этап на пути к вершинам профессионального мастерства, создали свой первый web-сайт. Пусть он не слишком велик, всего четыре страницы, пусть он не блещет богатым оформлением и динамичной графикой, но тем не менее это уже полноценный сайт, который можно выложить в WWW и посещать его с помощью web-браузера. Элементы, с которыми вы познакомились в этой главе, являются базовыми для всех статических web-сайтов, и теперь, когда вы знаете их, для вас не составит большого труда научиться работать с более сложными элементами HTML и глубже изучить элементы, уже известные.

В следующей главе вы узнаете, как добавить на web-страницы графику и звук, познакомитесь с некоторыми особенностями оформления текстов и многими другими вещами, которые позволят вашим web-сайтам стать богаче и выразительнее, засиять всеми цветами радуги.

## ГЛАВА 3.

# Приемы, без которых нельзя обойтись



Вы познакомились с основными элементами web-страниц и уже создали свой первый web-сайт. Пришло время делать следующий шаг в освоении высот web-мастеринга. В этой главе вы узнаете о том, как оживить ваши страницы, добавив на них графические элементы, звуки и видео. Вы познакомитесь с основами табличной верстки web-страниц, что позволит вам создавать страницы довольно сложного строения. Кроме того, в этой главе будет рассказано об основных элементах оформления текстов web-страниц, что позволит сделать их более наглядными и удобочитаемыми.

## Графика на web-страницах

Язык HTML предоставляет возможность использовать в оформлении web-страниц графические элементы (Рис. 3.1). Это позволяет сделать страницы более наглядными, красочными и яркими. Ведь, как известно, одна хорошая иллюстрация может стоять нескольких страниц текста. Но, с другой стороны, чрезмерно увлекаться графикой не следует, все хорошо в меру.

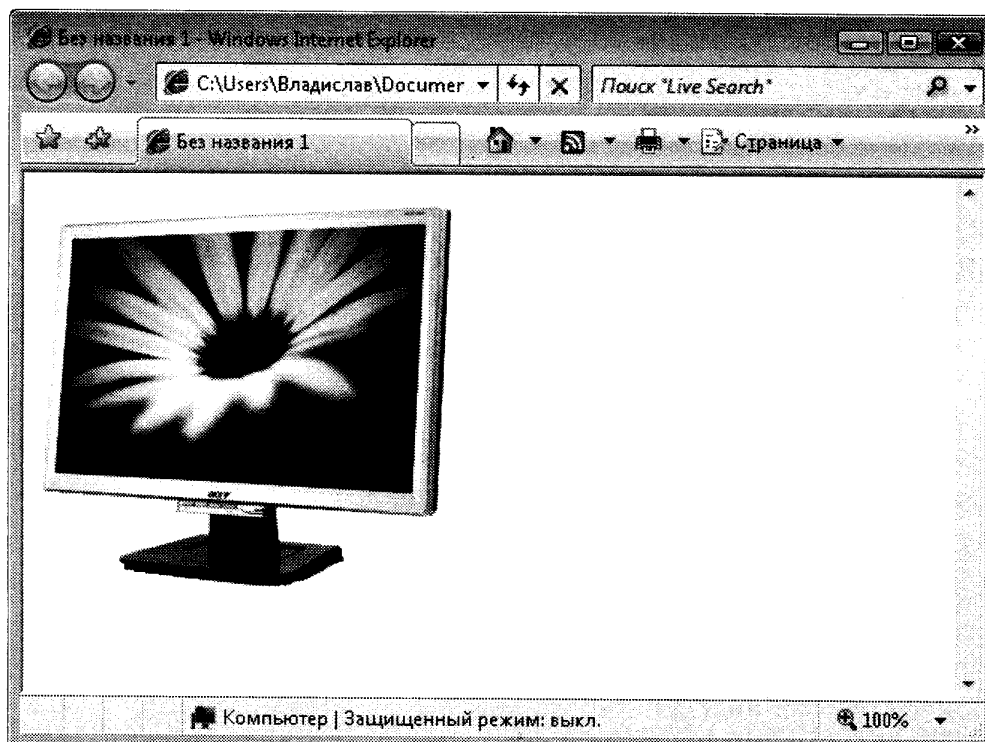


Рис. 3.1. Графика на web-странице

## Особенности графических файлов, используемых на web-страницах

На web-страницах можно размещать изображения из файлов форматов **GIF**, **JPEG** и **PNG**. Эти файлы обычно имеют расширения **.gif**, **.jpg** и **.png**. О принципиальных особенностях этих форматов и о том, как создавать такие файлы, подробно будет рассказано в следующей главе. Сейчас же мы лишь слегка коснемся общих моментов, важных для понимания материала данной главы.

Изображения в файлах всех перечисленных форматов хранятся в так называемом «растровом виде». Растровые изображения представляют собой множество точек разного цвета. Поскольку точки очень маленькие и близкорасположенные, то создается впечатление, будто это одна большая картинка. Ширина и высота растрового изображения определяются количеством точек в нем по горизонтали и по вертикали.

## Размещение графических элементов на web-страницах

В языке HTML картинки на web-страницу добавляются с помощью одиночного тега **<img>** и его атрибута **src**, с помощью которого задается адрес, по которому располагается картинка. Например, тег **** добавит на страницу картинку **face.jpg**, находящуюся по адресу **http://www.my\_site.com/** (Листинг 3.1).

*Листинг 3.1. HTML-документ, отображающий картинку на web-странице*

```
<html>
<head>
<title>Картинка</title>
</head>
<body>
 </BODY>
</html>
```

Если картинка по каким-либо причинам не смогла или не успела загрузиться, то на ее месте будет отображаться граница (Рис. 3.2). В этой рамке может содержаться текст, отражающий содержание рисунка. Этот текст задается с помощью атрибута **alt**. Текст, задаваемый этим атрибутом, называется альтернативным представлением картинки. Пользователь, который не может или не хочет познакомиться с самой картинкой, сможет прочитать текст, ее описывающий.

Рекомендуем вам ко всем изображениям, которые вы будете размещать на web-страницах, добавлять альтернативные описания, потому что достаточно велик процент пользователей, которые отключают отображение картинок в своих web-браузерах для уменьшения объема загружаемой информации. Кроме того, для людей с ослабленным

зрением и слепых практически единственным способом взаимодействия с WWW является использование «голосовых» браузеров, которые «озвучивают» содержимое web-страниц. Разумеется, картинки они озвучивать не могут, поэтому голосовые браузеры «читают» описания изображений, определяемые атрибутом **alt**.

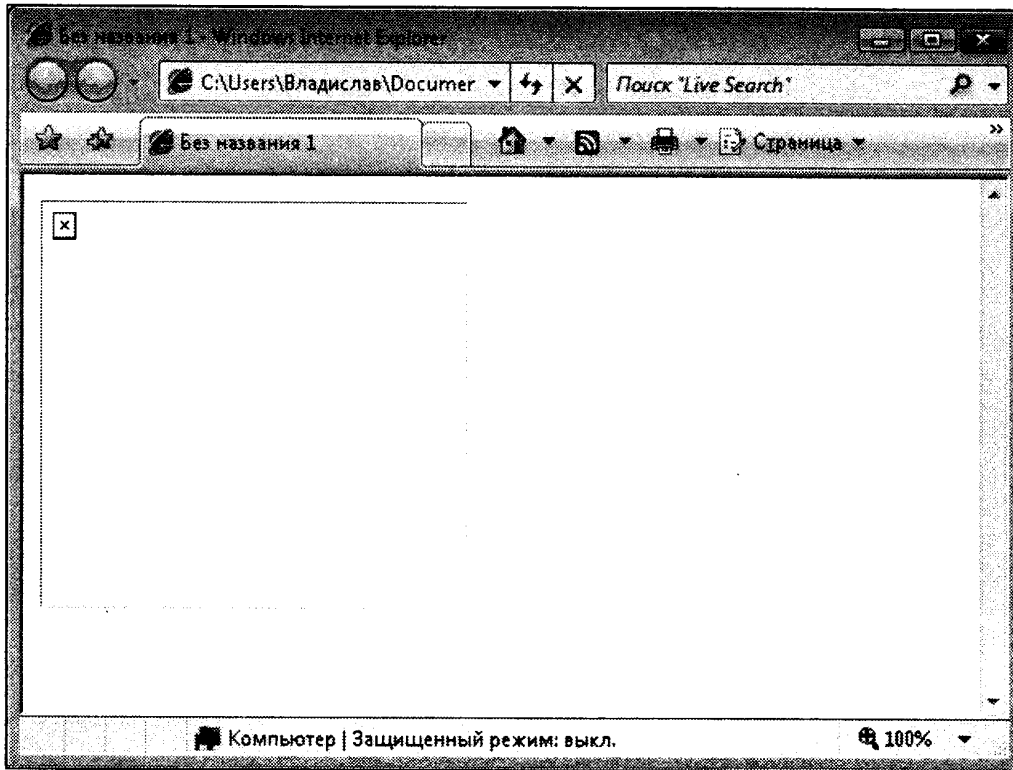
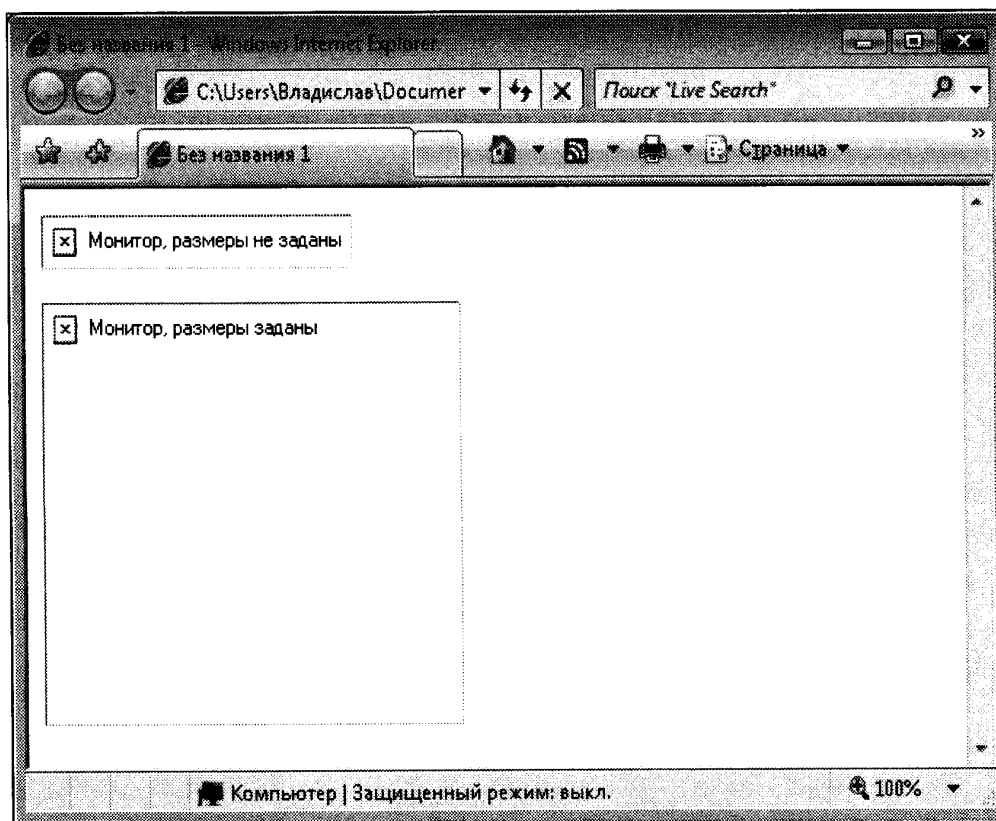


Рис. 3.2. Рамка на месте незагрузившегося изображения

Атрибуты **height** и **width** тега **<img>** позволяют указать браузеру размеры загружаемой картинки в точках или в процентах от всего размера web-страницы (Рис. 3.3). Атрибут **height** задает ее высоту, а **width** – ширину. Если размеры картинки с помощью этих атрибутов не задать, ничего страшного не произойдет, браузер все равно определит размеры изображения, когда будет его загружать, но, тем не менее, лучше указывать размеры картинки явно. Для этого есть несколько причин.

- ✓ Пока web-браузер не загрузит изображение на страницу, он не знает, какого оно размера. Поэтому, если изображение еще не загружено и его размеры явно не указаны, то граница будет занимать минимально возможное место. При этом может пострадать оформление всей страницы. Если же размеры картинки заранее определены, то граница займет место, равное размерам изображения.
- ✓ Вторая причина, по которой всегда следует использовать атрибуты **height** и **width**, является следствием первой. Представьте такую ситуацию: загрузка страницы идет медленно, а это достаточно частое явление, и размеры изображений заранее не указаны. В этом случае, по мере загрузки картинок на страницу, границы, окружающие место изображений, будут раздвигаться под размер картинок, и все элементы оформления страницы будут перемещаться друг относительно друга. Этот процесс выглядит не слишком привлекательно. Поэтому лучше таких случаев избегать и указывать размеры изображений.



*Рис. 3.3. Границы незагруженных изображений в случаях, когда размер рисунка не определен и задан явно*

Когда вы добавляете изображение на web-страницу при помощи программы **SharePoint Designer 2007**, она автоматически указывает в тегах HTML их размеры, позволяя вам не задумываться об этой проблеме.

Если при помощи атрибутов **height** и **width** указать размеры изображения, не равные его фактическим размерам, то при размещении на страницы картинка будет отмасштабирована в соответствии с указанными размерами. С использованием этой особенности тега **<img>** связаны некоторые приемы оформления web-страниц, но в обычной практике рекомендуется этого избегать.

Изображения можно размещать прямо в тексте (Рис. 3.4). При этом можно выбрать, как именно будет выравниваться текст относительно картинки. Согласно нынешним стандартам Консорциума WWW, для выравнивания текста рекомендуется использовать CSS. Так, для вертикального выравнивания используется свойство **vertical-align**. Наиболее часто применяемыми значениями этого свойства являются **top**, **middle** и **bottom** (Рис. 3.5).

Несмотря на то, что при работе в программе **SharePoint Designer 2007** CSS вам нет необходимости изучать синтаксис CSS, общими представлениями о нем все-таки желательно обладать. Свойства и их аргументы разделены двоеточием, различные свойства – точкой с запятой. Перед названием класса должна стоять точка, а все его свойства заключаются в фигурные скобки (Листинг 3.2).

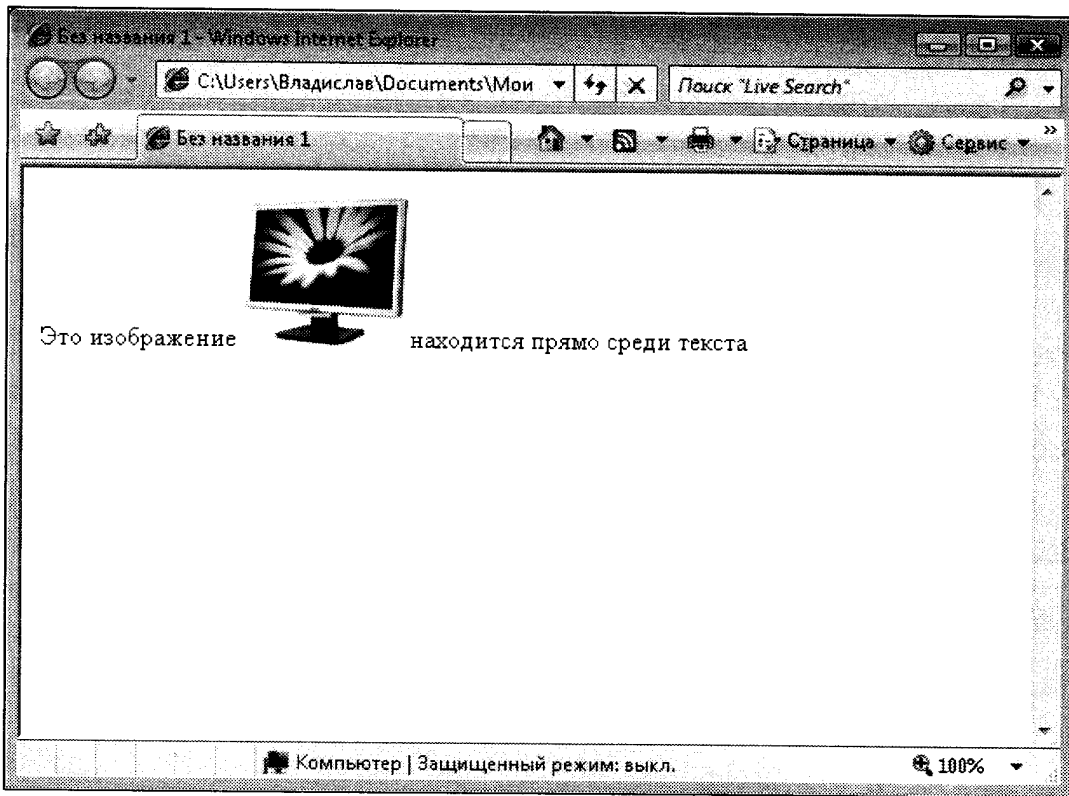


Рис. 3.4. Размещение изображения среди текста

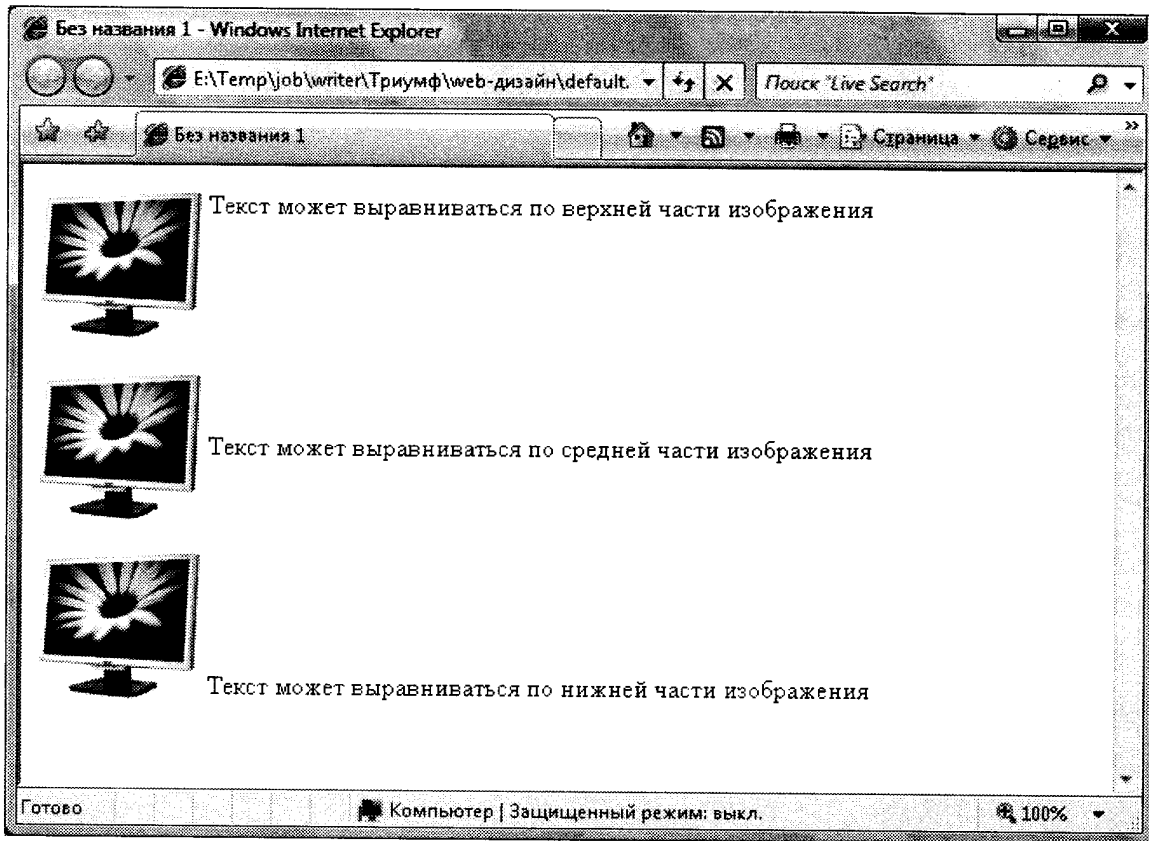


Рис. 3.5. Варианты выравнивания текста относительно изображения

*Листинг 3.2. Способы выравнивания текста относительно изображения*

```
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»»
<html xmlns=»http://www.w3.org/1999/xhtml»»

<head>
<meta http-equiv=»Content-Language» content=»en-us» />
<meta http-equiv=»Content-Type» content=»text/html; charset=utf-
8» />
<title>Выравнивание текста относительно изображения</title>
<style type=»text/css»»
.style1 {
    vertical-align: top;
}
.style2 {
    vertical-align: middle;
}
.style3 {
    vertical-align: bottom;
}
</style>
</head>

<body>
<img src=»file:///E:/Temp/job/writer/Триумф/web-дизайн/exam-
ple_small.jpg» class=»style1» width=»100» height=»94» /><span
lang=»ru»»
Текст может выравниваться по верхней части изображения<br />
<br />
</span>

<img src=»file:///E:/Temp/job/writer/Триумф/web-дизайн/example_
small.jpg» class=»style2» width=»100» height=»94» />
```

```
<span lang="ru">Текст может выравниваться по средней части
изображения<br />
<br />
</span>

<span lang="ru">Текст может выравниваться по нижней части изоб-
ражения</span>
</body>
</html>
```

Также, можно сделать так, чтобы картинка не находилась в строке наряду с текстом, а находилась слева или справа от текста, который будет ее обтекать (Рис. 3.6). Обтекание текстом задается свойством **float**. Значение этого свойства **left** соответствует расположению картинки слева от текста, а значение **right** – справа от него (Листинг 3.3).

*Листинг 3.3. Способы выравнивания текста относительно изображения*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Language" content="en-us" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />

<title>Без названия 1</title>

</head>

<body>
```



```
Также, можно сделать так, чтобы картинка не
находилась в строке наряду с текстом, а нахо-дилась слева
или справа от текста, который будет ее обтекать (Рис. 3.6).
Обтекание текстом задается свойством float. Значение этого
свойства left соответствует расположению картинки слева от
текста, а значение right - справа от него (Листинг 3.3).<br />
```

```
<br />
```

```
Также, можно сделать так, чтобы картинка не
находилась в строке наряду с текстом, а нахо-дилась слева
или справа от текста, который будет ее обтекать (Рис. 3.6).
Обтекание текстом задается свойством float. Значение этого
свойства left соответствует расположению картинки слева от
текста, а значение right - справа от него (Листинг 3.3).
```

```
</body>
```

```
</html>
```

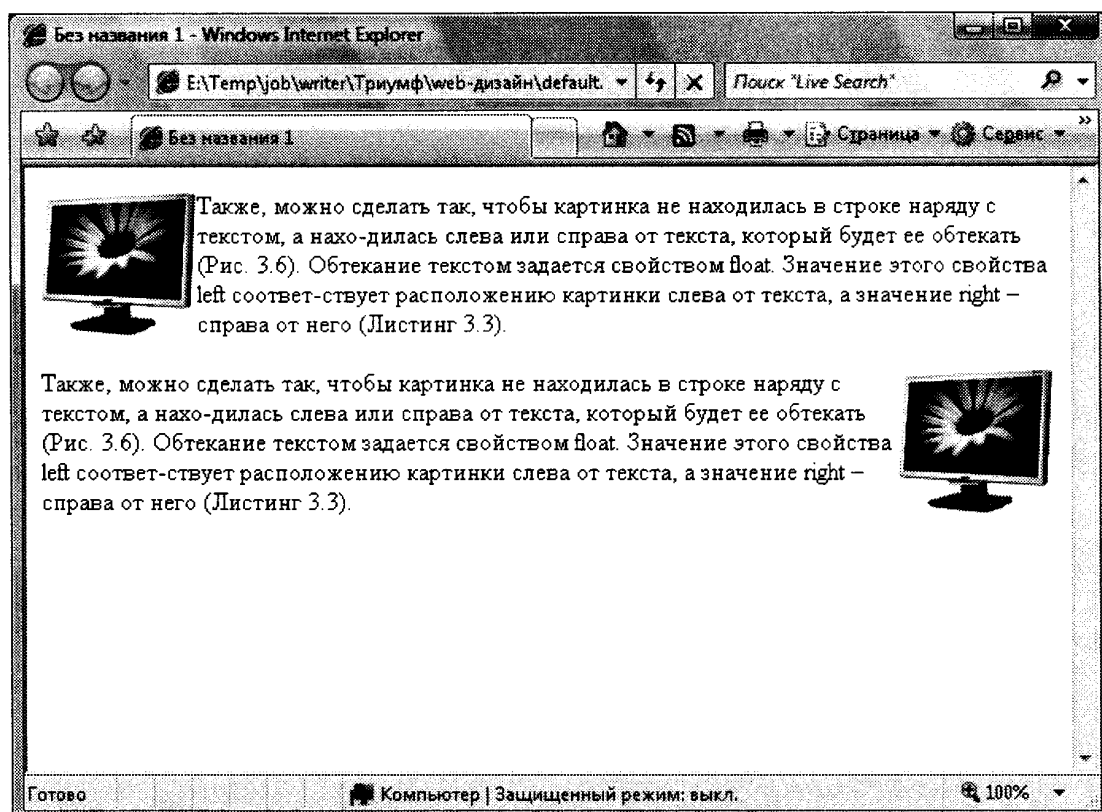
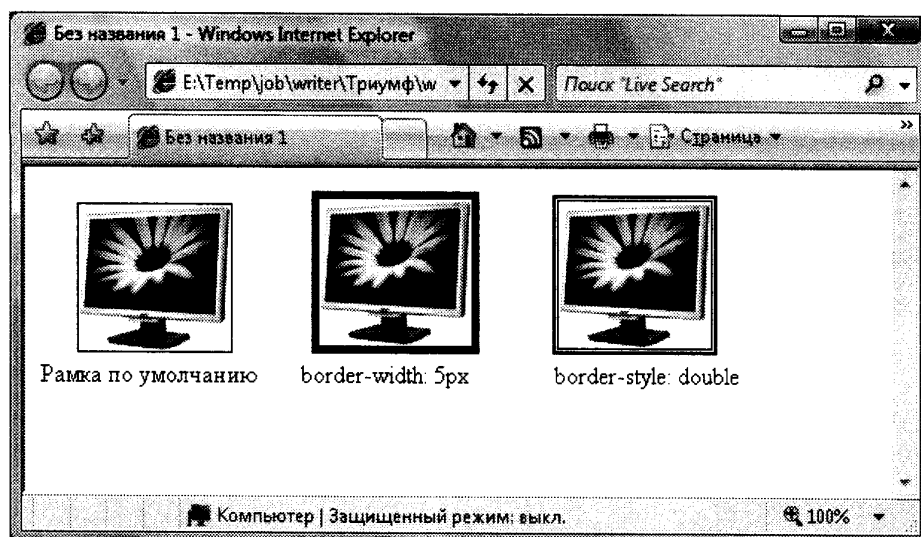


Рис. 3.6. Обтекание изображения текстом

Изображения на странице можно окружать рамкой (Рис. 3.7), параметры которой задаются двумя свойствами: **border-style** и **border-width** (Листинг 3.4). Первое из них определяет тип линии, которая может быть пунктирной, двойной и т.д., а второе – толщину линии.

*Листинг 3.4. Отображение границ картинок*

```
<style type=»text/css»>
.style1 {
    border-style: solid;
    border-width: 1px;
}
.style2 {
    border-style: solid;
    border-width: 5px;
}
.style3 {
    border-style: double;
}
</style>
```



*Рис. 3.7. Рамка вокруг изображения*

Интервалы между объектами задаются при помощи свойств **margin-left**, **margin-right**, **margin-top** и **margin-bottom**. Эти свойства задают интервал соответственно слева, справа, сверху и снизу (Листинг 3.5). Ниже вы можете видеть пример, иллюстрирующий применение данных свойств (Рис. 3.8).



```

gif» width=»100» height=»94» class=»style3» /><img src=»black.
gif» width=»100» height=»94» class=»style4» /><br />
<img src=»black.gif» width=»100» height=»94» class=»style5» />
margin-top:3px;
margin-bottom:3px<br />
<img src=»black.gif» width=»100» height=»94» class=»style6» />
margin-top:10px;
margin-bottom:10px
</body>
</html>

```

Обратите внимание, что в приведенном листинге свойства перечислены в одну строку. Такая форма записи позволяет уместить на экране большее количество классов одновременно, однако немного затрудняет восприятие.

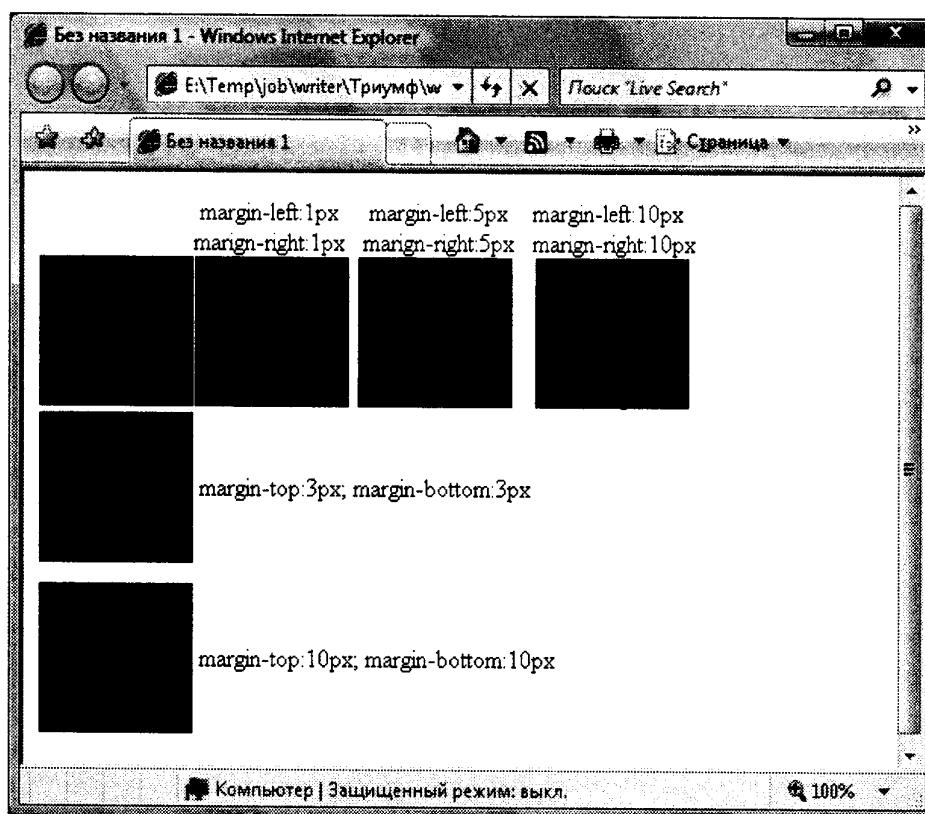


Рис. 3.8. Интервалы между объектами


## Добавление графики на HTML-страницу в программе SharePoint Designer 2007

Теперь, когда рассмотрены основные особенности размещения графики на страницах HTML, разберемся с тонкостями работы с картинками в программе SharePoint Designer 2007.

Чтобы добавить изображение на текущую web-страницу в программе SharePoint Designer 2007:

- Щелкните мышью в той части макета web-страницы, куда вы хотите добавить изображение.
- Выберите команду меню **Вставка ♦ Рисунок ♦ Из файла** (Insert ♦ Image ♦ From file),

либо

- Нажмите кнопку  на панели инструментов программы. На экране появится стандартный диалог выбора файла операционной системы.
- Перейдите к каталогу, в котором содержится графический файл, который вы хотите добавить на страницу, и щелкните мышью на его названии.
- В открывшемся диалоге **Свойства для специальных возможностей** (Properties for Custom Capabilities) щелкните мышью на кнопке **ОК**.
- Щелкните мышью на кнопке **Вставить** (Insert). Диалог будет закрыт, а изображение добавлено на web-страницу.

## Настройка параметров изображения

Чтобы настроить параметры изображения, размещенного на web-странице:

- Щелкните правой кнопкой мыши на изображении. Появится контекстное меню.
- Щелкните мышью на пункте **Свойства рисунка** (Image properties) контекстного меню. Появится диалог **Свойства рисунка** (Image properties).

Во вкладках этого диалога собраны все настройки изображений. Рассмотрим их по очереди.

Вкладка **Общие** (General) диалога **Свойства рисунка** (Image properties) содержит общие настройки свойств рисунка (Рис. 3.9). В поле ввода **Замещающий текст** (Replacing Text) группы настроек **Специальные возможности** (Accessibility Options) вы можете ввести текст, который будет назначен атрибуту **alt** тега `<img>`. Этот текст будет замещать изображение в том случае, если оно не будет загружено.

Щелкнув мышью на вкладке **Вид** (Appearance) (Рис. 3.10), вы перейдете к настройкам, касающимся отображения картинок на странице.

- ✓ Миниатюры **нет** (none), **слева** (left) и **справа** (right) группы настроек **Обтекание** (Flow) позволяют выбрать способ, которым текст будет обтекать изображение. Чтобы выбрать подходящий способ обтекания, щелкните по нему мышью.
- ✓ В группе элементов управления **Положение** (Position) находятся элементы управления расположением картинки на странице. В открывающемся списке **Выравнивание** (Align) можно выбрать способ выравнивания текста отно-

сительно изображения. Пункты этого списка соответствуют различным аргументам свойства **vertical-align** (Вертикальное выравнивание). Выбрав пункт **По верхнему краю** (Top) вы установите выравнивание по верхней границе изображения, пункты **По середине** (middle) и **По нижнему краю** (bottom) позволяют настроить выравнивание текста по середине и по нижней границе соответственно. Если вы выберете пункт меню **по умолчанию** (default), то изображение будет упорядочиваться так, как задано в настройках браузера или настройках всей страницы.

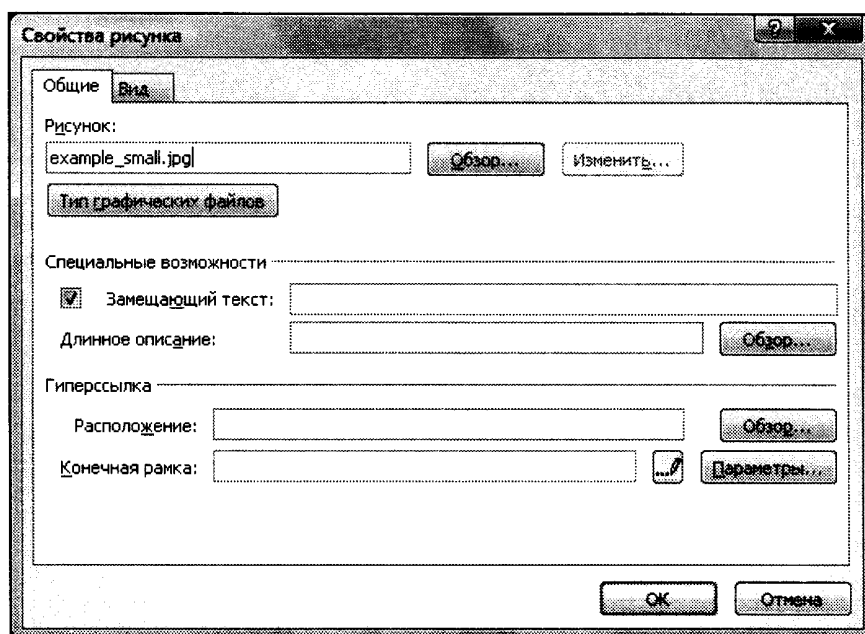


Рис. 3.9. Вкладка **Общие** (General) диалога **Свойства рисунка** (Image properties)

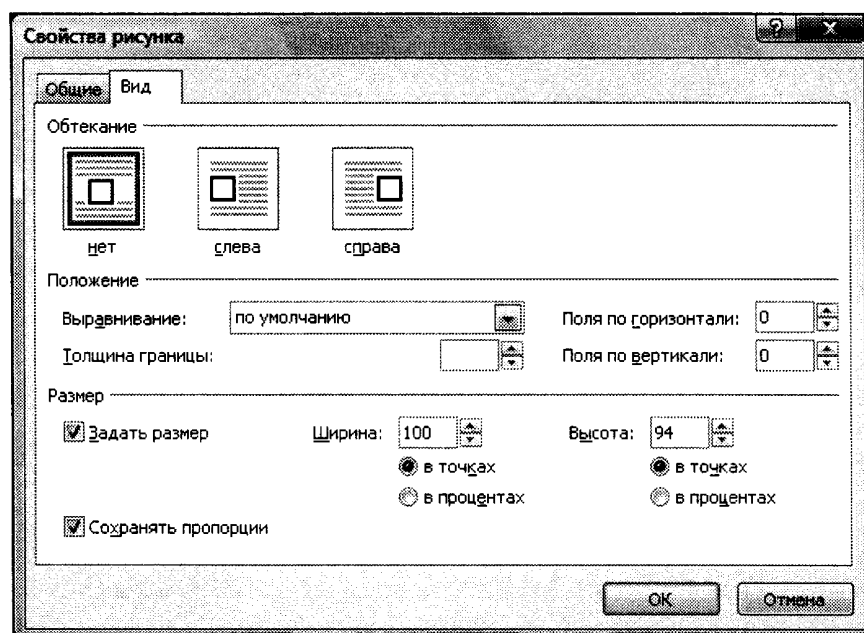


Рис. 3.10. Вкладка **Вид** (Appearance) диалога **Свойства рисунка** (Image properties)

- ✓ В поле ввода со счетчиком **Толщина границы** (border thickness) вы можете задать толщину рамки, окружающей картинку.
- ✓ Поле ввода со счетчиком **Интервал по горизонтали** (Horizontal space) задает расстояние между изображением и соседними элементами страницы по горизонтали, а поле **Интервал по вертикали** (Vertical space) – по вертикали.
- ✓ В группу элементов управления **Размер** (Size) сведены настройки, касающиеся размеров картинки на странице. При установленном флажке **Задать размер** (Set size) размеры картинки будут явно указываться в теге `<img>`. В полях ввода со счетчиком **Ширина** (Width) и **Высота** (Height) автоматически прописываются настоящие ширина и высота картинки, но при необходимости их можно изменить. Если установлен флажок **Сохранять пропорции** (Constrain proportions), то изменение одного из этих полей будет приводить к пропорциональному изменению другого поля так, чтобы пропорции изображения не менялись с изменением его размера. Установив переключатель под полем ввода **Ширина** (Width) или **Высота** (Height) в положение **в процентах** (percent), вы сможете задавать размеры изображения в процентах от общего размера окна web-браузера. Установив же переключатель в положение **в точках** (points), вы сможете указать размеры картинки в точках.

Сделав все необходимые настройки, щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Еще одним способом применения графики на web-страницах является использование изображений в качестве фона страницы (Рис. 3.11). Фоновый рисунок можно задать web-странице при помощи свойства **background-image:** тега `<body>` (Листинг 3.6).

*Листинг 3.6.*

*Интервал между изображением и соседними элементами страницы*

```
<html>
<!DOCTYPE html PUBLIC «-//W3C//DTD XHTML 1.0 Transitional//EN»
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»
<html xmlns=»http://www.w3.org/1999/xhtml»>

<head>
<meta http-equiv=»Content-Type» content=»text/html; charset=utf-
8» />
<title>Без названия 1</title>
</head>

<body style=»background-image: url('background.png')»>
```

```
<h1>Содержимое страницы располагается перед фоновым изображением ;</h1>
```

```
<p>Содержимое страницы располагается перед фоновым изображением. Содержимое
```

```
страницы располагается перед фоновым изображением. Содержимое страницы
```

```
располагается перед фоновым изображением. Содержимое страницы располагается
```

```
перед фоновым изображением.</p>
```

```
</body>
```

```
</html>
```

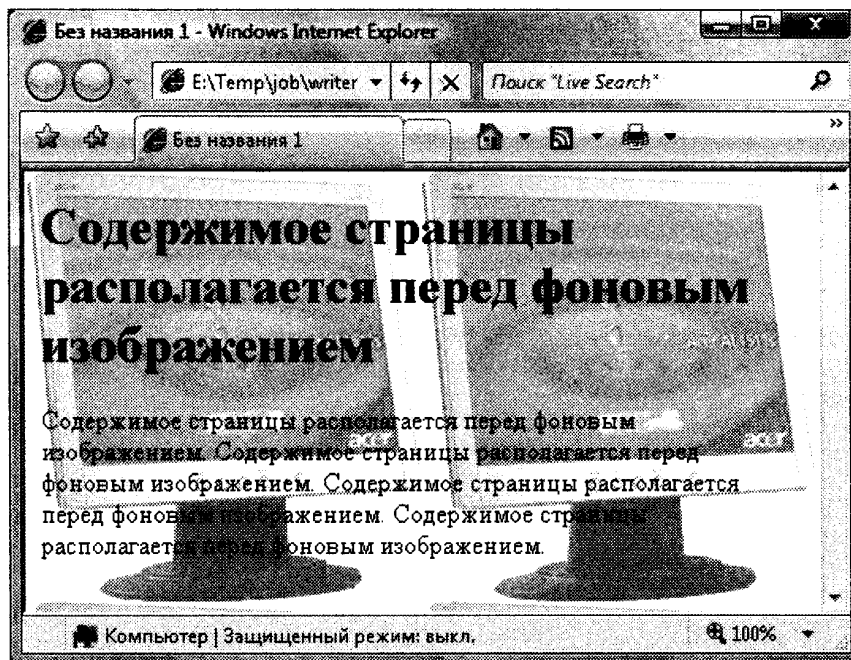


Рис. 3.11. Фоновое изображение на странице

Чтобы добавить фоновое изображение на текущую web-страницу в программе SharePoint Designer 2007:

- Выберите команду меню **Формат ♦ Фон** (Format ♦ Background). Откроется вкладка **Форматирование** (Format) диалога **Свойства страницы** (Page properties), представленная на Рис. 3.12.
- Установите флажок **Фоновый рисунок** (Background picture). Этим вы активизируете возможность размещения картинки в качестве фона страницы.
- Щелкните мышью в поле ввода и введите адрес, по которому располагается картинка, которую вы хотите сделать фоновой.



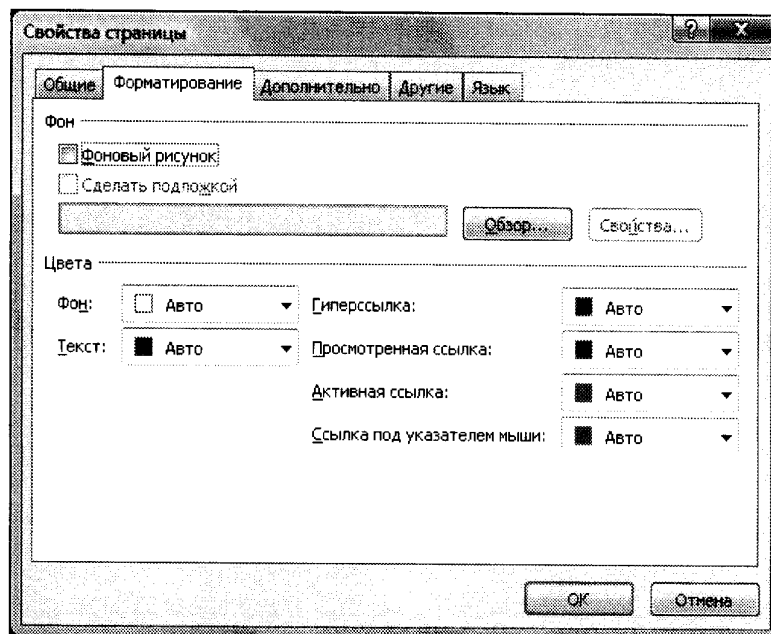


Рис. 3.12. Вкладка **Форматирование** (*Format*) диалога **Свойства страницы** (*Page properties*)

Либо

- Щелкните мышью на кнопке **Обзор** (Browse), откроется стандартный диалог выбора файла операционной системы.
- Перейдите к каталогу, где находится изображение, которое вы хотите сделать фоновым, и щелкните мышью на его названии.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы принять сделанный выбор и закрыть диалог **Выбрать фоновый рисунок** (Choose background picture).
- Выбрав фоновое изображение, щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог **Свойства страницы** (Page properties).

При использовании фоновых изображений на web-страницах не забывайте об удобстве восприятия информации пользователем. Старайтесь, чтобы содержимое страницы легко воспринималось на выбранном фоне.

Наряду с текстами, изображения можно использовать в качестве элементов привязки для гиперссылок. Чтобы добавить гиперссылку к изображению на web-странице:

- Щелкните правой кнопкой мыши на изображении, к которому вы хотите добавить гиперссылку. Появится контекстное меню.
- Щелкните мышью на пункте **Гиперссылка** (Hyperlink) контекстного меню. Появится диалог **Добавление гиперссылки** (Add hyperlink), Рис. 3.13.

О создании новых гиперссылок вкратце рассказывалось во второй главе данной книги. Подробнее этот вопрос будет рассмотрен в конце данной главы.

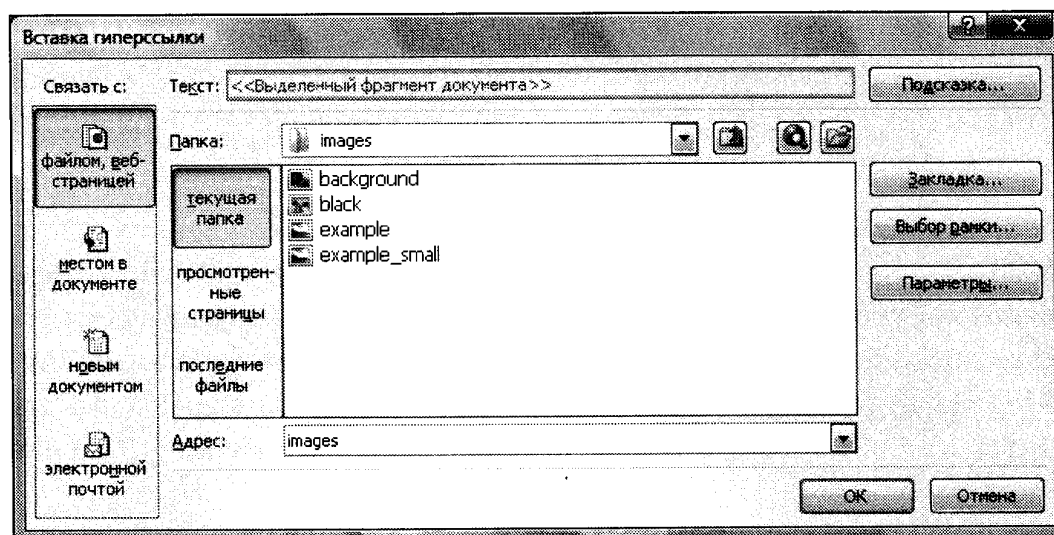


Рис. 3.13. Диалог **Добавление гиперссылки** (Add hyperlink)

## Использование звука и видео на web-страницах

Чтобы сделать ваши web-страницы более динамичными и привлекательными, можно разнообразить их оформление звуковыми и видеовставками. Главное, что необходимо принимать во внимание, – это то, что звук и видео обычно занимают достаточно большой объем, что негативно скажется на времени загрузки вашей web-страницы пользователями, у которых медленное подключение к Интернету. Кроме того, для воспроизведения звука и видео web-браузер пользователя должен быть оснащен специальными модулями для работы с теми или иными форматами файлов. Пользуясь некоторыми наиболее популярными форматами, вы можете быть практически уверены, что пользователь сможет насладиться красотами вашей web-страницы в полной мере. Поддержка же менее популярных форматов браузером пользователя не гарантируется. И хотя не составляет большого труда скачать дополнительные модули для воспроизведения того или иного типа файлов, совсем необязательно, что ради просмотра пары клипов с вашего сайта пользователь будет устанавливать какие-либо модули или программы на свой компьютер.

### Форматы звуковых файлов

Для размещения на web-страницах можно использовать звуковые файлы двух разных видов. Первый вид – оцифрованный звук. В таких файлах звук записывается в виде последовательной записи уровня громкости звучания. Уровень громкости записывается несколько десятков тысяч раз в секунду. Преимуществом таких звуковых файлов является то, что в них может быть записан абсолютно любой звук – музыка, человеческая речь, различные шумы, и звучать такие файлы будут примерно одинаково на любом компьютере, оборудованном аудиосистемой. Недостатком оцифрованного звука является большой объем, занимаемый такими файлами.

Самым простым форматом оцифрованного звука является формат WAV. Поддержка этого формата есть практически везде. Но главным недостатком этого формата является то, что данные в нем хранятся без сжатия, занимая очень много места. Файлы этого формата имеют расширение **.wav**.

Для размещения на web-страницах большого объема звука удобнее использовать формат MP3. Этот формат характерен тем, что звуковые данные в таком файле упаковываются с учетом особенностей восприятия звука человеческим ухом. Это позволяет уменьшать размеры файла до 10 раз по сравнению в форматом WAV. Файлы формата MP3 имеют расширение **.mp3**. Также достаточно популярен для размещения звука в Интернете другой формат сжатого оцифрованного звука – Real Audio. Эти файлы имеют расширение **.ra** или **.ram**.

Другим видом звуковых компьютерных файлов являются MIDI-файлы. В эти файлы записывается подобие нотных партитур музыкальных инструментов, на основе которых воспроизводят музыкальную последовательность специальные MIDI-синтезаторы, входящие в состав любой звуковой карты. Файлы такого типа занимают очень мало места. Музыкальное произведение длительностью несколько минут может занимать всего пару десятков килобайт, но звучание таких файлов обычно очень «искусственное» и в них невозможно записать звуки, для которых нет подходящих музыкальных инструментов. Файлы MIDI имеют расширение **.mid** или **.rmi**.

Самыми популярными форматами видеофайлов в сети Интернет являются формат QuickTime, файлы которого имеют расширение **.mov**, формат RealMedia, файлы которого имеют расширение **.rm**, и файлы MPEG с расширением **.mpg** или **.mpeg**.

## Добавление фонового звука на web-страницу

Звуковой файл можно назначить в качестве фонового сопровождения web-страницы. Для этих целей можно использовать файлы WAV, MIDI и Real Audio. Звуковые файлы добавляются на web-страницу с помощью тега **<bgsound>** в заголовке HTML-документа.

Добавить фоновый звук на web-страницу в программе SharePoint Designer 2007 можно с помощью диалога **Свойства страницы** (Page properties), Рис. 3.14.

- ▶ Чтобы открыть диалог **Свойства страницы** (Page properties), выберите команду меню **Файл ♦ Свойства** (File properties).

В группу элементов управления **Фоновый звук** (Background sound) собраны настройки, управляющие воспроизведением фонового звука.

Выберите звуковой файл, который будет фоном вашей страницы:

- ▶ Щелкните мышью в поле ввода **Расположение** (Location) и введите адрес, по которому располагается звуковой файл, который вы хотите сделать фоновым.

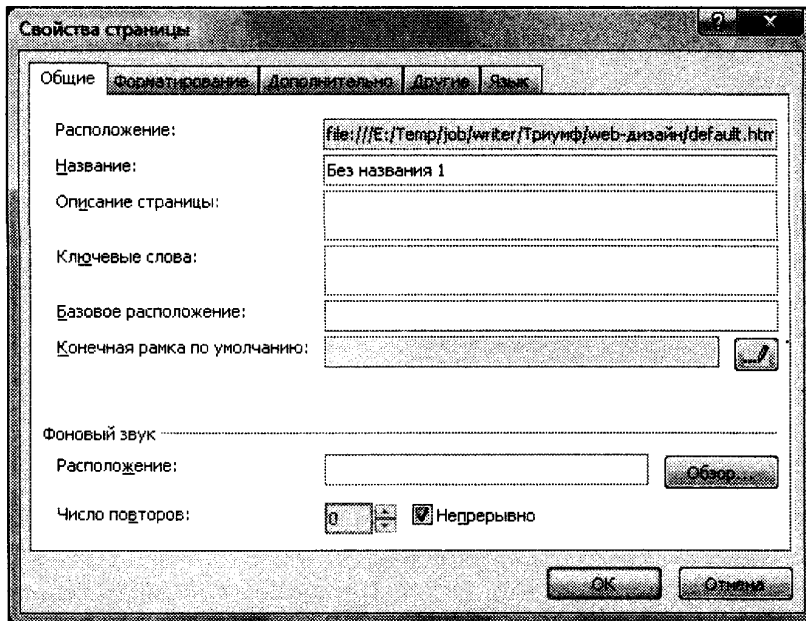


Рис. 3.14. Диалог **Свойства страницы** (Page properties)

Либо

- Щелкните мышью на кнопке **Обзор** (Browse), откроется стандартный диалог открытия файла операционной системы.
- Перейдите к каталогу, где находится звуковой файл, который вы хотите сделать фоновым, и щелкните мышью на его названии.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы принять сделанный выбор и закрыть диалог **Фоновый звук** (Background sound).

Если установлен флажок **Непрерывно** (Continuous), то фоновый звук будет непрерывно воспроизводиться, пока страница будет открыта. Если этот флажок сбросить, то в поле ввода со счетчиком **Число повторов** (Repeats number) вы можете указать, сколько раз будет воспроизведен фоновый звук после открытия страницы.

- После настройки всех параметров фонового звука щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог **Свойства страницы** (Page properties).

## Добавление звука и видео на web-страницу

Звуковые и видеофайлы можно размещать также на самой web-странице (Рис. 3.15). При условии, что к web-браузеру пользователя подключены дополнительные модули по воспроизведению нужного типа файлов, перед пользователем предстанет некое подобие видео/музыкального плеера с элементами управления: перемотка, пауза/воспроизведение, регулятор громкости.

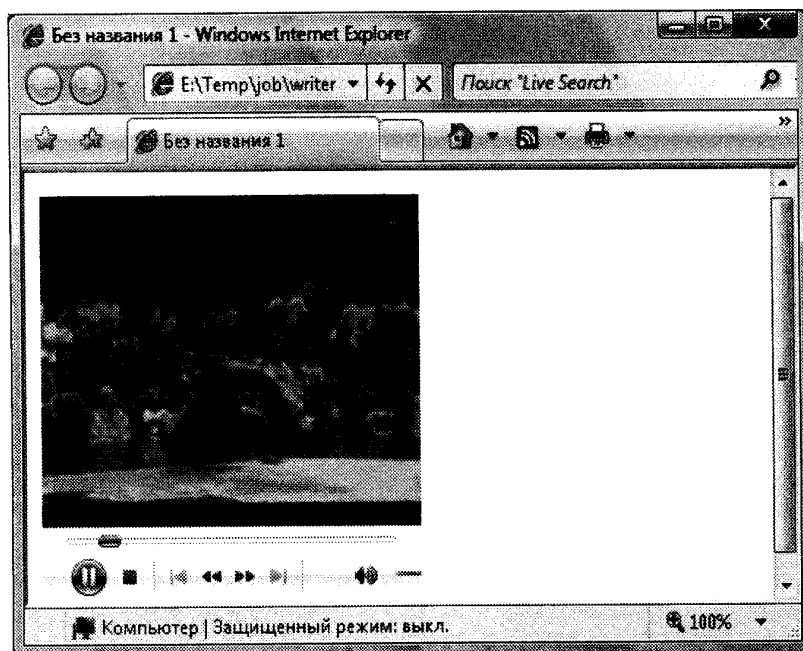


Рис. 3.15. Размещение звуковых и видеофайлов на web-странице

Чтобы добавить звуковой или видеофайл на web-страницу:

- Выберите пункт меню **Вставка ♦ Веб-компонент** (Insert Web-component). Откроется диалог **Вставка компонента веб-узла** (Web-component insertion), Рис. 3.16.

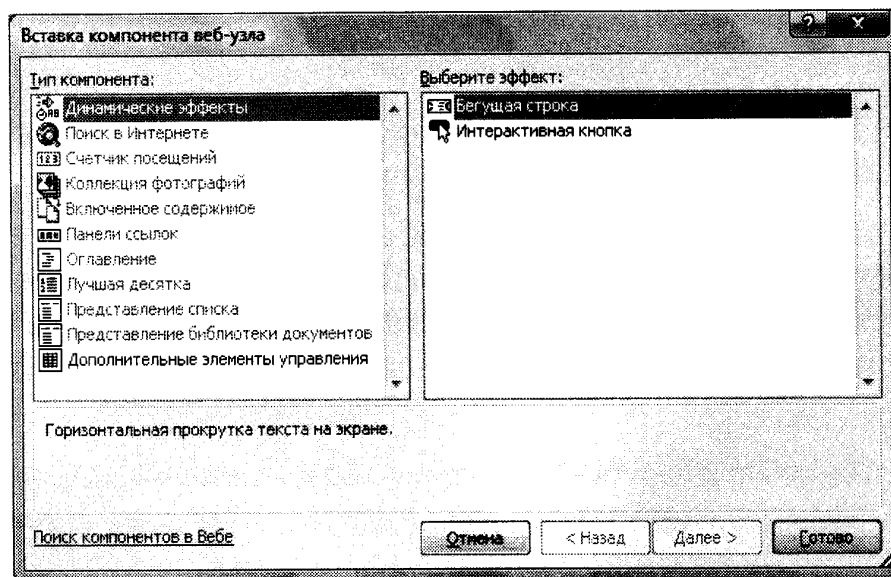


Рис. 3.16. Диалог **Вставка компонента веб-узла** (Web-component insertion)

- В списке **Тип компонента** (Component type) щелкните мышью на пункте **Дополнительные элементы** (Additional elements).
- В списке **Выберите элемент управления** (Choose control element) щелкните мышью на пункте **Подключаемый модуль** (Plug-in module).
- Щелкните мышью на кнопке **Готово** (Finish), чтобы закрыть текущий диалог и открыть диалог **Свойства подключаемого модуля** (Plug-in module properties), представленный на Рис. 3.17.

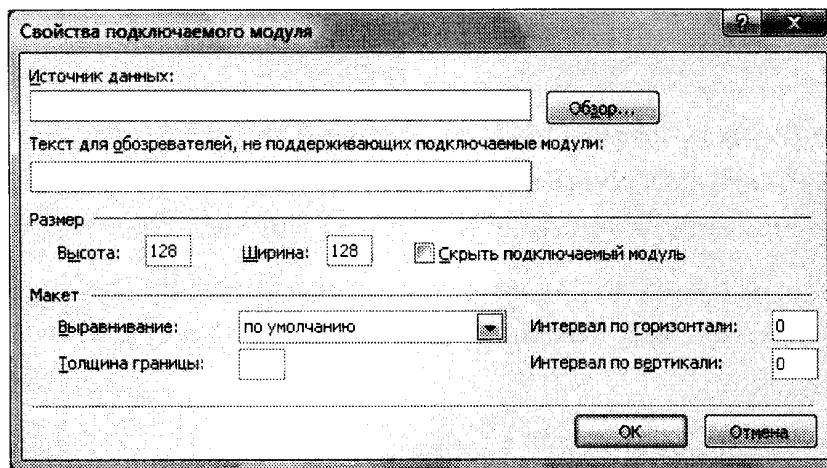


Рис. 3.17. Диалог **Свойства подключаемого модуля**  
(*Plug-in module properties*)

В появившемся диалоге настраиваются все параметры воспроизведения подключаемого файла.

Выберите файл, который будете размещать на web-странице:

- Щелкните мышью в поле ввода **Источник данных** (Data source) и введите адрес, по которому располагается файл,

либо

- Щелкните мышью на кнопке **Обзор** (Browse), откроется стандартный диалог выбора файла операционной системы.
- Перейдите к каталогу, в котором находится искомый файл.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы принять сделанный выбор и закрыть диалог **Выбор источника данных подключаемого модуля** (Choose data source for plug-in module).

В поле ввода **Текст для обозревателей, не поддерживающих подключаемые модули** (Text for browsers, that don't support plug-in modules) вы можете ввести текст, который будет отображаться вместо звукового или видеофайла, в том случае, если браузер пользователя не поддерживает внешние модули, необходимые для воспроизведения файла.

В полях ввода **Ширина** (Width) и **Высота** (Height) группы настроек **Размер** (Size) вы можете задать размер, который примет плеер звука или видео на вашей web-странице.

В группе элементов управления **Макет** (Prototype) находятся элементы управления расположением картинки на странице. В открывающемся списке **Выравнивание** (Align) можно выбрать способ выравнивания текста относительно изображения:

- ✓ Выбрав пункт **по левому краю** (left), вы установите расположение картинки слева от обтекающего ее текста.

- ✓ Если вы выберете пункт **по правому краю** (right), картинка будет располагаться справа от текста.
- ✓ Пункты **по верхнему краю** (top), **по середине** (middle) и **по нижнему краю** (bottom) позволяют настроить выравнивание текста по верхней границе изображения, по его середине и по нижней границе, соответственно.
- ✓ Если вы выберете пункт меню **по умолчанию** (default), то изображение будет упорядочиваться так, как задано в настройках браузера или настройках всей страницы.

В поле ввода со счетчиком **Толщина границы** (Border thickness) вы можете задать толщину рамки, окружающей картинку.

Поле ввода со счетчиком **Интервал по горизонтали** (Horizontal space) задает расстояние между изображением и соседними элементами страницы по горизонтали, а поле **Интервал по вертикали** (Vertical space) – по вертикали.

- Произведя все необходимые настройки, щелкните мышью на кнопке **ОК**, чтобы закрыть диалог и разместить на странице видео/звук (Рис. 3.18).

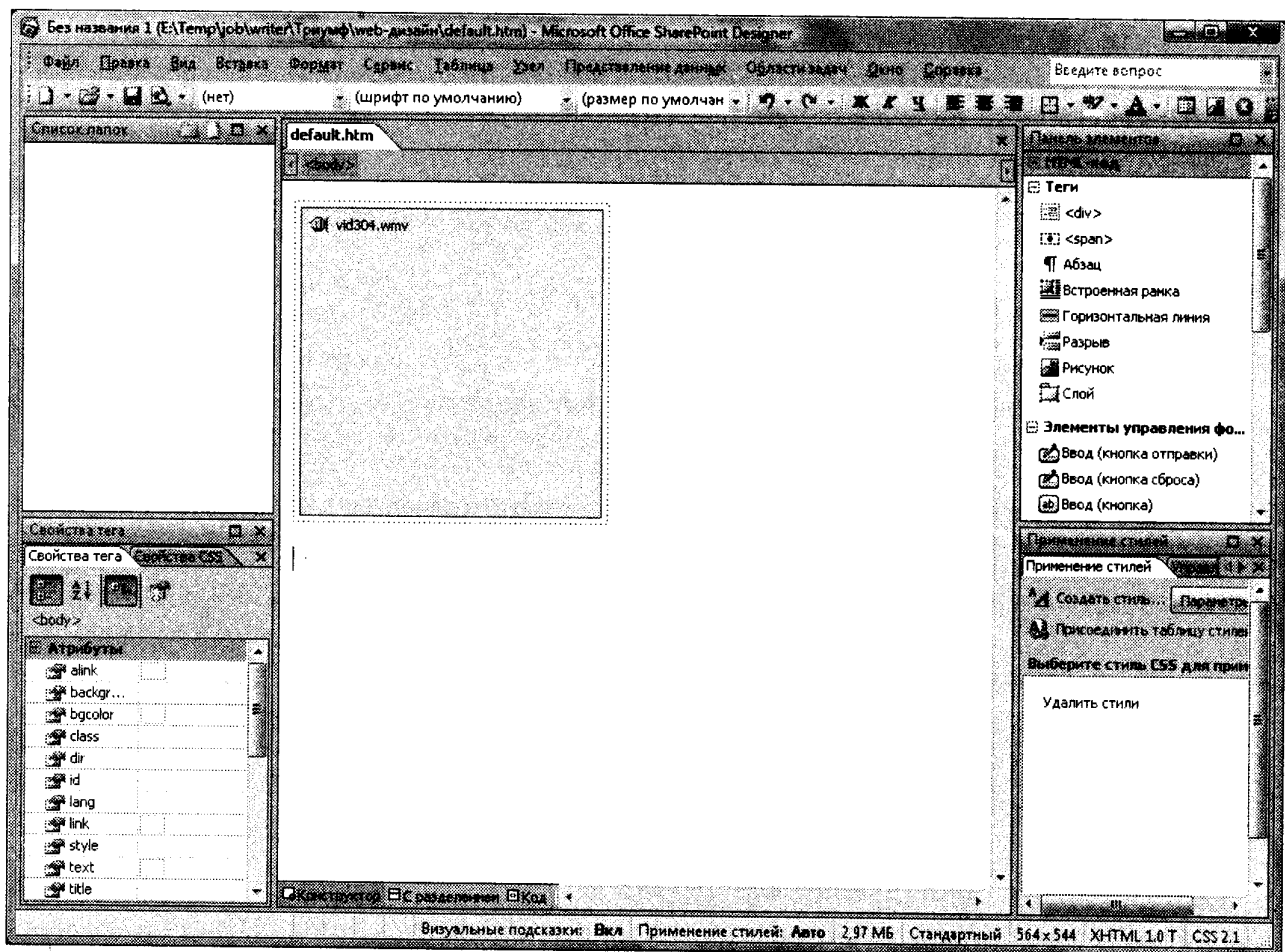


Рис. 3.18. Размещение файла видео на макете страницы

После этого, когда вы будете в очередной раз сохранять свой web-узел, откроется диалог **Сохранение внедренных файлов** (Save embedded files), Рис. 3.19.

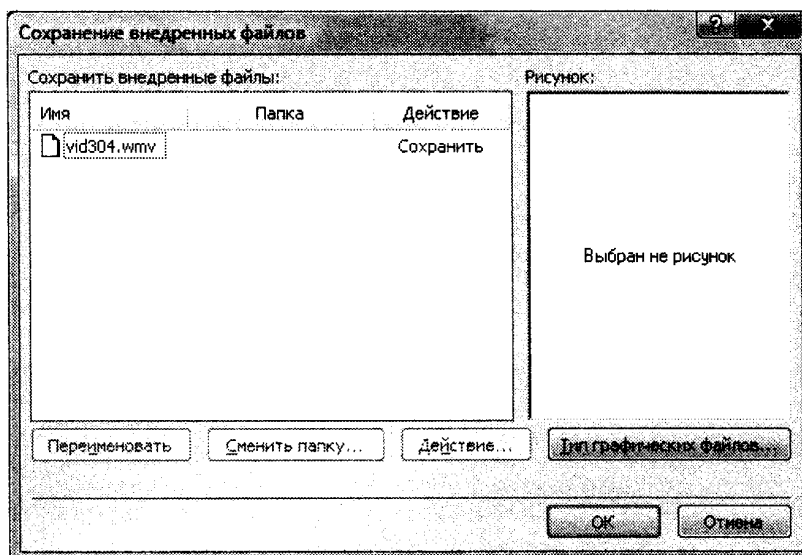


Рис. 3.19. Диалог **Сохранение внедренных файлов** (Save embedded files)

- Щелкните в диалоге **Сохранение внедренных файлов** (Save embedded files) мышью на кнопке **ОК**, чтобы сохранить все звуковые/видеофайлы в каталоге вашего web-сайта.

## Карты ссылок

Для создания красивых и наглядных меню навигации по разделам сайта можно использовать карты ссылок. Карта ссылок – это изображение, разделенное на несколько областей; щелкая мышью на разных областях карты ссылок, пользователь будет переходить по разным ссылкам.


Рассмотрим в качестве примера использования карт ссылок следующую ситуацию: вы создаете сайт, посвященный космическому кораблю собственной конструкции. На главной странице вы размещаете рисунок корабля, щелкая на различных частях которого, посетитель страницы может перейти на разделы сайта, посвященные различным модулям корабля (Рис. 3.20).

Области на карте ссылок бывают трех различных видов: прямоугольные, круглые и многоугольные. Новые области к карте ссылок добавляются с помощью панели инструментов **Рисунки** (Pictures).

- Если этой панели инструментов на экране нет, выберите пункт меню **Вид ♦ Панели Инструментов ♦ Рисунки** (View ♦ Toolbars ♦ Pictures), Рис. 3.21.

Чтобы добавить новую область к карте ссылок, следует щелкнуть мышью на картинке, к которой добавляете область карты ссылок.

Чтобы добавить круглую область на карту ссылок:

- Щелкните мышью на кнопке  панели инструментов **Рисунки** (Pictures).
- Переместите указатель мыши в ту часть картинки, где будет располагаться середина будущей области.



- Нажмите левую кнопку мыши и, не отпуская ее, перемещайте указатель мыши. Круг, появившийся под указателем мыши и изменяющий свои размеры при перемещении мыши, отображает размер будущей области карты ссылок.
- Когда окружность примет подходящий размер, отпустите левую кнопку мыши. Откроется диалог **Вставка гиперссылки** (Insert hyperlink).
- Создайте гиперссылку и щелкните мышью на кнопке **ОК**, чтобы закрыть диалог и завершить создание области гиперссылки.

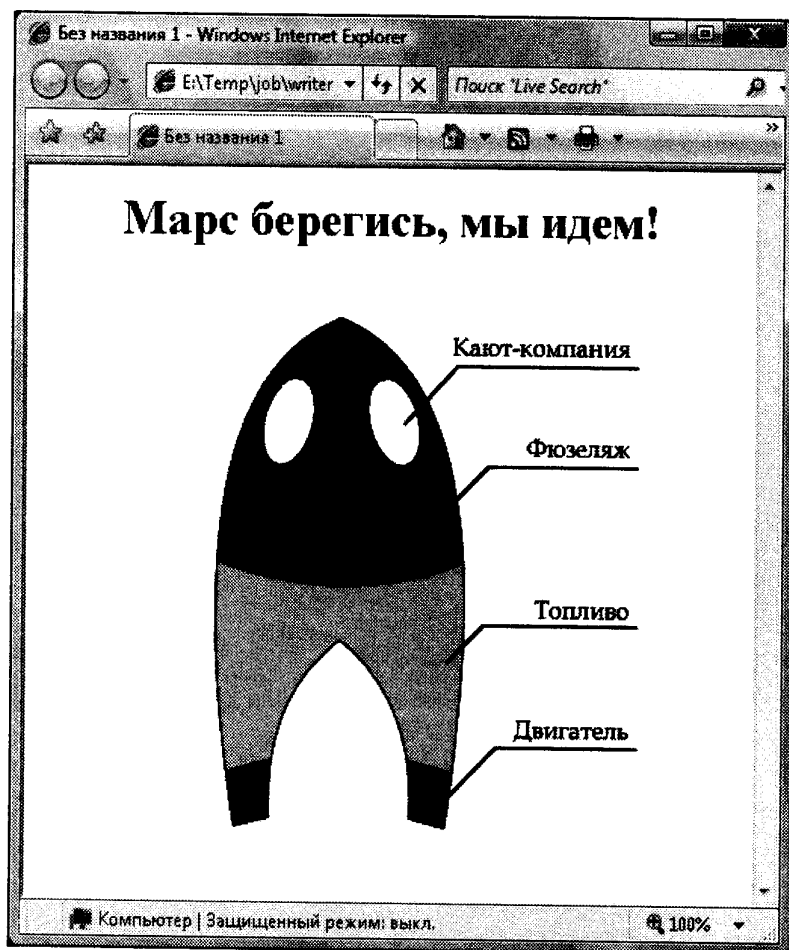


Рис. 3.20. Карта ссылок




Рис. 3.21. Панель инструментов **Рисунки** (Pictures)

Чтобы добавить многоугольную область на карту ссылок:

- Щелкните мышью на кнопке  панели инструментов **Рисунки** (Pictures).
- Переместите указатель мыши в ту часть картинки, где будет располагаться один из углов будущей области.

- Нажмите левую кнопку мыши и, не отпуская ее, перемещайте указатель мыши. Прямоугольник, появившийся под указателем мыши и изменяющий свои размеры при перемещении мыши, отображает размер будущей области карты ссылок.
- Когда окружность примет подходящий размер, отпустите левую кнопку мыши. Откроется диалог **Вставка гиперссылки** (Insert hyperlink).
- Создайте гиперссылку и щелкните мышью на кнопке **ОК**, чтобы закрыть диалог и завершить создание области гиперссылки.

Чтобы добавить прямоугольную область на карту ссылок:

- Щелкните мышью на кнопке  панели инструментов **Рисунки** (Pictures).
- Переместите указатель мыши в ту часть картинки, где будет располагаться один из углов будущей области.
- Щелкните мышью над точкой одного из углов будущей области.
- Переместите указатель мыши к месту следующего угла будущей области и щелкните на нем мышью. От места первого щелчка к месту второго протянется прямая.
- Последовательно щелкайте на местах всех оставшихся углов будущей карты ссылок. Образующаяся ломаная линия будет границей области.
- В месте последнего угла создаваемого многоугольника дважды щелкните мышью. Откроется диалог **Вставка гиперссылки** (Insert hyperlink).
- Создайте гиперссылку и щелкните мышью на кнопке **ОК**, чтобы закрыть диалог и завершить создание области гиперссылки.

Чтобы изменить размеры области карты ссылок после ее создания:

- Щелкните мышью на области карты ссылок, которую хотите изменить. Область будет окружена рядом квадратных маркеров.
- Переместите указатель мыши на один из этих маркеров и нажмите левую кнопку мыши.
- Не отпуская кнопку мыши, перемещайте указатель мыши в другую позицию. Маркер будет перемещаться вслед за указателем мыши, а вслед за ним будут передвигаться и линии, показывающие границу области.
- Когда вы будете удовлетворены результатом, отпустите левую кнопку мыши. Область карты ссылок изменена.
- При необходимости проделайте описанную процедуру и с остальными маркерами, окружающими область карты ссылок.

## Табличная верстка web-страниц

Все HTML-страницы, которые создавались нами до сих пор, были достаточно просты: расположенный в одну колонку текст с некоторым количеством ссылок и графики. Но большинство настоящих web-страниц имеют более сложное строение (Рис. 3.22).

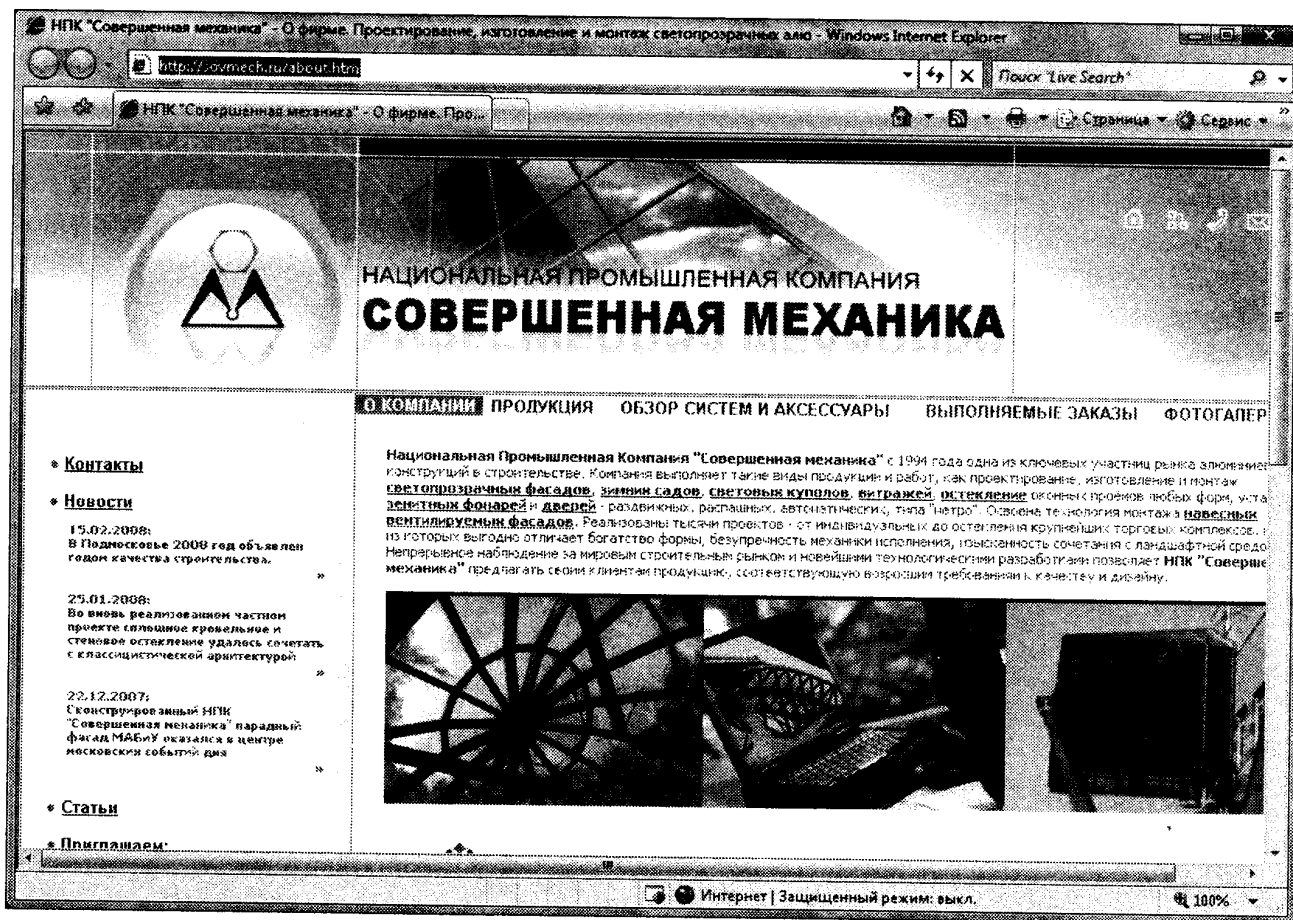


Рис. 3.22. Web-страница средней степени сложности

### Таблица в качестве каркаса web-страницы

Каким же образом упорядочиваются элементы оформления на таких web-страницах? Одним из способов является использование таблиц в качестве каркаса всей web-страницы. На Рис. 3.23 изображено типичное устройство страницы такого сайта. В основе его лежит таблица из трех столбцов и трех строк, причем в верхней и нижней строках таблицы ячейки объединены между собой.

Верхняя строка таблицы отведена под «шапку» web-страницы. Здесь обычно располагается логотип и название сайта.

Первая ячейка второй строки таблицы содержит блок ссылок, с навигацией по разделам сайта. Во второй ячейке второй строки размещается основное содержимое web-страницы, ради которого она и создавалась. В третьей ячейке располагаются новости сайта и различные объявления.

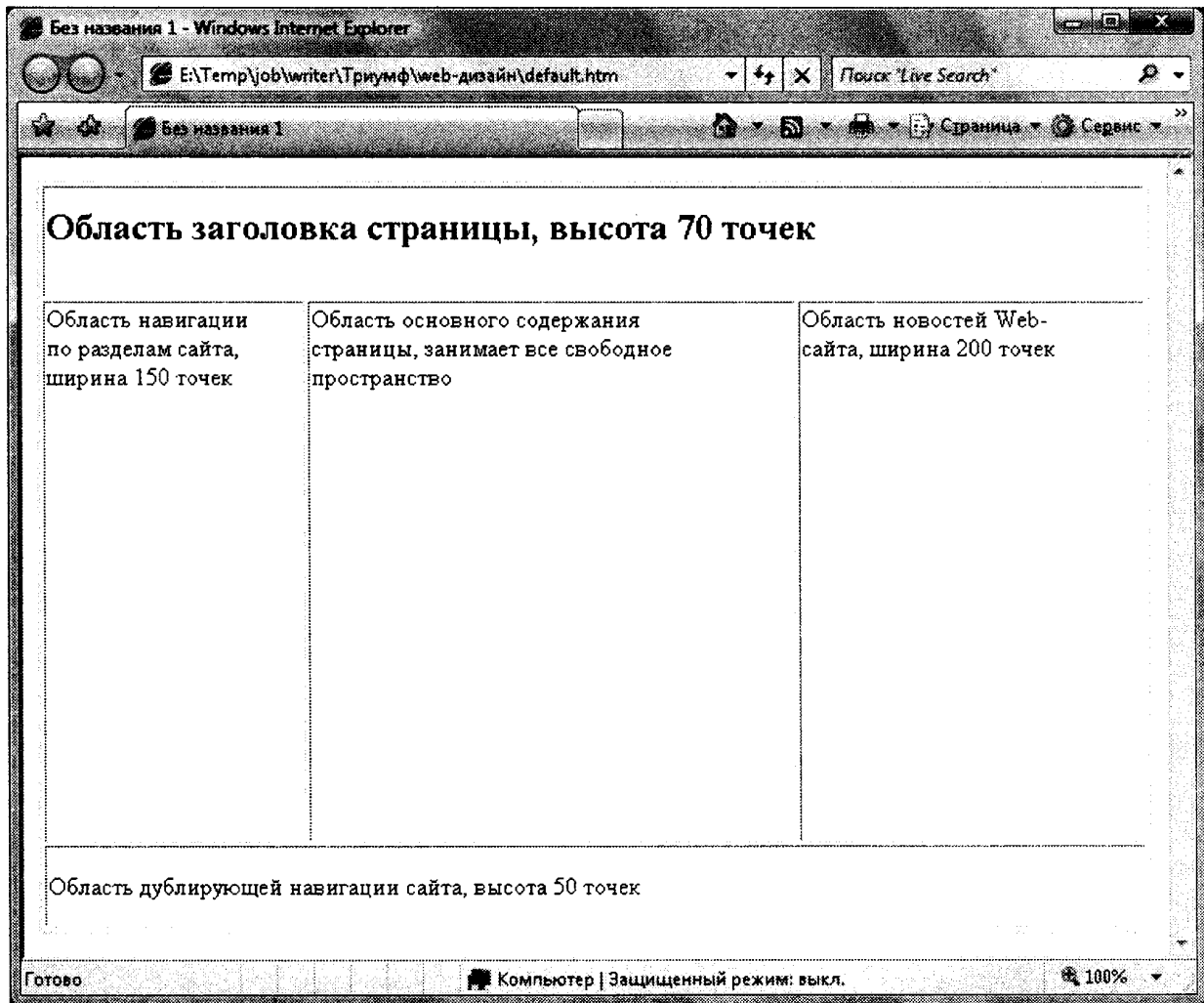
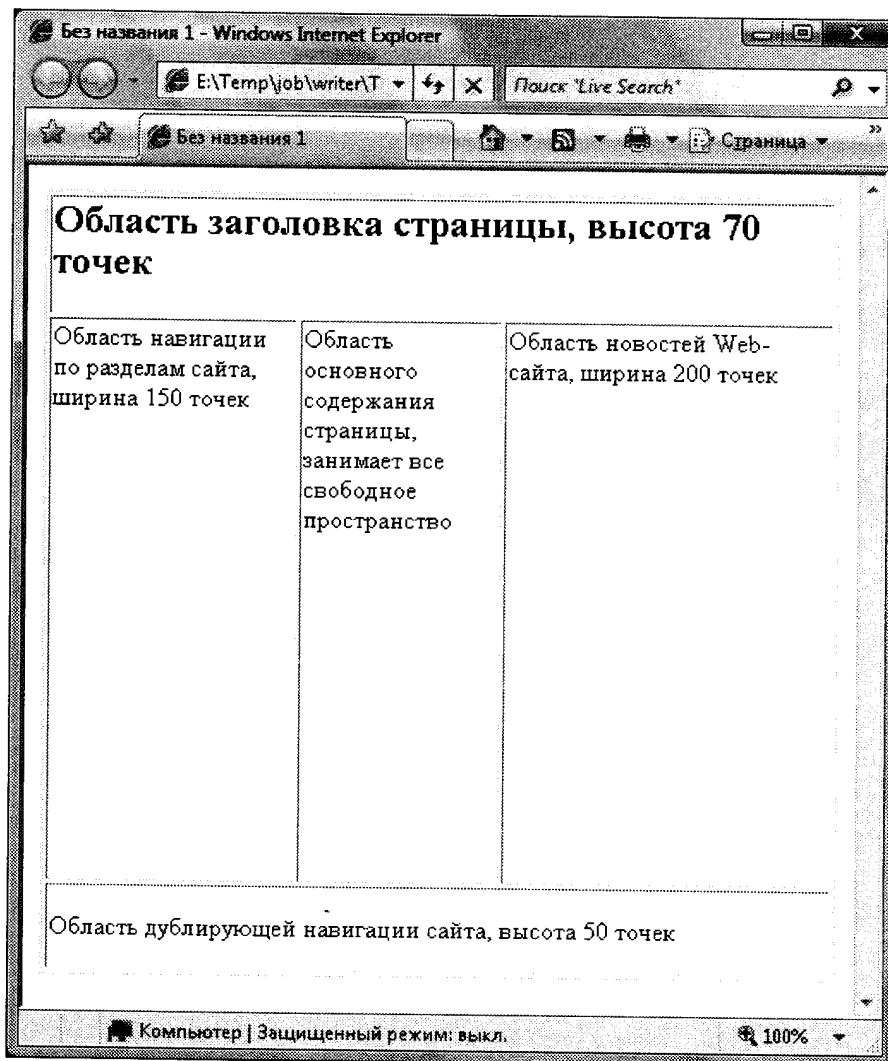


Рис. 3.23. Таблица, используемая в качестве каркаса web-страницы

В третьей строке таблицы-каркаса находится навигационный блок, дублирующий блок, находящийся в первой ячейке второй строки. Это сделано на тот случай, если длина страницы будет существенно превышать высоту окна браузера, и когда посетитель пролистает страницу до конца, блок навигации из второй строки таблицы скроется с экрана.

Размеры всей таблицы задаются равными размерам окна браузера. Все ячейки таблицы, кроме второй ячейки второй строки имеют жестко заданные размеры. Благодаря этому, при различных размерах окна браузера размеры дополнительных элементов страницы остаются постоянными в отличие от основного содержания, которое подстраивается под размеры окна (Рис. 3.24). Благодаря этому можно создавать web-страницы, хорошо выглядящие при различных размерах окна браузера и при этом рационально использующие все доступное пространство. Такой способ создания web-страниц называется «резиновой версткой».

Приведенная схема не является обязательной, существует масса ее вариаций и различий, но главные принципы остаются.



*Рис. 3.24. Поведение каркаса web-страницы при изменении размера окна браузера*

## Работа с таблицами в программе SharePoint Designer 2007

Давайте детально разберемся с особенностями работы с таблицами в программе SharePoint Designer 2007.

Чтобы создать новую таблицу:

- Выберите команду меню **Таблица ♦ Вставить таблицу** (Table ♦ Insert Table). Появится диалог **Вставка таблицы** (Insert table), позволяющий настроить параметры создаваемой таблицы Рис. 3.25.

В полях ввода с счетчиком **Строк** (Rows) и **Столбцов** (Columns) группы элементов настройки **Размер** (Size) задается количество строк и столбцов, которые будет содержать таблица.

В группе элементов управления **Положение** (Position) находятся элементы управления расположением таблицы на странице и положением содержимого ячеек внутри таблицы.

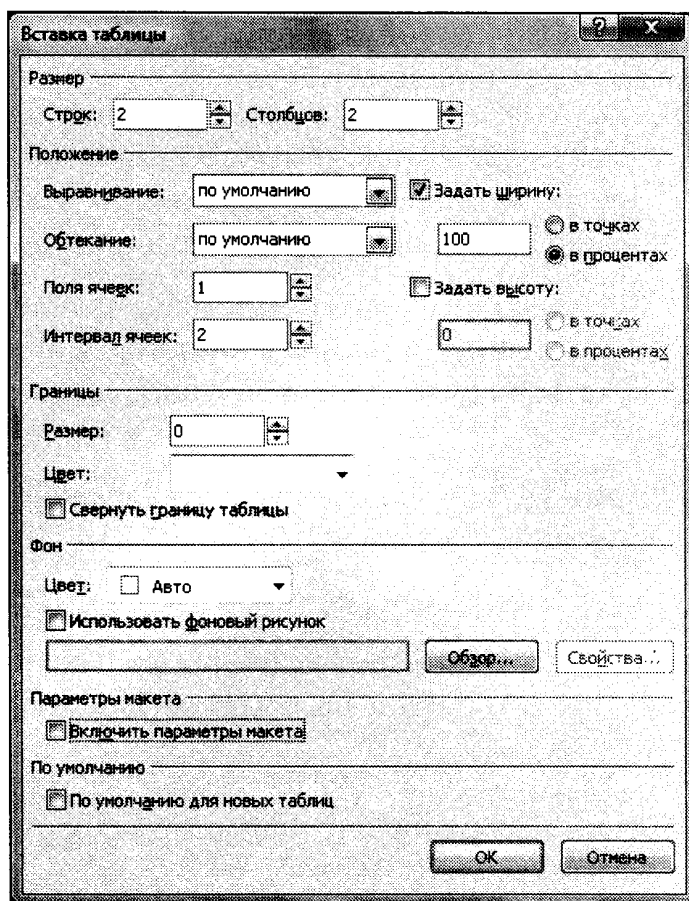


Рис. 3.25. Диалог **Вставка таблицы** (Insert table)

В открывающемся списке **Выравнивание** (Align) можно выбрать способ выравнивания содержимого ячеек. Впоследствии для каждой конкретной ячейки можно отдельно настроить параметры выравнивания ее содержимого.

Открывающийся список **Обтекание** (Flow) задает способ обтекания таблицы текстом и другими элементами web-страницы.

Поле ввода со счетчиком **Поля ячеек** (Cell fields) определяет расстояние между границами ячеек и их содержимым.

В поле ввода с счетчиком **Интервал ячеек** (Cells interval) можно задать расстояние между соседними ячейками таблицы.

Установив флажок **Задать ширину** (Set width), вы можете явно задать ширину таблицы. Ширина задается в поле ввода под флажком **Задать ширину** (Set width). В зависимости от положения переключателя справа от поля ввода, ширину можно задавать **в точках** (points) и **в процентах** (percents). Под шириной 100% подразумевается ширина окна web-браузера.

Установив флажок **Задать высоту** (Set height), вы можете явно задать высоту таблицы. Высота задается в поле ввода под флажком **Задать высоту** (Set height). В зависимости от положения переключателя справа от поля ввода, высоту можно задавать **в точках** (points) и **в процентах** (percents). Под высотой 100% подразумевается высота окна web-браузера.

В группе элементов управления **Границы** (Borders) располагаются настройки, связанные с отображением границ таблицы.

- ✓ Поле ввода со счетчиком **Размер** (Size) задает толщину границы таблицы в точках.
- ✓ Открывающийся список **Цвет** (Color) позволяет выбрать цвет, которым будет отображаться граница.
- ✓ Если установить флажок **Свернуть границу таблицы** (Cut-down table border), то граница будет отображаться в виде одиночной линии, иначе она будет принимать псевдообъемный вид.

В группе **Фон** (Background) собраны элементы, управляющие фоновым изображением таблицы.

- ✓ Открывающийся список **Цвет** (Color) позволяет выбрать цвет фона таблицы.
- ✓ Если установлен флажок **Использовать фоновый рисунок** (Use background picture), то вы сможете задать таблице в качестве фона изображение.

Чтобы установить фоновое изображение для таблицы:

- Щелкните мышью в поле ввода и введите адрес, по которому располагается картинка, которую вы хотите сделать фоновой,

либо

- Щелкните мышью на кнопке **Обзор** (Browse), откроется диалог **Выбрать фоновый рисунок** (Choose background picture).
- Перейдите к каталогу, где находится изображение, которое вы хотите сделать фоновым, и щелкните мышью на его названии.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы принять сделанный выбор и закрыть диалог **Выбрать фоновый рисунок** (Choose background picture).

Если установить флажок **По умолчанию для новых таблиц** (By default for new tables) группы элементов управления **По умолчанию** (By default), то все параметры, которые вы установили для этой таблицы, станут настройками по умолчанию для всех следующих таблиц, которые вы будете создавать.

- Чтобы завершить процесс создания таблицы и закрыть диалог **Вставка таблицы** (Insert table), щелкните мышью на кнопке **ОК**. Таблица будет создана.

Впоследствии, если у вас возникнет необходимость изменить параметры уже созданной таблицы:

- Щелкните правой кнопкой мыши где-нибудь внутри таблицы. Появится контекстное меню.
- Щелкните мышью на пункте контекстного меню **Свойства таблицы** (Table properties). Откроется диалог **Свойства таблицы** (Table properties) с параметрами настройки таблицы.

Настройки, содержащиеся в диалоге **Свойства таблицы** (Table properties), аналогичны настройкам в диалоге за одним исключением. В полях ввода со счетчиком **Строк** (Rows) и **Столбцов** (Columns) группы элементов настройки **Размер** (Size), в отличие от диалога **Вставка таблицы** (Insert table), нельзя задавать любое количество столбцов и строк, а можно лишь добавлять к уже существующим новые строки и столбцы.

Чтобы изменить параметры отдельной ячейки таблицы:

- Щелкните правой кнопкой мыши внутри этой ячейки. Появится контекстное меню.
- Щелкните мышью на пункте контекстного меню **Свойства ячейки** (Cell properties). Откроется диалог **Свойства ячейки** (Cell properties), представленный на Рис. 3.26.

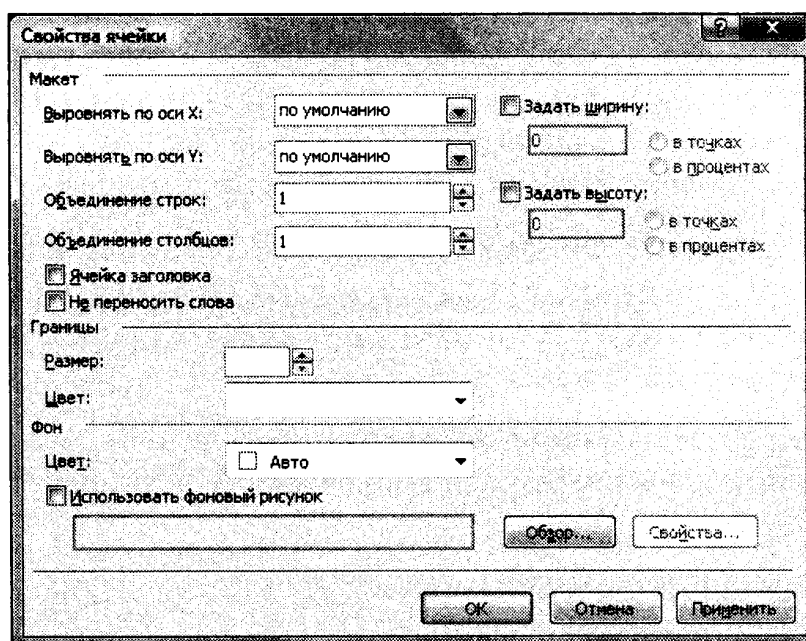


Рис. 3.26. Диалог **Свойства ячейки** (Cell properties)

Чтобы изменить параметры сразу нескольких ячеек таблицы:

- Выделите ячейки таблицы, параметры которых хотите изменить. Для этого переместите указатель мыши в положение над крайней выделяемой ячейкой, нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши так, чтобы область выделения накрыла все ячейки, параметры которых вы хотите изменить.
- Отпустите левую кнопку мыши, выделение создано.
- Щелкните правой кнопкой мыши внутри области выделения. Появится контекстное меню.
- Щелкните мышью на пункте контекстного меню **Свойства ячейки** (Cell properties). Откроется диалог **Свойства ячейки** (Cell properties).



В группе элементов управления **Макет** (Prototype) находятся элементы управления расположением содержимого ячейки.

- ✓ В открываемся списке **Выровнять по оси X** (align on axis X) можно выбрать способ выравнивания содержимого ячейки по горизонтали. Открывающийся список **Выровнять по оси Y** (align on axis Y) позволяет выровнять содержимое ячейки по вертикали.
- ✓ Установив флажок **Задать ширину** (Set width), вы можете явно задать ширину ячейки. Ширина задается в поле ввода под флажком **Задать ширину** (Set width). В зависимости от положения переключателя справа от поля ввода ширину можно задавать **в точках** (points) и **в процентах** (percents). Под шириной 100% подразумевается ширина окна web-браузера за вычетом ширины остальных ячеек таблицы.
- ✓ Установив флажок **Задать высоту** (Set height), вы можете явно задать высоту ячейки. Высота задается в поле ввода под флажком **Задать высоту** (Set height). В зависимости от положения переключателя справа от поля ввода высоту можно задавать **в точках** (points) и **в процентах** (percents). Под высотой 100% подразумевается высота окна web-браузера за вычетом высоты остальных ячеек таблицы.
- ✓ Поля ввода со счетчиком **Объединение строк** (Rows integration) и **Объединение столбцов** (Columns integration) позволяют объединять строки и столбцы соответственно. Вводимые в эти поля числа определяют, какое количество строк или столбцов будет объединено.
- ✓ Установленный флажок **Ячейка заголовка** (Header Cell) означает, что данная ячейка является заголовком. Шрифт в такой ячейке будет полужирным и выровнен посередине ячейки.
- ✓ Установка флажка **Не переносить слова** (No wrap words) запрещает в данной ячейке перенос слов со строки на строку.
- ✓ В группе элементов управления **Границы** (Borders) располагаются настройки, связанные с отображением границ таблицы. Открывающийся список **Цвет** (Color) позволяет выбрать цвет, которым будет отображаться граница ячейки.
- ✓ В группе **Фон** собраны элементы, управляющие фоновым изображением ячейки. Открывающийся список **Цвет** (Color) позволяет выбрать цвет фона ячейки. Установив флажок **Использовать фоновый рисунок** (Use background picture), вы сможете задать таблице в качестве фона изображение.

Чтобы установить фоновое изображение для ячейки:

- Щелкните мышью в поле ввода и введите адрес, по которому располагается картинка, которую вы хотите сделать фоновой,  
либо
- Щелкните мышью на кнопке **Обзор** (Browse), откроется диалог **Выбрать фоновый рисунок** (Choose background picture).

- Перейдите к каталогу, где находится изображение, которое вы хотите сделать фоновым, и щелкните мышью на его названии.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы принять сделанный выбор и закрыть диалог **Выбрать фоновый рисунок** (Choose background picture).
- Установив все параметры выделенных ячеек таблицы, щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог **Свойства ячейки** (Cell properties).

Помимо использования полей ввода со счетчиком **Объединение строк** (Rows integration) и **Объединение столбцов** (Columns integration), объединить ячейки можно следующим образом:

- Выделите ячейки таблицы, которые хотите объединить. Для этого переместите указатель мыши в положение над крайней выделяемой ячейкой, нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши так, чтобы область выделения накрыла все ячейки, которые необходимо выделить.
- Отпустите левую кнопку мыши, выделение создано.
- Щелкните правой кнопкой мыши в пределах выделения. Появится контекстное меню.
- Щелкните мышью на пункте **Изменить ♦ Объединить ячейки** (Change ♦ Combine cells) контекстного меню. Ячейки будут объединены.

Чтобы разбить одну ячейку на несколько:

- Щелкните правой кнопкой мыши внутри разбиваемой ячейки. Появится контекстное меню.
- Щелкните мышью на пункте **Изменить ♦ Разбить ячейки** (Change ♦ Divide cells) контекстного меню. Появится диалог **Разбиение ячеек** (Divide cells), Рис. 3.27.

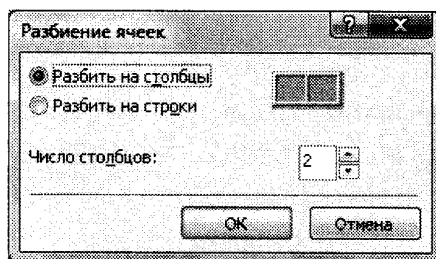


Рис. 3.27. Диалог **Разбиение ячеек** (Divide cells)

Если вы хотите разбить ячейку по горизонтали на несколько столбцов:

- Установите переключатель в положение **Разбить на столбцы** (Divide to columns).
- Впишите в поле с счетчиком **Число столбцов** (Columns number) число ячеек, на которое хотите разбить исходную ячейку.

- ▶ Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Если вы хотите разбить ячейку по вертикали на несколько строк:

- ▶ Установите переключатель в положение **Разбить на строки** (Divide to rows).
- ▶ Впишите в поле с счетчиком **Число строк** (Rows number) число ячеек, на которое хотите разбить исходную ячейку.
- ▶ Щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

Чтобы добавить новые строки или столбцы к таблице:

- ▶ Щелкните правой кнопкой мыши на строке таблицы, ниже которой вы хотите добавить новую строку. Появится контекстное меню.
- ▶ Выберите пункт **Вставить** (Insert).
- ▶ В открывшемся меню выберите один из пунктов **Столбец слева** (Column left), **Столбец справа** (Column right), **Строка слева** (Row left) или **Строка справа** (Row right), в зависимости от требуемого расположения строки или столбца,

либо

- ▶ Выберите пункт **Вставить ♦ Строки или столбцы** (Rows or Columns).
- ▶ На экране появится диалог **Вставка строк или столбцов** (Rows or Columns Insert).
- ▶ Чтобы добавить строки, оставьте расположенный вверху переключатель в положении **Строки** (Rows), в поле ввода со счетчиком **Число строк** (Rows number) укажите количество добавляемых строк и в области элементов управления **Расположение** (Location) укажите, в какое место должны быть добавлены строки: **Над выделенным** (Above selected) или **Под выделенным** (Under selected) (Рис. 3.28).
- ▶ Чтобы добавить столбцы, установите расположенный вверху переключатель в положение **Столбцы** (Columns), в поле ввода со счетчиком **Число столбцов** (Columns number) укажите количество добавляемых строк и в области элементов управления **Расположение** (Location) укажите, в какое место должны быть добавлены столбцы: **Слева от выделенного** (Left from selected) или **Справа от выделенного** (Right from selected).

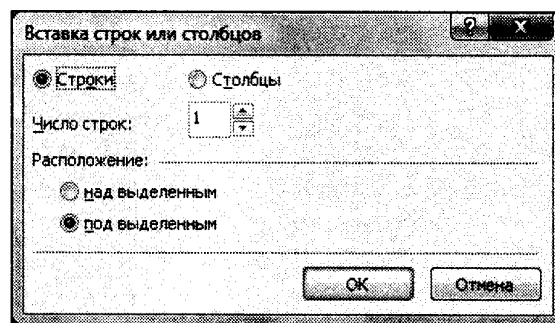


Рис. 3.28. Диалог  
**Вставка строк или столбцов**  
(Rows or Columns Insert)

Вооружившись полученными в этом разделе знаниями, вы теперь можете создавать таблицы любой сложности и конфигурации для любых целей.

## Форматирование текстов

Несмотря на то, что язык HTML предлагает обширные возможности по оформлению web-страниц с использованием графики, звука и видео, основным содержимым большинства web-сайтов остаются тексты. Поэтому необходимо всесторонне ознакомиться с особенностями работы с текстами в HTML.

### Кодировки текста

Сначала давайте разберемся в том, что такое текст.

Любой текст – это некоторая последовательность различных символов из определенного набора знаков. В компьютерах каждый символ текста кодируется одним или двумя байтами. Если символы текста кодируются двумя байтами, то всего возможно 65 535 различных знаков. Этого вполне достаточно, чтобы закодировать символы всех основных алфавитов мира и даже иероглифы. Такая кодировка текста называется Unicode (от Universal-code – универсальный код). В современном компьютерном мире идет постепенный переход к всеобщему использованию кодировки Unicode, но, тем не менее, в подавляющем числе случаев до сих пор используется система, при которой один символ текста кодируется одним байтом.

Такая система кодирования появилась, когда на компьютерах начали обрабатывать тексты и возникла необходимость упорядочивать системы кодирования символов в разных компьютерных системах. Была разработана система кодирования ASCII (American standard code for information interchange – Американский стандартный код обмена информацией). В этой системе кодирования под один символ текста отводилось семь бит, то есть всего 128 вариантов символов. Кодировка ASCII содержала прописные и строчные буквы английского алфавита, цифры, знаки препинания и некоторые дополнительные управляющие символы.

Когда компьютеры стали широко применяться и за пределами Северной Америки, возникла необходимость создавать тексты не только на английском языке. Кодировка ASCII была расширена до 8 бит, т. е. одного байта, и количество вариантов символов текста увеличилось ровно в два раза, до 256 вариантов.

В стандартной 8-битной кодировке ASCII лишние 128 вариантов символов были отданы под различные дополнительные буквы алфавитов, основанных на латинице, например французского, немецкого, испанского, и некоторые дополнительные символы.

Для алфавитов, не основанных на латинице, создавались дополнительные кодировки, первая половина которых совпадала с 7-битной кодировкой ASCII, а вторая содержала символы национального алфавита. Для русского языка, в свое время, было создано как минимум четыре различных кодировки символов, две из которых широко используются и поныне.

Первая из них – стандартная русская кодировка ОС Windows, называемая также 1251 кодовой страницей. Эта кодировка наиболее распространена как среди обыч-

ных текстов, так и на web-сайтах. Вторая кодировка – KOI8-R, эта кодировка часто используется при написании писем электронной почты и является основной русской кодировкой на компьютерах с ОС Linux, Unix и пр.

Чтобы браузер правильно отображал текст web-страницы, он должен знать кодировку, в которой эта страница написана, и правильно определить эту кодировку. Название кодировки, в которой написана web-страница, web-сервер может сам указать браузеру, также можно явно указать кодировку в теле HTML-страницы.

Чтобы явно задать кодировку страницы в программе SharePoint Designer 2007:

- Выберите пункт меню **Файл ♦ Свойства** (File ♦ Properties). Откроется диалог **Свойства страницы** (Page properties), представленный на Рис. 3.14.
- Щелкните мышью на ярлыке вкладки настроек **Язык** (Language). Откроется вкладка настроек языковых параметров страницы (Рис. 3.29).

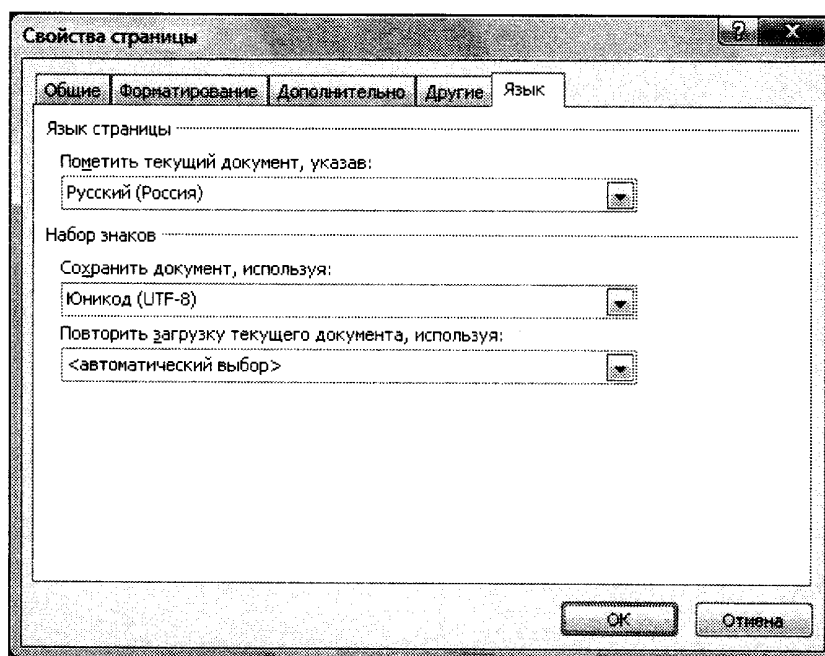


Рис. 3.29. Вкладка **Язык** (Language) диалога **Свойства страницы** (Page properties)

В открываемом списке **Сохранить документ, используя** (Save page using) группы элементов управления **Набор знаков** (Characters set) можно выбрать кодовую страницу, в которой будет создаваться текст страницы.

- ✓ Чтобы установить кодировку страниц Windows, следует выбрать пункт открываемого списка **Сохранить документ, используя** (Save page using) – **Кириллица** (Cyrillic).
- ✓ Чтобы установить кодировку страниц KOI8-R, следует выбрать пункт **Кириллица (KOI8-R)** (Cyrillic (KOI8-R)).
- ✓ Чтобы установить кодировку страниц Unicode, следует выбрать пункт открываемого списка **Сохранить документ, используя** (Save page using) – **Юникод** (Unicode).

- ▶ Установив желаемую кодовую страницу, щелкните мышью на кнопке **ОК**, чтобы принять настройки и закрыть диалог.

## Элементы текстового оформления в документах HTML

Существует множество вариантов различного оформления текстов на web-страницах (Рис. 3.30). Использование потенциала текстового форматирования языка HTML позволяет делать тексты более приятными визуально, оформлять их в согласии с вашими авторскими задумками, дает возможность явно подчеркивать важные слова и мысли.

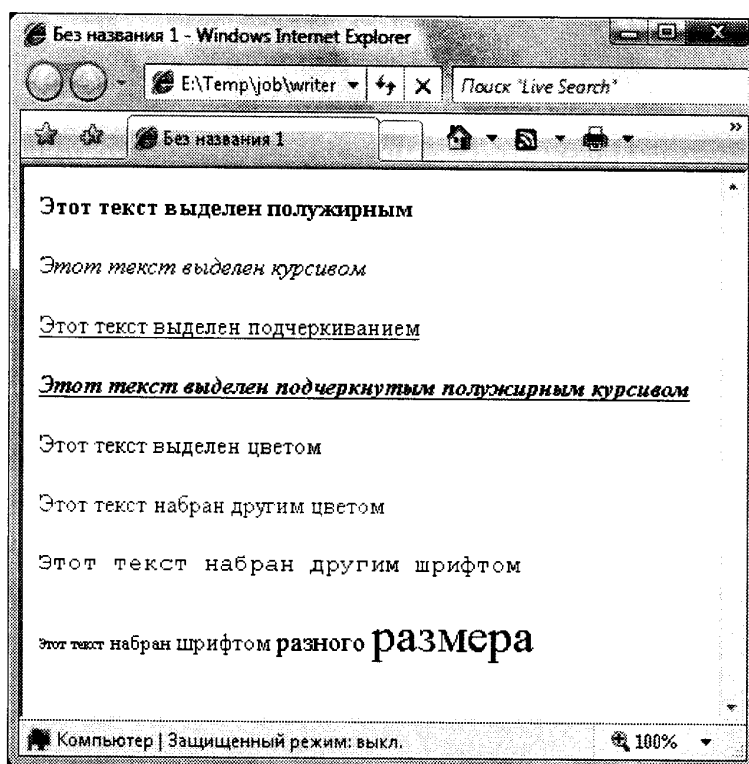




Рис. 3.30. Различные варианты оформления текста на web-страницах

Рассмотрим основные элементы текстового оформления, доступные на HTML-страницах.

- ✓ Полужирное начертание текста. Нажав кнопку **Ж** панели инструментов программы SharePoint Designer 2007, вы можете вводить полужирный текст. Повторное нажатие этой кнопки вернет вас в режим ввода обычного текста.
- ✓ Курсивное начертание текста. Нажав кнопку **К** панели инструментов программы SharePoint Designer 2007, вы можете вводить курсивный текст. Повторное нажатие этой кнопки вернет вас в режим ввода обычного текста.
- ✓ Подчеркнутый текст. Нажав кнопку **Ч** панели инструментов программы SharePoint Designer 2007, вы можете вводить подчеркнутый текст. Повторное нажатие этой кнопки вернет вас в режим ввода обычного текста.

- ✓ Цветовое выделение текста. Нажав кнопку  панели инструментов программы **SharePoint Designer 2007**, вы можете вводить текст, выделенный цветом. Щелкнув мышью на маленьком треугольнике справа от этой кнопки, вы откроете меню выбора цвета выделения. Чтобы выключить режим выделения цветом, выберите пункт **Авто (Auto)** меню выбора цвета выделения.
- ✓ Изменение цвета шрифта. Нажав кнопку  панели инструментов программы **SharePoint Designer 2007**, вы можете вводить текст шрифтом другого цвета. Щелкнув мышью на маленьком треугольнике справа от этой кнопки, вы откроете меню выбора цвета шрифта. Чтобы вернуть обычный цвет шрифта, выберите пункт **Авто (Auto)** меню выбора цвета выделения.
- ✓ Изменение размера шрифта. Выбрав размер шрифта в выпадающем списке **Размер (Size)** панели инструментов программы **SharePoint Designer 2007**, вы можете вводить текст разного размера. Всего в документах **HTML** поддерживается семь разных размеров шрифта, от самого маленького, 1, до самого большого, 7. В скобках рядом с номером указывается эквивалент выбранного размера в типографских пунктах. Чтобы вернуть обычный размер шрифта, следует выбрать пункт **Обычный (Default)** открывающегося списка **Размер (Size)**.
- ✓ Изменение гарнитуры шрифта. Тексты на **HTML**-страницах можно набирать разными гарнитурами шрифтов. Выбрать подходящий шрифт можно в открывающемся списке **Шрифт (Font)** панели инструментов программы **SharePoint Designer 2007**. Выбирая шрифты, помните, что в зависимости от шрифтов, установленных на компьютере посетителя вашей **web**-страницы, тексты могут отображаться не совсем такими же шрифтами, что у вас.

Разные способы форматирования символов текста можно комбинировать между собой, получая достаточно сложное оформление текстов.

## Работа с гиперссылками

И, завершая рассказ об особенностях оформления текстов **web**-страниц, снова коснемся самого интересного элемента **HTML**-документов, а именно – гиперссылок.

Адрес, на который ссылается гиперссылка, по-другому еще называется **URL (Uniform Resource Locator – Унифицированный указатель информационного ресурса)**. Такое название говорит о том, что **URL** может ссылаться не только на **HTML**-документы, но и на любые другие ресурсы Интернета.

В начале текста ссылки указывается название протокола, по которому доступен ресурс. Если **URL** ссылается на файл, расположенный на **web**-сервере, то ссылка будет начинаться с текста **http://**. Ссылки на почтовые адреса начинаются с **mailto://**. Если **URL** ссылается на файл, расположенный на **FTP**-сервере, его текст будет начинаться с **ftp://**, и т. д.

Например, текст **URL** файла **explosive.html**, находящегося в каталоге **main/bombs** **web**-сайта с адресом **www.danger.com**, будет выглядеть так: **http://www.danger.com/main.bombs/explosive.html**.

Ссылка, в которой указываются название протокола и полный путь к файлу, называется абсолютной. Если же ссылка ведет на файл, расположенный на том же компьютере, или сервере, что и web-страница, на которой расположена гиперссылка, то возможен более короткий способ записи URL. Можно просто указать расположение файла, на который ведет ссылка, относительно текущей страницы. Такие ссылки называются относительными.

Например, если гиперссылка, размещенная на странице **index.html**, расположенной в корневом каталоге некоего web-сайта, ведет к файлу **image.jpg**, расположенному в подкаталоге **images** этого же сайта, то адрес файла можно записать так: **images/image.jpg**.

Рассмотрим механизм создания гиперссылок в программе SharePoint Designer 2007.

Чтобы добавить гиперссылку к одному или нескольким элементам web-страницы:

- Выделите этот элемент. Для этого переместите указатель мыши к одному из углов выделяемого элемента или элементов и нажмите левую кнопку мыши.
- Не отпуская кнопку, перемещайте указатель мыши так, чтобы выделенными оказались все необходимые элементы. После этого отпустите кнопку. Выделение создано.
- Щелкните мышью на пункте **Гиперссылка** (Hyperlink) контекстного меню. Появится диалог **Добавление гиперссылки** (Add hyperlink), представленный на Рис. 3.31.

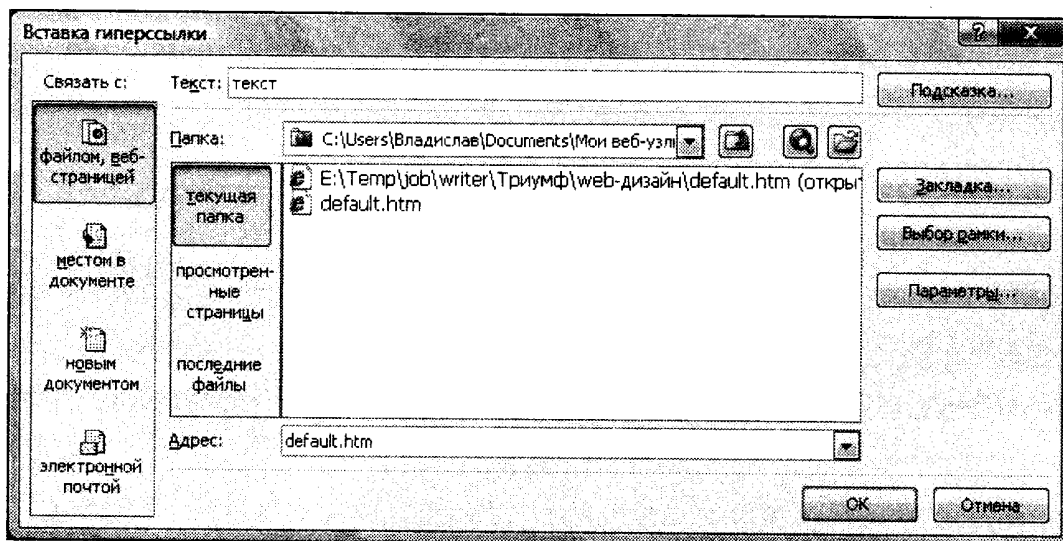


Рис. 3.31. Диалог **Добавление гиперссылки** (Add hyperlink)

С помощью диалога **Добавление гиперссылки** (Add hyperlink) можно создавать несколько видов гиперссылок.

Чтобы создать гиперссылку на уже существующую web-страницу либо файл:

- Щелкните мышью на кнопке **файлом, веб-страницей** (file, web-page) группы элементов управления **Связать с** (Link with).



- ▶ Щелкните мышью в поле ввода **Адрес** (Address) и введите в него URL страницы или файла, на который ссылаетесь.
- ▶ Либо перейдите к папке, в которой находится искомый файл, и щелкните мышью на его названии.
- ▶ Щелкните мышью на кнопке **ОК**, чтобы подтвердить создание ссылки и закрыть диалог.

Чтобы создать гиперссылку на HTML-документ, которого еще не существует, попутно создав этот документ:

- ▶ Щелкните мышью на кнопке **новым документом** (new document) группы элементов управления **Связать с** (Link with).
- ▶ Щелкните мышью в поле ввода **Имя нового документа** (New document name) и введите в него URL страницы или файла, который будет создан,

либо

- ▶ Щелкните мышью на кнопке **Изменить** (Change). На экране появится стандартный диалог открытия файла операционной системы.
- ▶ Перейдите к папке, в которой вы хотите создать новый документ.
- ▶ Щелкните мышью в поле ввода **Имя файла** (Filename) и введите название файла создаваемого документа.
- ▶ Щелкните мышью на кнопке **ОК**, чтобы подтвердить создание документа и закрыть диалог **Создать документ** (Create document).
- ▶ Щелкните мышью на кнопке **ОК**, чтобы подтвердить создание ссылки и закрыть диалог **Добавление гиперссылки** (Add hyperlink).

Чтобы создать ссылку на адрес электронной почты:

- ▶ Щелкните мышью на кнопке **электронной почтой** (e-mail) группы элементов управления **Связать с** (Link with).
- ▶ Щелкните мышью в поле ввода **Адрес эл. почты** (e-mail address) и введите в него почтовый адрес.
- ▶ Щелкните мышью на кнопке **ОК**, чтобы подтвердить создание ссылки и закрыть диалог.

Для упрощения навигации по объемным HTML-страницам можно использовать механизм внутренних ссылок. Суть его заключается в том, что в теле HTML-страницы размещаются специальные «якоря», на которые затем можно делать гиперссылки.

Создание внутритекстовых гиперссылок состоит из двух этапов:

- ✓ Этап первый – создание якоря в теле страницы.
- ▶ Щелкните мышью в месте web-страницы, на которое хотите создать ссылку.

- Выберите пункт меню **Вставка ♦ Закладка** (Insert ♦ Bookmark). Появится диалог **Закладка** (Bookmark), представленный на Рис. 3.32.
- Щелкните мышью в поле ввода **Имя закладки** (Bookmark name) и введите название якоря.
- Щелкните мышью на кнопке **ОК**, чтобы создать якорь и закрыть диалог.
- ✓ Этап второй – создание ссылки на якорь:
- Выделите элемент HTML-страницы, к которому хотите добавить ссылку. Для этого переместите указатель мыши к одному из углов выделяемого элемента или элементов и нажмите левую кнопку мыши.
- Не отпуская кнопку, перемещайте указатель мыши так, чтобы выделенными оказались все необходимые элементы. После этого отпустите кнопку. Выделение создано.
- Щелкните мышью на пункте **Гиперссылка** (Hyperlink) контекстного меню. Появится диалог **Добавление гиперссылки** (Add hyperlink).
- Щелкните мышью на иконке **местом в документе** (place in document) группы элементов управления **Связать с** (Link with).
- В списке **Выберите место в документе** (Choose place in document) щелкните мышью на названии якоря, на который хотите сделать ссылку.
- Щелкните мышью на кнопке **ОК**, чтобы подтвердить создание ссылки и закрыть диалог.

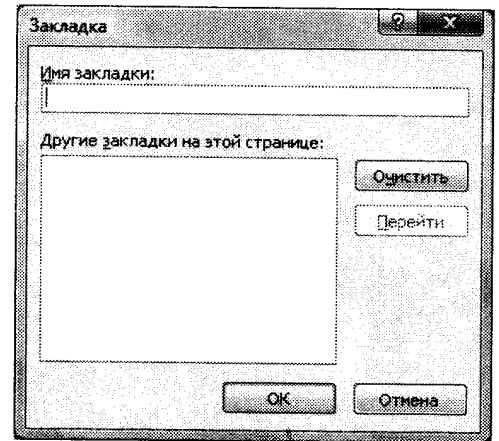


Рис. 3.32. Диалог **Закладка** (Bookmark)

## Заключение

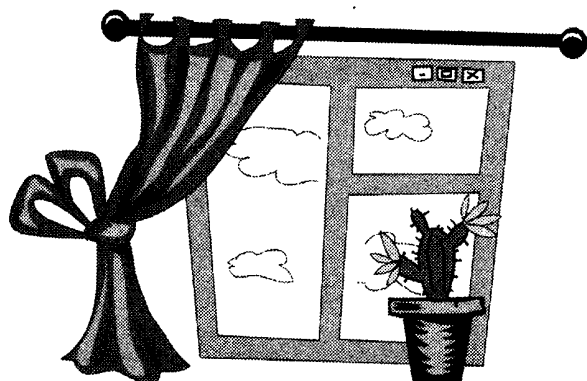
Ваш арсенал разработчика web-страниц пополнился массой методов, позволяющих вам создавать красивые и удобные web-сайты, на которые будет удобно заходить и которыми будет приятно пользоваться.

Теперь вы умеете добавлять на ваши web-страницы графические изображения, видео, звук. Можете разнообразить тексты ваших страниц различными видами форматирования. Познакомились с основами табличной верстки web-страниц и научились создавать карты ссылок. Узнали, что такое URL и как с ними работать.

Полученных вами знаний уже достаточно для создания статических web-страниц высокой степени сложности. В следующей главе речь пойдет о подготовке графических файлов для размещения на web-страницах, что позволит вам еще больше расширить ваши творческие горизонты и даст вам возможность создавать еще более яркие, запоминающиеся, уникальные web-страницы.

## ГЛАВА 4.

# Web-дизайн и графика



Позади три главы книги. Вы уже владеете основными приемами создания статических web-сайтов, знаете, что такое табличная верстка и «резиновый дизайн» сайтов, умеете размещать на web-страницах тексты, графику и прочие элементы оформления.

В этой главе основное внимание мы уделим неотъемлемой части практически любой web-страницы – графике. Вы узнаете новые подробности о графических форматах, используемых в web, узнаете, где лучше брать изображения и как их подготавливать к размещению на ваших страницах. И напоследок вы познакомитесь с некоторыми особенностями оптимизации web-страниц, которые позволят вашим страницам выглядеть одинаково хорошо на компьютерах всех посетителей вашего сайта и быстро загружаться через каналы сети Интернет.

## Форматы графических файлов для web

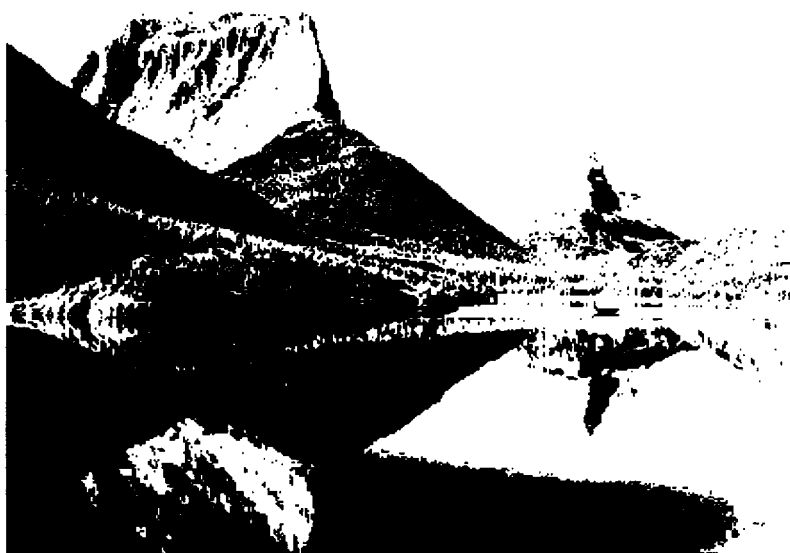
Ранее в этой книге уже заходила речь о компьютерной графике. Теперь пришло время поговорить о ней подробнее. Компьютерная графика бывает двух видов: растровая и векторная.

### Растровая графика

Растровая графика уже упоминалась в предыдущей главе. Растровое изображение состоит из совокупности маленьких точек разного цвета. Отображаемые вместе, они создают на экране монитора иллюзию изображения. Этот тип графики наиболее прост и понятен компьютеру, поскольку все изображения на его экране формируются подобным образом. Если вы присмотритесь к экрану монитора внимательнее, то, безусловно, сможете разглядеть сетку отдельных точек, из которых он состоит. Точки на экране делятся на группы, по три точки разного цвета в каждой. Такая группа из трех точек – красного, зеленого и синего цвета, называется триадой. Одна триада отвечает за одну точку изображения на экране, называемую также пикселем. Количество пикселей, умещающееся на экране по горизонтали и по вертикали, называют разрешением экрана. Например, когда говорится, что разрешение экрана 1024x768, это значит, что в текущем режиме на экране размещается 1024 пикселя по горизонтали и 768 по вертикали.

Зажигая точки триады с разной интенсивностью, можно из трех ее цветов получать любые цвета и оттенки. Такую цветовую систему, построенную на трех основных цветах: красном, зеленом и синем, называют **RGB** (от Red, Green, Blue – красный, зеленый, синий). Интенсивность каждого из основных цветов может принимать 256 различных значений. От «0» – цвета нет, до «255» – максимальная интенсивность. Общее количество цветов, отображаемое в такой системе, достигает 16,7 миллиона.

Количество цветов в изображении называется глубиной цвета и измеряется в битах. Если изображение состоит из двух цветов – черного и белого, то глубина цвета будет 1 бит, каждая его точка описывается 1 битом информации – есть цвет / нет цвета, черное / белое (Рис. 4.1).



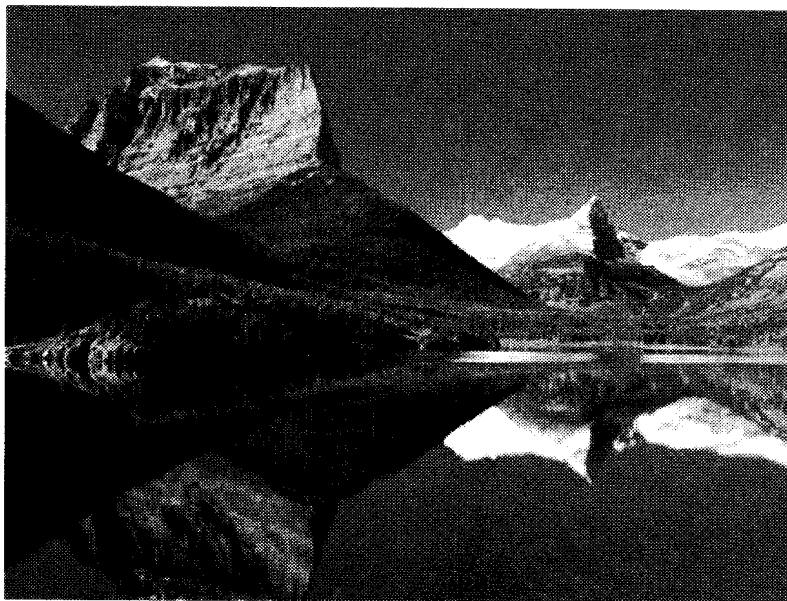
*Рис. 4.1. Черно-белое изображение*

Если изображение состоит из оттенков серого цвета, как черно-белая фотография, то его глубина цвета будет равняться 8 битам, или одному байту. Оттенки серого цвета получаются смешением в равных пропорциях всех основных цветов, поэтому в таком изображении всего 256 различных оттенков (Рис. 4.2). Такое количество оттенков как раз можно записать одним байтом информации.

Полноцветные изображения, содержащие до 16,7 миллиона цветов, являются 24-битными, по 8 битов на основной цвет. Существуют также и 32-битные полноцветные изображения, в них еще один байт информации задает прозрачность точки изображения, от «0» до «255».

Кроме полноцветных, существуют еще так называемые «палитровые» цветные изображения. Файлы таких изображений содержат таблицу, называемую палитрой, в которой определены несколько **RGB**-цветов, и само изображение может содержать только цвета из этой палитры. Цвет каждой точки изображения задается номером цвета из палитры. Глубина цвета таких изображений зависит от количества цветов в палитре. 1-битные изображения содержат не более 2-х цветов, 2-х битные – не более

4, 3-х битные – не более 8, 4-х битные – не более 16 цветов, 5-битные – не более 32-х, 6-битные – не более 64-х цветов, 7-битные – 128 и, наконец, 8-битные – 256 цветов.



*Рис. 4.2. Изображение из оттенков серого цвета*

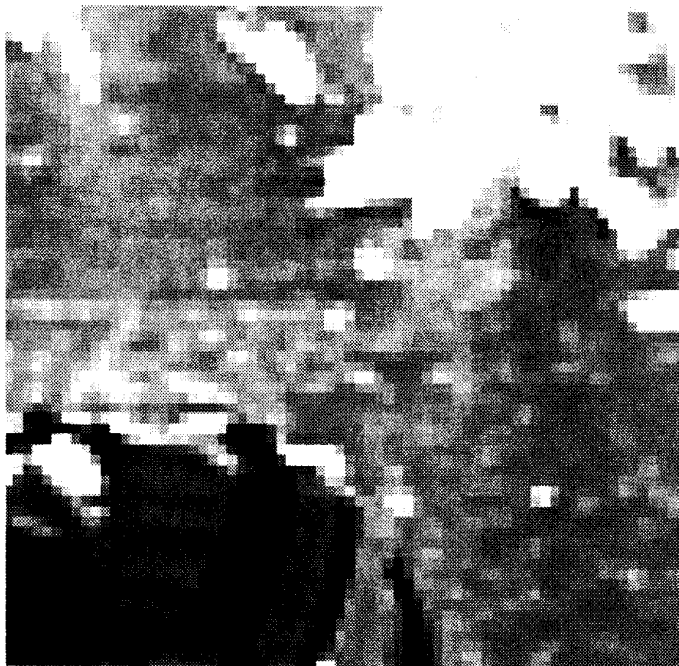
В простейшем случае, для того чтобы отобразить на экране растровое изображение, компьютеру необходимо всего лишь прочитать графический файл и последовательно отобразить на экране цвета точек изображения. Поэтому растровый формат графики является наиболее популярным в компьютерных технологиях. Вторым преимуществом растровых форматов является то, что с их помощью можно достоверно передать любое изображение, практически без ограничений, любую фотографию, набросок, репродукцию картины и т. д. Достаточно лишь оцифровать исходное изображение, чтобы получить его электронную копию.

Но есть у растровых изображений и существенные недостатки. Первый – сложность масштабирования. Допустим, у вас есть картинка некоторого размера (Рис. 4.3). Если вы захотите увеличить ее, чтобы разглядеть отдельные детали, то вместо этого вы увидите расплывшуюся грудку увеличенных точек (Рис. 4.4).

Второй недостаток – объем, который занимает изображение. Полноцветная картинка, размером 1024x768 точек будет занимать более двух мегабайт. Это несущественно в том случае, если вы держите графику на своем компьютере, но если вы хотите сделать ее доступной через Интернет, то размер становится критически важным. Чтобы уменьшить объем, занимаемый картинкой, можно уменьшить ее размеры, но это также уменьшит и ее детальность. В некоторых случаях можно уменьшать количество цветов в картинке, например от 24-х битного цвета перейти к 8-битному, – это позволит сократить объем картинки примерно в 3 раза. Но этот способ подходит отнюдь не всегда. В графических форматах, применяемых для размещения картинок в WWW, используются различные способы уменьшения объема файлов, основанные на разных принципах, что позволяет значительно их уменьшать, но только до определенной степени.



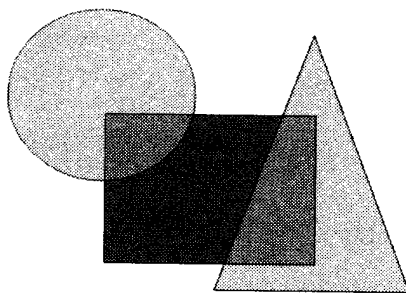
*Рис. 4.3. Исходное изображение*



*Рис. 4.4. Увеличенный фрагмент изображения*

## **Векторная графика**

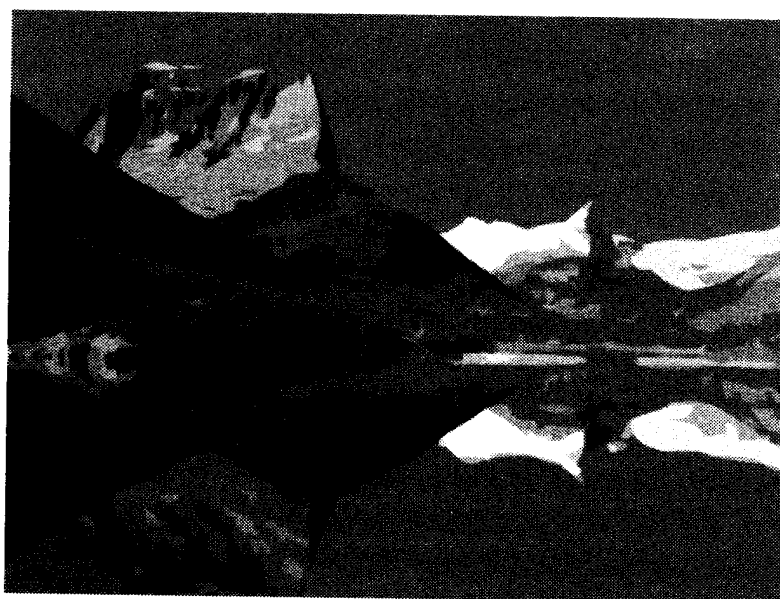
Многих недостатков растровых изображений лишена так называемая векторная графика (Рис. 4.5). Файлы векторной графики, в отличие от растровой, содержат не готовую картинку, а только инструкции, как эту картинку нарисовать. Например, окружность в векторном формате будет записана диаметром, координатами центра, цветами самого круга и его границы. Также и другие геометрические фигуры. Более сложные изображения состояются из простейших геометрических фигур, собранных из линий в различных комбинациях.



*Рис. 4.5. Простейшее векторное изображение*

Преимущества такого подхода очевидны: любое векторное изображение можно растягивать и сжимать во много раз без потери качества самого изображения, и объем картинке не зависит от ее размеров. Еще одним плюсом векторной графики является легкость манипуляции уже созданными изображениями. Вы всегда можете изменить прочерченную линию, подправить нарисованную геометрическую фигуру, поменять цвета и т. д. Все это делает векторную графику очень удобной для рисования на компьютере. Одним из примеров векторной графики, с которым вы сталкиваетесь каждый раз, работая за компьютером, являются текстовые шрифты. Каждый символ большинства из компьютерных шрифтов представляет собой отдельный векторный рисунок, что позволяет изменять его размеры в широких пределах, менять его цвет и т. д.

Но у векторной графики есть свои специфические недостатки. Для отображения в векторном формате подходят далеко не все изображения. Лучше всего для отображения в векторном виде подходят изображения, которые можно разбить на геометрические фигуры и линии: рисунки мультипликационного типа, схемы и планы и т. д. Категорически не подходят для преобразования в векторный вид изображения фотографического типа, с множеством непредсказуемых и мелких деталей (Рис. 4.6). Кроме того, отображение векторных картинок создает существенную нагрузку на компьютер, которому нужно преобразовать инструкции из файла в картинку на экране.



*Рис. 4.6. Фотографическое изображение, преобразованное в векторный формат*

## Форматы графики, применяемые в web

В связи с большей универсальностью и меньшей требовательностью к ресурсам компьютера, технологии растровых изображений получили повсеместное распространение, в том числе и в WWW. И хотя сейчас вопрос производительности компьютеров не является столь острым и векторные технологии постепенно приходят в web, подавляющая часть графики на web-страницах выполняется в растровых форматах.

Для размещения картинок на web-страницах в основном используют два растровых графических формата: **GIF** и **JPEG**. Хотя оба эти формата созданы для того, чтобы служить средством хранения и переноса графики, каждый из них обладает рядом особенностей, знание которых даст вам возможность сделать правильный выбор формата для вашей графики и позволит с минимальными потерями реализовать ваш творческий замысел.

### Формат GIF

Формат **GIF** (Graphic Interchange Format – формат обмена графикой) изначально создавался корпорацией CompuServe для передачи графики в своей компьютерной сети, но стал пользоваться большой популярностью и за ее пределами. Основной особенностью формата **GIF** является то, что он поддерживает только палитровые изображения не более чем с 256 цветами, поэтому его нецелесообразно использовать для размещения фотографических и других полноцветных изображений.

Чтобы изображения было проще передавать по каналам связи, графическая информация в **GIF**-файлах хранится в сжатом виде. Механизмы сжатия, заложенные в этот формат, лучше всего работают с изображениями, имеющими достаточно большие однотонные области с ровными границами. Поэтому **GIF** используют обычно для размещения рисованных элементов оформления сайтов, рисунков в мультипликационном стиле, чертежей, схем, а также других изображений с ограниченным количеством цветов.

В один файл формата **GIF** можно записать несколько изображений, которые будут показываться последовательно, через заданные промежутки времени. Эта возможность формата позволяет создавать с его помощью анимации и небольшие мультфильмы без звукового сопровождения.

Полезной особенностью формата **GIF** является возможность сделать один из цветов картинки прозрачным, при этом через участки прозрачного цвета будет видно то, что находится под изображением. То есть, в случае web-страницы – это будет фоновое изображение или цвет. Эта возможность широко используется в тех случаях, когда фон является одним из важных элементов оформления страницы.

Для ускорения отображения **GIF**-картинок на web-страницах в формате **GIF** предусмотрено чересстрочное отображение графики. Сначала загружаются четные строки изображения, а затем нечетные, или наоборот.



## Формат JPEG

В отличие от формата **GIF**, который создавался одной фирмой, стандарт **JPEG** (Joint Photographic Experts Group – Объединенная группа экспертов по фотографии) был разработан под эгидой ISO (International Organization for Standardization – Международная организация по стандартизации) – ведущей международной организации по стандартам и ССИТ (Consultative Committee for International Telephone and Telegraphy – Консультативный комитет по международной телефонии и телеграфу), организации, которая устанавливает стандарты для телефонии, радио, телевидения и т. п. Стандарт **JPEG** разрабатывался специально для хранения и передачи фотографических изображений и лучше всего он подходит именно для этих целей.

Изображения в формате **JPEG** могут содержать до 16,7 миллиона цветов, этого более чем достаточно для передачи деталей любого фотографического изображения. При преобразовании в формат **JPEG** изображения подвергаются сжатию с потерей данных. Изображение анализируется программой преобразования с точки зрения особенностей человеческого восприятия и отбрасывает детали, человеку незаметные. Это позволяет уменьшать размер фотографических изображений в десятки раз без видимых искажений картинки. Размер изображения и его качество определяются степенью сжатия, выбранной в настройках программы, создающей **JPEG**-файлы.

Особенности алгоритма сжатия **JPEG** не позволяют добиться столь же впечатляющих результатов при работе с рисунками, имеющими не очень много цветов, с ровными линиями и большими участками одного цвета. На таких изображениях **JPEG** может серьезно «портить» итоговую картинку уже при сравнительно небольших степенях сжатия. В таких случаях разумнее пользоваться форматом **GIF**.

Для ускорения визуальной загрузки картинок на ваши **web**-страницы можно использовать режим прогрессивной загрузки **JPEG**-изображений. В этом режиме в файле **JPEG** кроме основного изображения хранятся еще несколько его копий с уменьшенным разрешением. Сначала загружается копия с самым маленьким разрешением, потом та, что чуть больше, и так далее. И, хотя общий размер файла при этом немного увеличивается и время загрузки в целом возрастает, создается впечатление, что картинка загрузилась почти сразу и только потом постепенно подгрузила свои недостающие детали.

Подведем итог, в каком же формате размещать графику на **web**-страницах? Если изображение содержит немного цветов, состоит из четких линий и геометрических фигур, содержит достаточно большие области одного цвета, то лучшим выбором будет формат **GIF**. Также этот формат незаменим для создания небольших анимированных вставок, рекламных баннеров и в тех случаях, когда необходима частичная прозрачность картинки.

Для размещения многоцветных изображений фотографического типа лучше всего подходит формат **JPEG**, в этой области ему нет равных.

Если вы затрудняетесь, к какому типу отнести изображение, то создайте две его версии – в форматах **GIF** и **JPEG** – и сравните их. Ту, что будет при сравнимом качестве занимать меньше места, и используйте.

## Источники изображений для web-страниц

Перед человеком, решившим использовать графику на своих web-страницах, непременно встанет вопрос: где же брать картинки для оформления страниц?

Существует несколько вариантов ответа. Первый и самый очевидный – создать самому. Необходимые изображения можно нарисовать, отсканировать, сфотографировать и т. д. Существует и другой способ получения картинок – использование коллекций уже готовых изображений, клипартов.

### Коллекции изображений, клипарты

Клипарты существуют в виде компакт-дисков со сборниками картинок и Интернет-коллекций.

Все клипарты делятся на коммерческие и бесплатные. Целью создания коммерческих коллекций графики является зарабатывание денег, поэтому в открытом доступе, как правило, доступны лишь уменьшенные версии изображений из таких клипартов. Чтобы получить в свое распоряжение полную версию картинки, в большом разрешении, необходимо уплатить определенную сумму. Обычно изображения для таких клипартов создают профессиональные художники и фотографы, поэтому картинки в них обычно очень качественные, но могут стоить достаточно больших денег. Кроме того, зачастую оплачивать их можно только с помощью кредитных карт, выпущенных западными банками. При создании крупных коммерческих сайтов использование таких коллекций может быть вполне оправданным.

Бесплатные коллекции изображений выставляются на публичное обозрение совершенно свободно, но это не значит, что картинки из них можно использовать в абсолютно любых целях. Обычно на сайтах авторов таких коллекций приводятся цели, в которых можно использовать выставляемые изображения. Как правило, против использования графики в некоммерческих целях никто не возражает, но если вы хотите использовать ее для получения выгоды, то необходимо договориться с автором. Это не только поможет вам избежать юридических проблем в будущем, но и простимулирует автора на создание новых изображений.

### Особенности работы с изображениями из клипартов

Используя картинки из готовых коллекций, старайтесь не размещать их на своем web-сайте в неизменном виде, модифицируйте их так, чтобы они максимально отвечали вашим дизайнерским замыслам. Это не только увеличит ценность оформления вашей страницы, но и сделает изображения уникальными, не такими, какими они были в клипарте. Не забывайте, что кроме вас этой коллекцией пользуются сотни и тысячи людей, поэтому вероятность, что кто-то воспользуется такой же картинкой, как и у вас, достаточно велика. Ваша web-страница должна быть уникальной, не похожей на другие. Из этих же соображений старайтесь не использовать изображения, находящиеся в самом начале клипарта, лучше пролистайте его на несколько страниц вперед. По статистике, наибольшей популярностью пользуются изображения, находящиеся на первой – третьей страницах коллекций.

В комплекте со многими графическими и офисными программами часто идут коллекции изображений. Проще всего использовать именно их, но прежде чем делать это, трижды подумайте, каким тиражом расходятся эти программы и сколько еще людей пользуются этими клипартами.

Постарайтесь найти несколько изображений, подходящих для ваших целей, желательно не менее трех. Расположите их в одном каталоге и просмотрите последовательно. Подумайте, какое из них больше отвечает вашим целям. Не поленитесь и проверьте, как будет смотреться каждая из картинок на вашей web-странице, создайте несколько версий web-страницы. Сравните их сейчас и, по прошествии некоторого времени, еще раз.

## Поиск изображений в сети Интернет

Как лучше всего искать рисунки в Интернете? Большинство поисковых систем предусматривает поиск изображений с заданными названиями. Воспользуемся услугами двух наиболее популярных поисковых систем: **Google** и **Yandex**.

### Поиск изображений с помощью поисковой системы Yandex

Чтобы найти изображения с помощью системы **Yandex**:

- ▶ Находясь в Интернете, введите в адресной строке браузера адрес поисковой системы: **http://www.yandex.ru** и нажмите клавишу **Enter**. Откроется web-страница системы **Yandex** (Рис. 4.7).

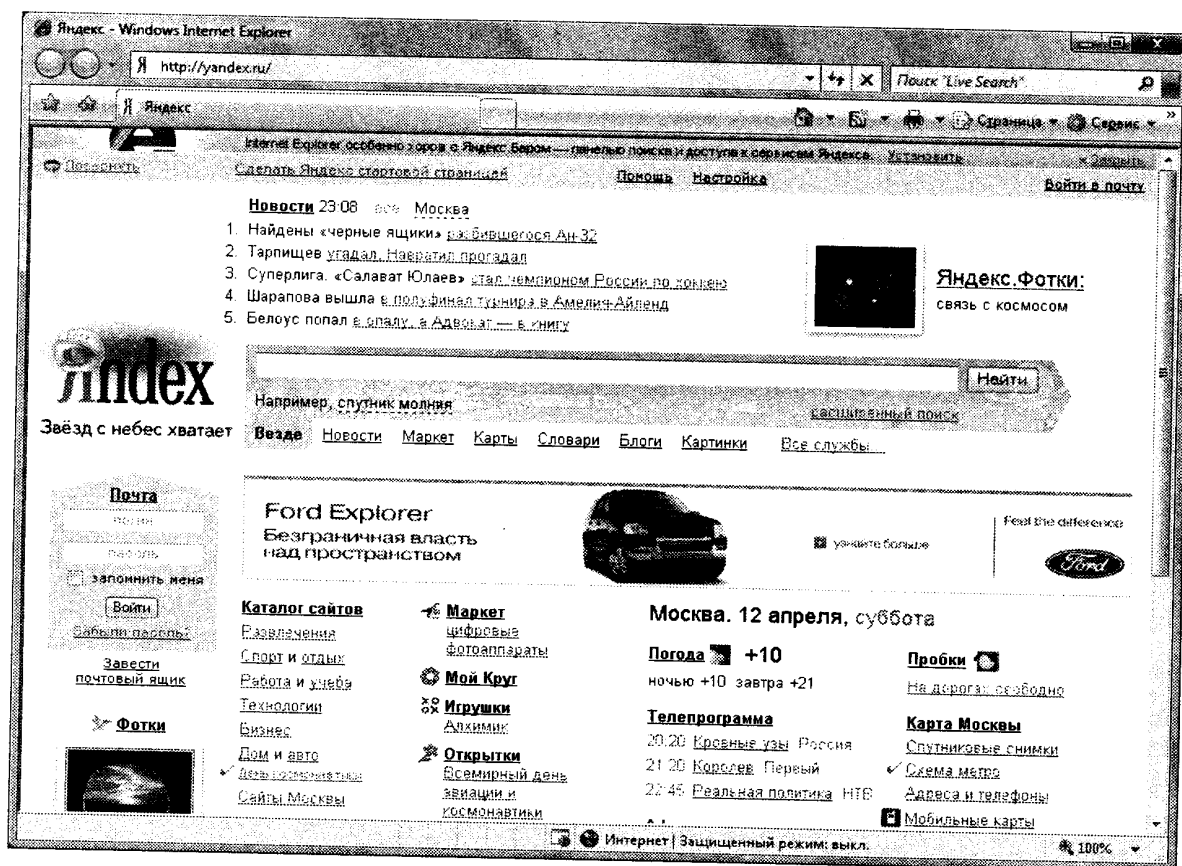


Рис. 4.7. Поисковая система **Yandex**

- Щелкните мышью на ссылке **Картинки** под полем ввода. Откроется страница поиска изображений (Рис. 4.8).

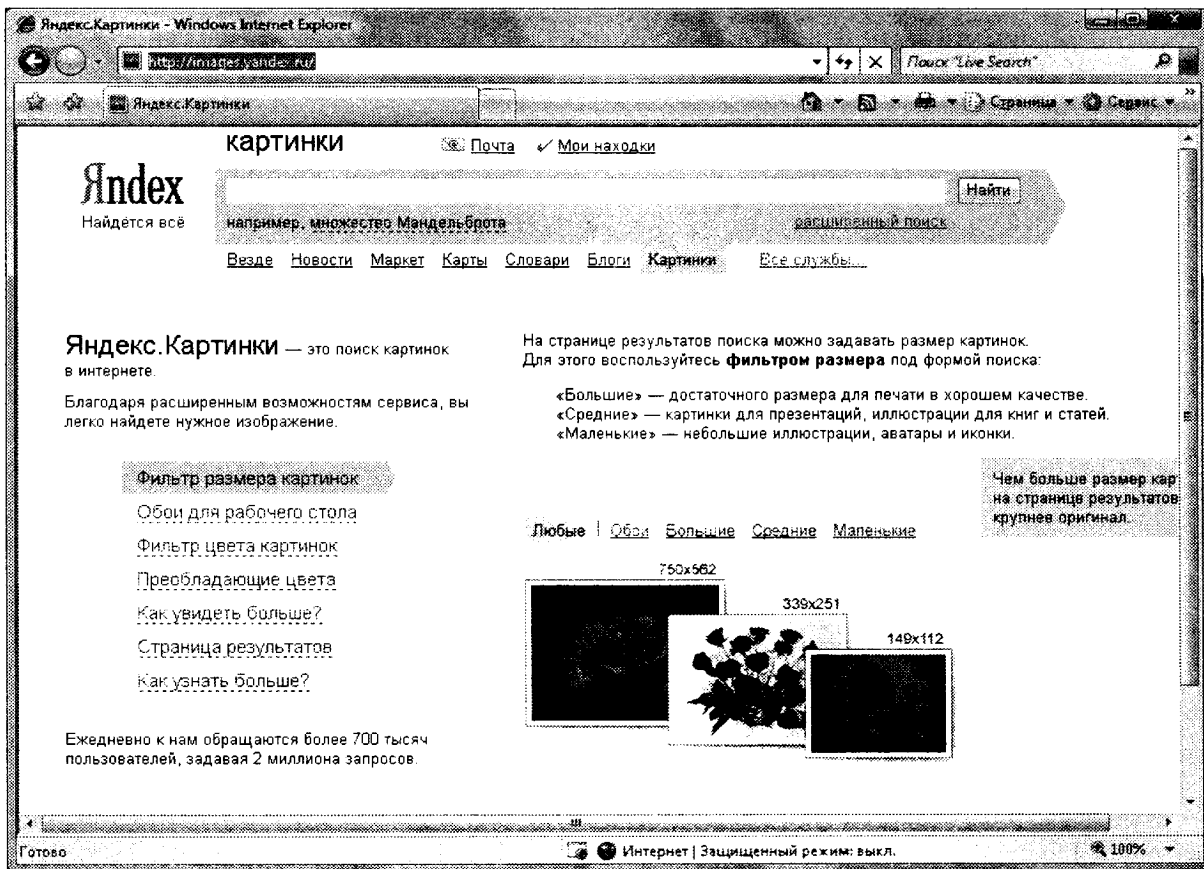


Рис. 4.8. Поиск картинок в системе **Yandex**

- Щелкните мышью в поле ввода и введите слова, которые должны присутствовать в названии картинки. Нажмите клавишу **Enter**. Будет выполнен поисковый запрос.

В результатах поиска вы увидите уменьшенные копии найденных изображений и их названия. Под названиями файлов будут приведены адреса сайтов, на которых находятся найденные изображения (Рис. 4.9).

Обязательно посещайте сайты, с которых вы берете изображения, чтобы урегулировать правовые вопросы. Кроме того, вы можете обнаружить на этих сайтах немало других хороших изображений.

- Щелкните мышью на понравившемся изображении, чтобы открыть его в полный размер в новом окне (Рис. 4.10).
- Щелкните правой кнопкой мыши на полном изображении, появится контекстное меню.
- Щелкните мышью на пункте **Сохранить рисунок как** (Save image as) контекстного меню, появится стандартный диалог сохранения файла операционной системы.
- Перейдите к каталогу, в который вы хотите сохранить изображение, введите имя файла и сохраните изображение на вашем жестком диске.

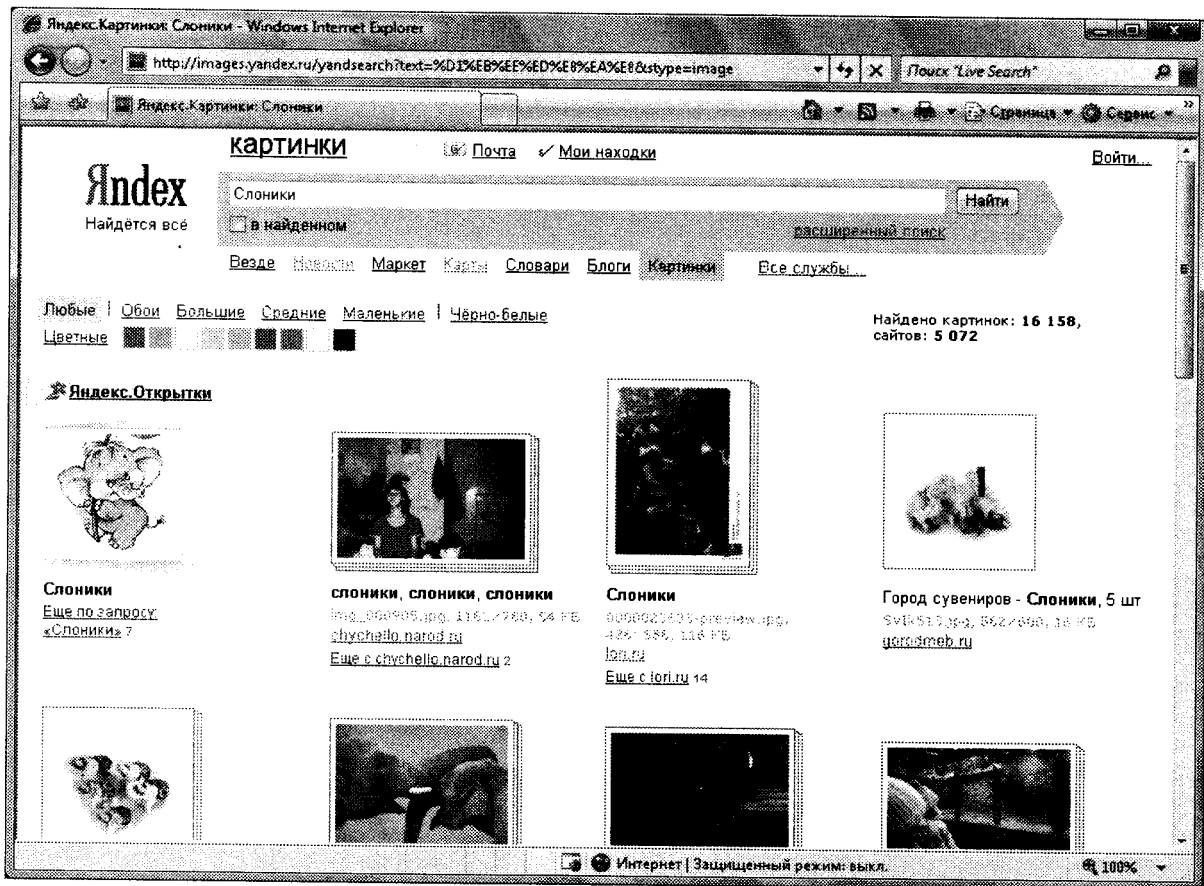


Рис. 4.9. Результаты поиска картинок в системе Яндекс

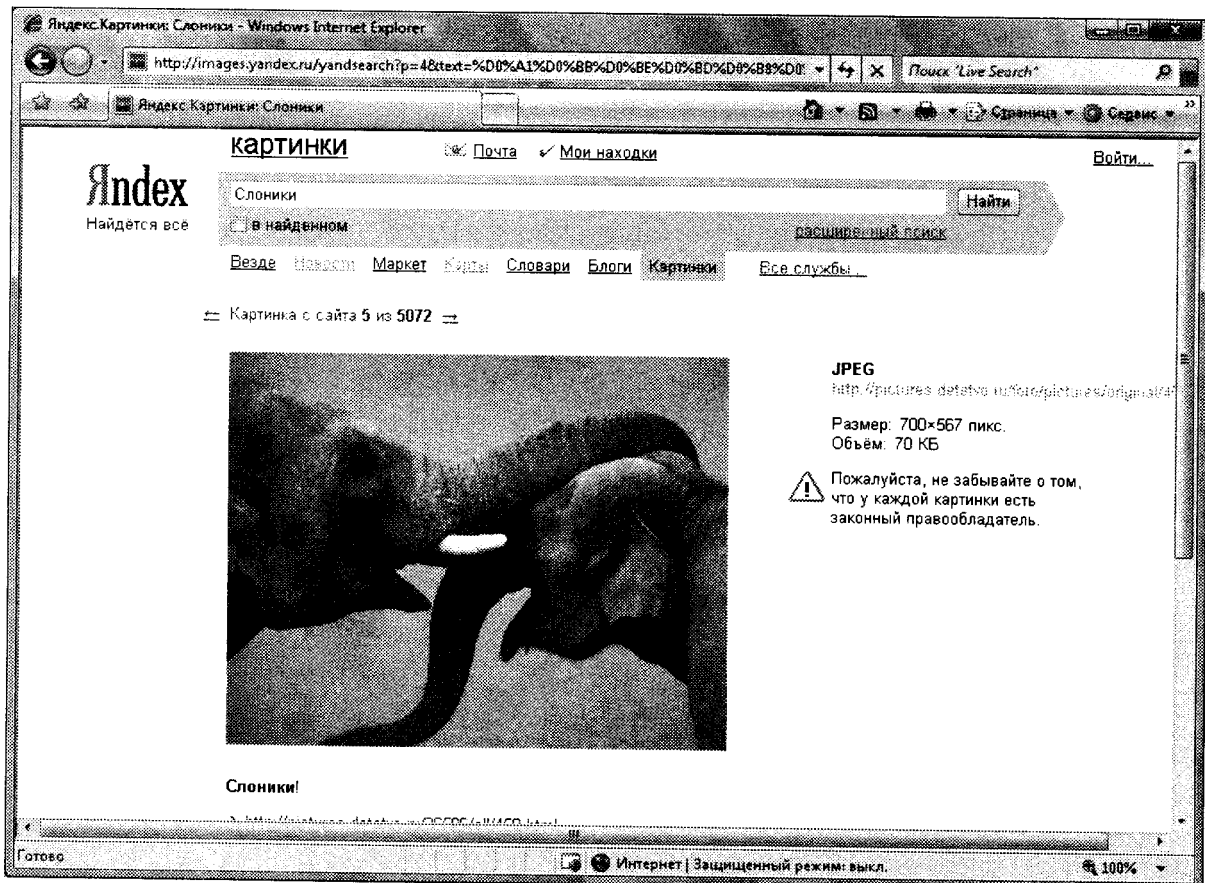


Рис. 4.10. Полная версия изображения

- Щелкните мышью на кнопке браузера **Назад** (Back), чтобы вернуться к странице с уменьшенными копиями изображений.

Пролистывайте дальше результаты поиска и сохраняйте понравившиеся вам изображения, как описано выше.

## Поиск изображений с помощью поисковой системы Google

Чтобы использовать для нахождения изображений в сети Интернет поисковую систему **Google**:

- Находясь в сети Интернет, введите в адресной строке браузера адрес поисковой системы **http://www.google.ru** и нажмите клавишу **Enter**. Откроется web-страница системы **Google**.
- Щелкните мышью на ссылке **Картинки** в левом верхнем углу страницы. Откроется страница поиска изображений.
- Щелкните мышью в поле ввода и введите слова, которые должны присутствовать в названии картинки, либо в ее описании. Нажмите клавишу **Enter**. Будет выполнен поисковый запрос.

В результатах поиска вы увидите уменьшенные копии найденных изображений и их названия. Под названиями файлов будут приведены адреса сайтов, на которых находятся найденные изображения (Рис. 4.11).

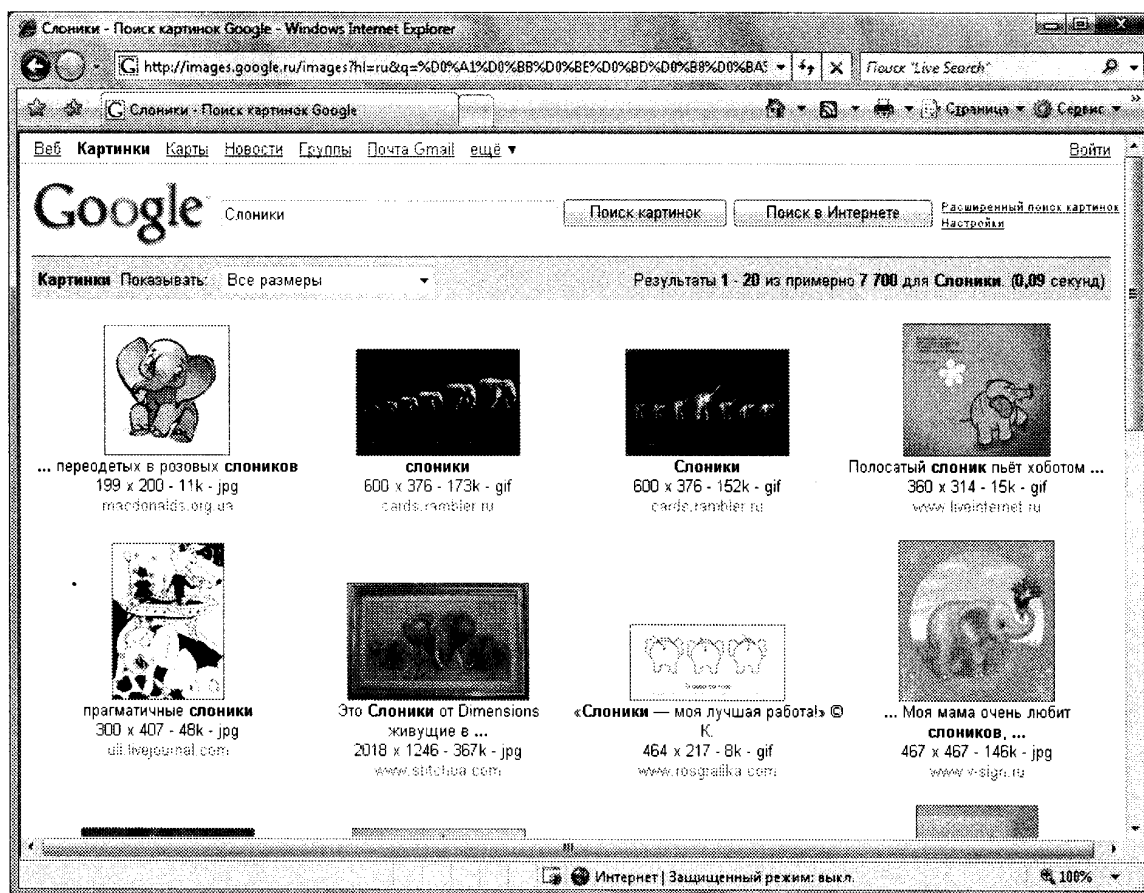


Рис. 4.11. Отображение результатов поиска изображений в системе **Google**

- Щелкните мышью на понравившемся изображении. Браузер откроет страницу сайта, на которой находится полное изображение (Рис. 4.12).

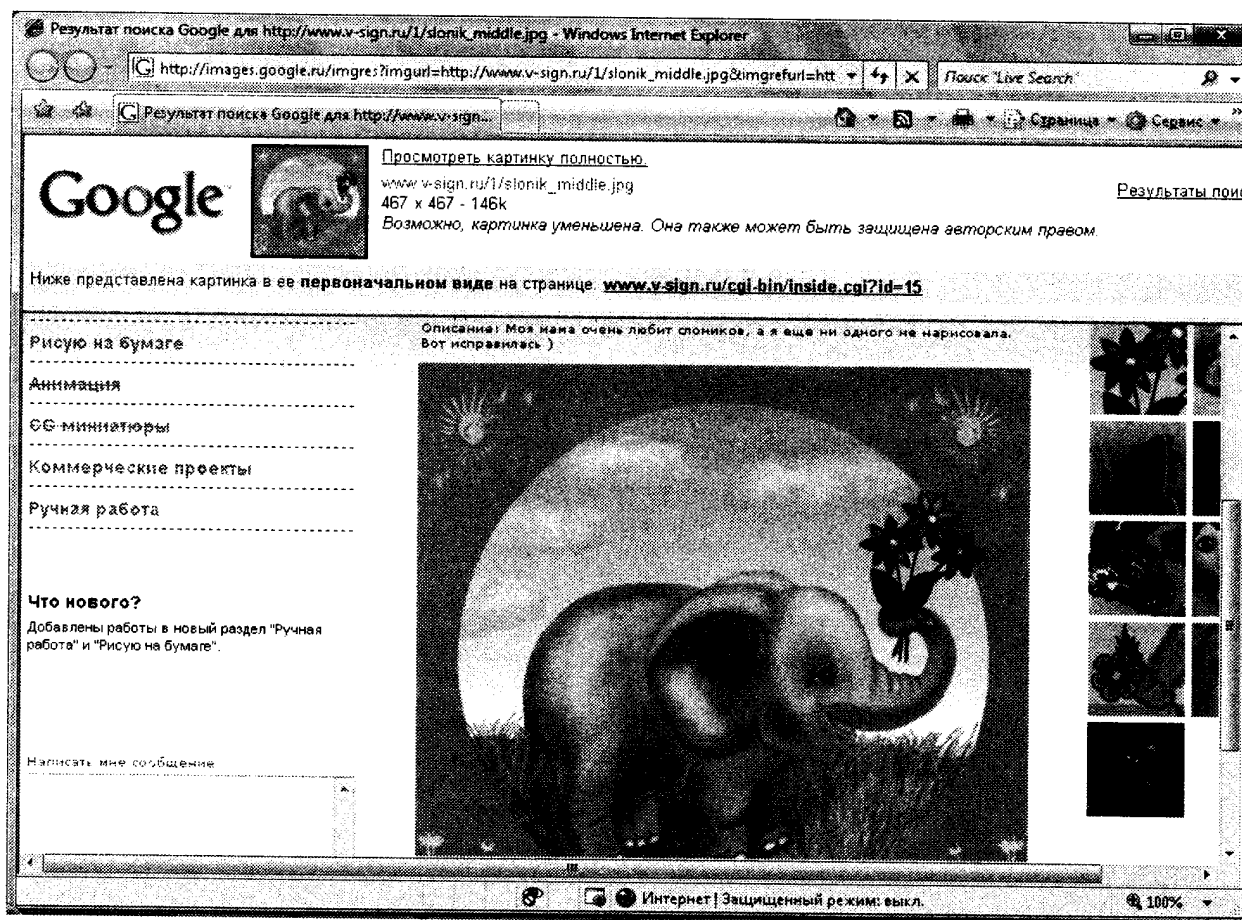


Рис. 4.12. Отображение полной версии изображения в системе **Google**

- Щелкните правой кнопкой мыши на полном изображении, появится контекстное меню.
- Щелкните мышью на пункте **Сохранить рисунок как** (Save image as) контекстного меню, появится диалог **Сохранение рисунка** (Save image).
- Перейдите к каталогу, в который вы хотите сохранить изображение.
- Щелкните мышью в поле ввода **Имя файла** (File name) и введите желаемое название файла.
- Перейдите к каталогу, в который вы хотите сохранить изображение, введите имя файла и сохраните изображение на вашем жестком диске.
- Закройте окно браузера с полной копией изображения, щелкнув на кнопке закрытия окна.

Пролистывайте дальше результаты поиска и сохраняйте понравившиеся вам картинки, как было описано выше.

## Обработка изображений в программе Adobe Photoshop

В процессе создания web-страниц вам придется иметь дело с графикой из совершенно разных источников: картинками из клипартов, отсканированными изображениями, цифровыми фотографиями и т. д. О некоторых приемах подготовки этих изображений к размещению на ваших страницах будет рассказано в этом разделе.

Для работы с изображениями мы будем использовать программу Adobe Photoshop версии CS3, которая является признанным лидером по обработке растровых изображений. Она предоставляет все мыслимые и немыслимые возможности монтажа, цветовой коррекции и ретуши. Позволяет создавать различные спецэффекты и делать многое другое. Поэтому основное внимание мы уделим обработке графики именно в этой программе.

Подробную информацию по установке, настройке и использованию всего спектра функций программы Adobe Photoshop вы можете почерпнуть из книг, посвященных работе с этой замечательной программой. В этой же книге мы лишь коснемся некоторых аспектов применения Adobe Photoshop для подготовки изображений к размещению во Всемирной паутине.

- ▶ Запустите программу Adobe Photoshop, выполнив команду стартового меню **Пуск ♦ Все программы ♦ Adobe Photoshop CS3 (Start ♦ Programs ♦ Adobe Photoshop CS3)**, откроется главное окно программы Adobe Photoshop CS3 (Рис. 4.13).

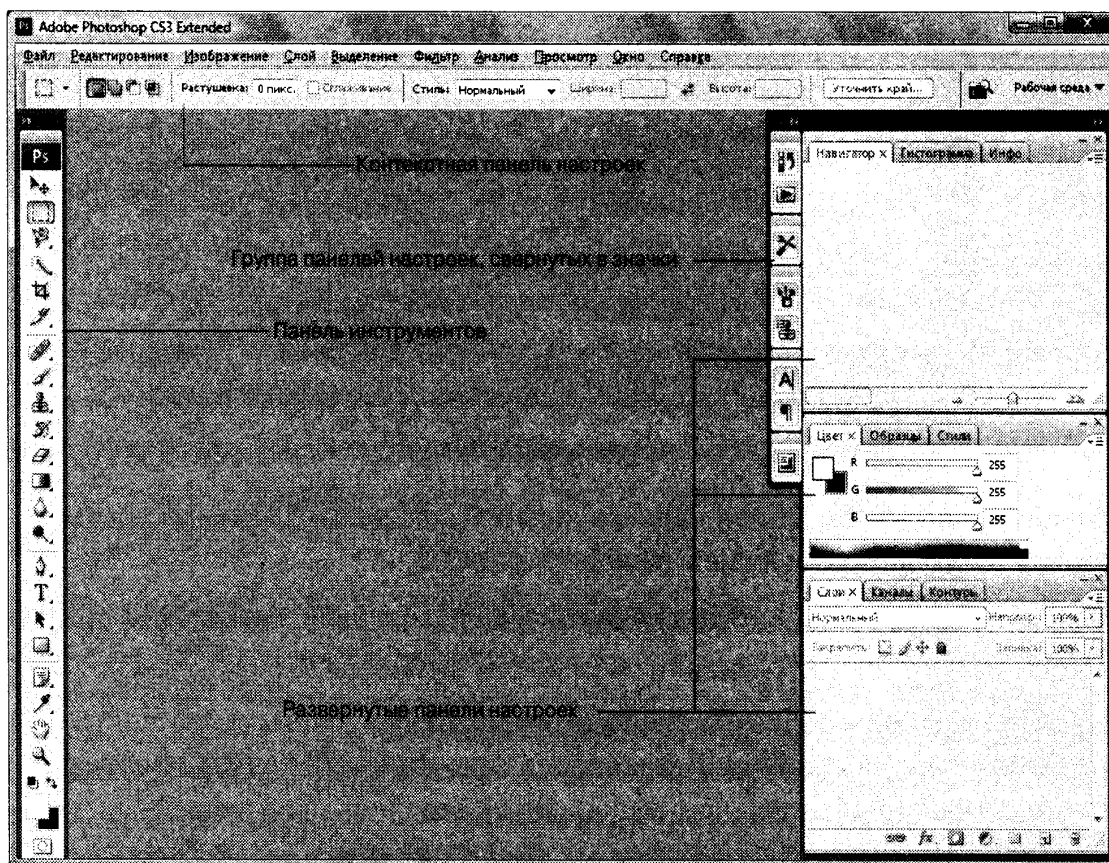


Рис. 4.13. Главное окно программы Adobe Photoshop



Большую часть основного окна программы Adobe Photoshop занимает рабочая область, в которой располагаются окна открытых изображений, плавающая панель инструментов и несколько дополнительных панелей со всевозможными настройками. Также в главном окне под строкой меню находится контекстная панель настроек, в которой доступны все основные настройки для активного рабочего инструмента.

Чтобы открыть какое-либо изображение в программе Adobe Photoshop CS3:

- Выберите команду меню **Файл ♦ Открыть** (File ♦ Open). Появится диалог **Открыть** (Open), Рис. 4.14.

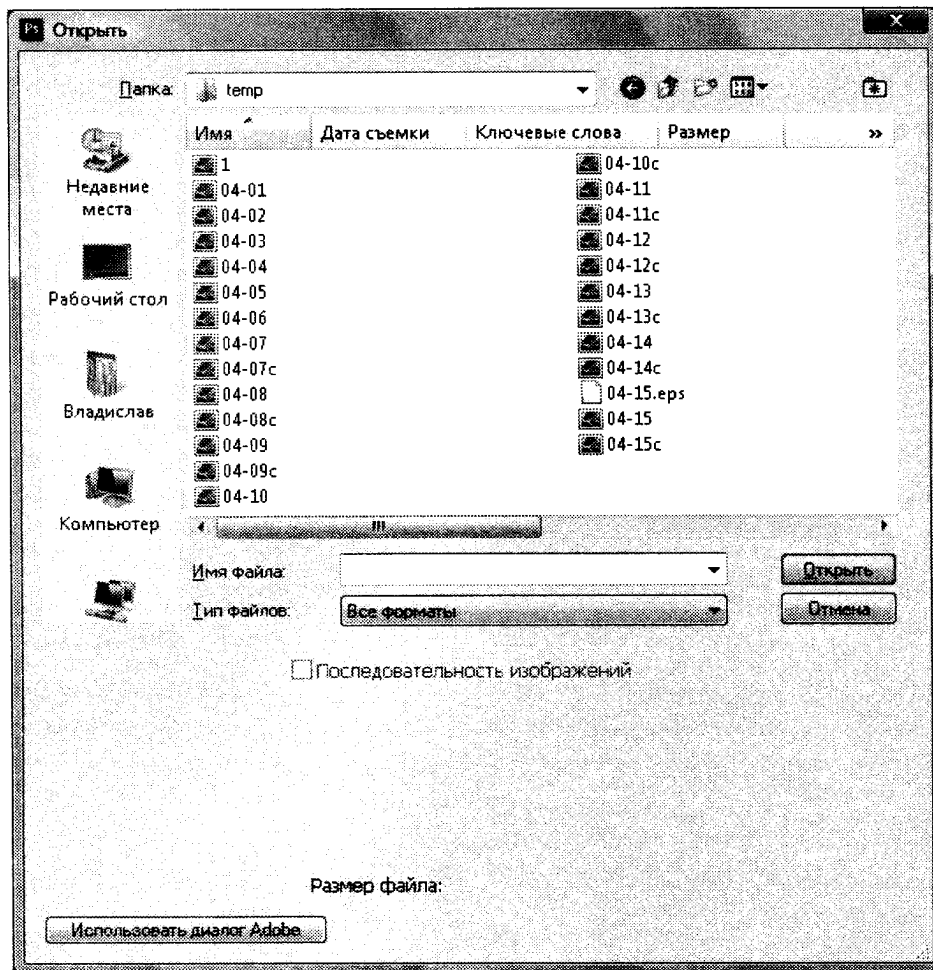


Рис. 4.14. Диалог **Открыть** (Open)

- Перейдите в этом диалоге к папке, в которой содержится нужный графический файл, и щелкните мышью на названии этого файла.
- Щелкните мышью на кнопке **Открыть** (Open), чтобы открыть файл и закрыть диалог.

После того как изображение изменено, его необходимо сохранить на жесткий диск. Если изображение раньше не сохранялось, то, чтобы сохранить изображение в программе Adobe Photoshop:

- Выберите команду меню **Файл ♦ Сохранить** (File ♦ Save). Появится диалог **Сохранить как** (Save As), Рис. 4.15.

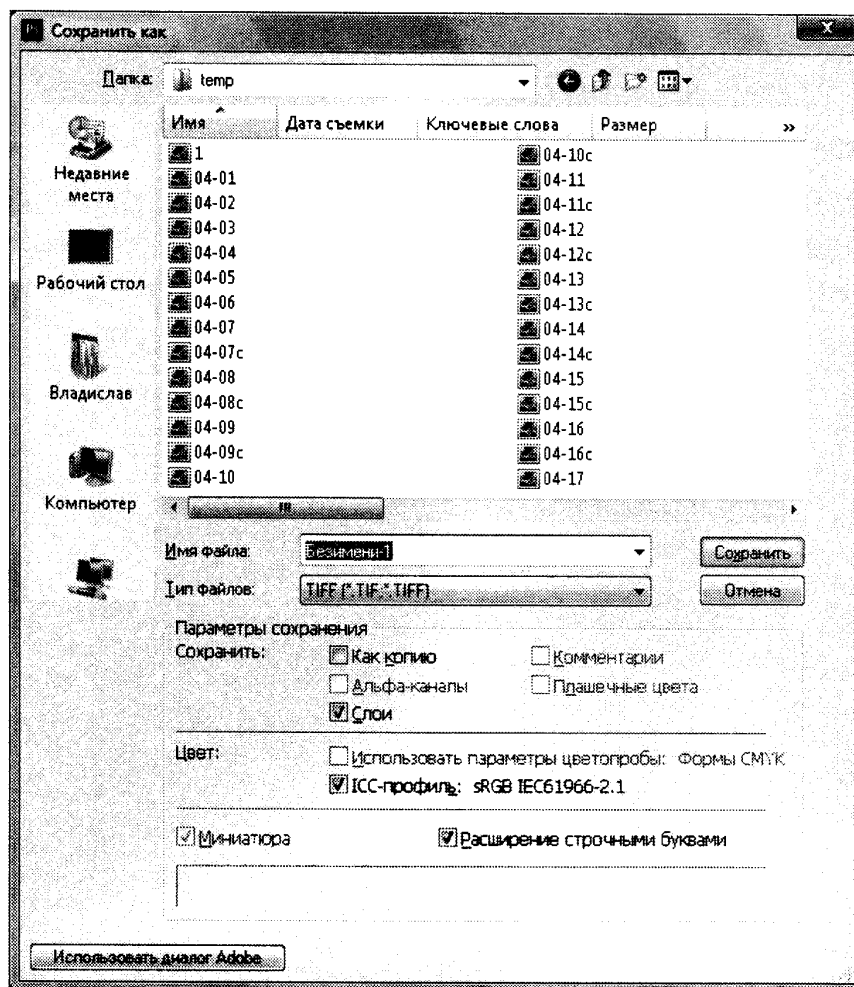


Рис. 4.15. Диалог **Сохранить как** (Save As)

- Перейдите к папке, в которой вы хотите сохранить файл.
- Щелкните мышью в поле ввода **Имя файла** (Filename) и введите название, под которым вы хотите сохранить файл изображения.
- Щелкните мышью на кнопке **Сохранить** (Save), чтобы сохранить файл и закрыть диалог.

Чтобы повторно сохранить изображение в файле, достаточно выбрать команду меню **Файл ♦ Сохранить** (File ♦ Save), изображение будет сохранено.

## Тоновая и цветовая коррекция изображений

Как правило, прежде чем размещать изображения на web-странице, необходимо провести их тоновую и цветовую коррекцию: осветлить слишком темные участки изображения, затемнить слишком светлые, немного изменить цвета, чтобы изображение было более естественным либо больше соответствовало художественному замыслу. Если на web-странице планируется использовать несколько изображений, взятых из разных источников, то желательно «подогнать» их цвета друг к другу, чтобы между ними не было дисгармонии.

В программе Adobe Photoshop есть много инструментов для работы с тоновым и цветовым балансом изображений, но мы ограничимся рассмотрением основных из них.

## Работа с диалогом Уровни (Levels)

Основным средством для тоновой коррекции, т. е. изменения соотношения светлых и темных участков изображения, в программе Adobe Photoshop является диалог **Уровни (Levels)**, Рис. 4.16.

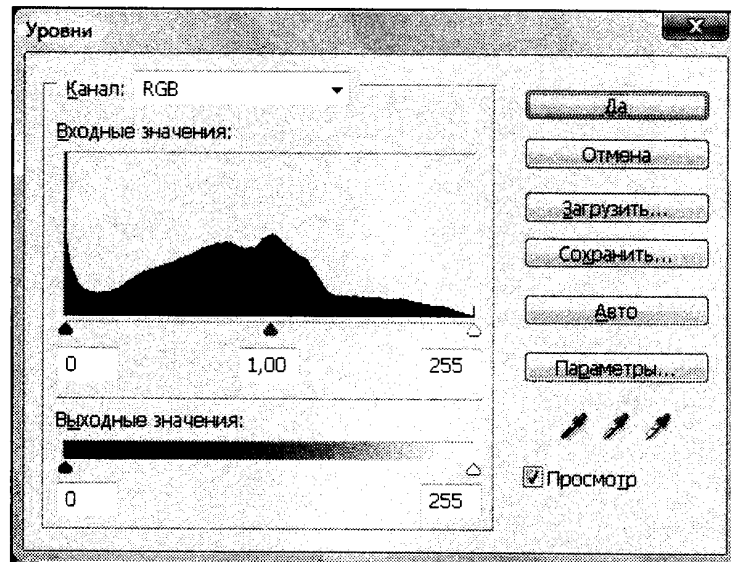


Рис. 4.16. Диалог **Уровни (Levels)**

- Чтобы открыть диалог **Уровни (Levels)**, выберите команду меню **Изображение ♦ Коррекция ♦ Уровни (Image ♦ Adjustments ♦ Levels)**.

Основную часть диалога занимает график тонового диапазона, показывающий, какая доля изображения имеет определенную яркость. Слева направо яркость увеличивается, от полностью черного на левом краю до полностью белого на правом. Левый и правый края графика называются «черной» и «белой» точками. Высота графика в определенной точке показывает, какая доля пикселей изображения имеет такую яркость.

Если на графике преобладает левая часть (Рис. 4.17), то в изображении, скорее всего, переизбыток темных тонов, если правая часть – то светлых (Рис. 4.18). В графике сбалансированного изображения будут и темные тона, и светлые (Рис. 4.16).

Корректируя тоновый баланс, учитывайте характер самого изображения. Например, если на нем представлена ночная или вечерняя сцена, то, разумеется, должны преобладать темные тона. Сцена же снежного рассвета должна быть светлой. Но для большинства изображений оптимальной является сбалансированная яркость.

В диалоге **Уровни (Levels)** представлено несколько способов тоновой коррекции изображения. Если вы установите флажок **Просмотр (Preview)**, то все ваши действия будут немедленно отображаться на самой обрабатываемой картинке.

Непосредственно под графиком тонового диапазона находятся три ползунковых регулятора – черный, серый и белый. Черный и белый регуляторы олицетворяют собой «черную» и «белую» точки изображения.

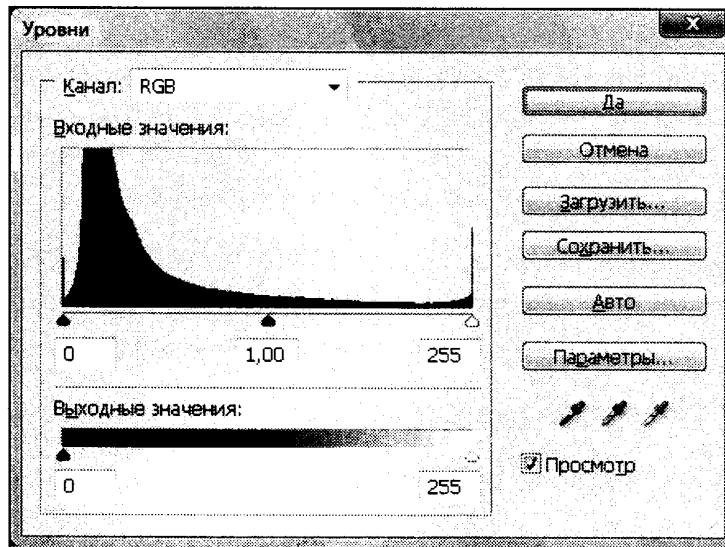


Рис. 4.17. График тонового диапазона для темного изображения

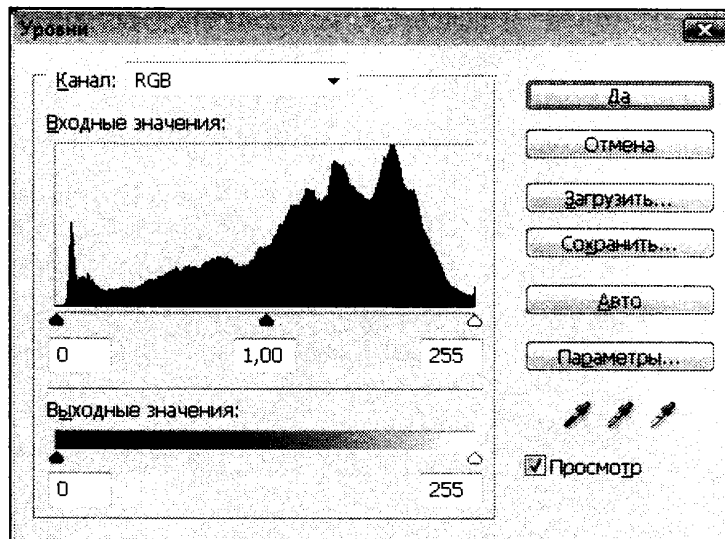





Рис. 4.18. График тонового диапазона для светлого изображения



- ✓ Если вы сдвинете черный регулятор вправо по графику, то все точки изображения, находящиеся по яркости между регулятором и левым краем графика, станут полностью черными. Яркость же остальных точек перераспределится в соответствии с новой черной точкой.
- ✓ Точно так же, перемещая влево белый регулятор, вы делаете все точки, находящиеся справа от него, белыми, а яркость остальных точек перераспределяется в соответствии с новой белой точкой.
- ✓ Серый ползунковый регулятор позволяет сдвигать баланс яркостей пикселей изображения. Место расположения этого регулятора на графике является точкой средней яркости. Сдвигая ее вправо, вы сместите баланс яркостей в темную сторону, а сдвигая влево – в светлую.

Ниже расположены три поля ввода, озаглавленные **Входные значения** (Input Levels). В первое из них можно ввести положение черного ползункового регулятора,

во второе – серого и в третье – белого. Положения черного и белого регулятора могут принимать значение яркости от «0» до «255», а положение серого регулятора задается отношением длины светлого участка на графике тонового диапазона к длине темного участка. Если значение в этом поле ввода больше единицы, то длина светлого участка больше, если же меньше, то наоборот.

Другим способом задания черной, белой и серой точек изображения является использование для этой цели самого изображения. В диалоге **Уровни (Levels)** есть три кнопки с изображением черной, серой и белой пипеток: ,  и .

Чтобы задать черную точку изображения таким способом:

- ▶ Нажмите кнопку  диалога **Уровни (Levels)**, указатель мыши приобретет вид пипетки, заполненной черной краской: .
- ▶ Щелкните мышью на точке обрабатываемого изображения, уровень яркости которой хотите приравнять к черной точке. График в диалоге **Уровни (Levels)** изменится в соответствии с новой черной точкой.
- ▶ Если вас не устроил результат, повторите предыдущее действие.

Если вы хотите отменить выбор черной точки:

- ▶ Нажмите клавишу **[Alt]**. Кнопка **Отмена (Cancel)** диалога **Уровни (Levels)** сменится на кнопку **Восстановить (Reset)**, как представлено на Рис. 4.19.

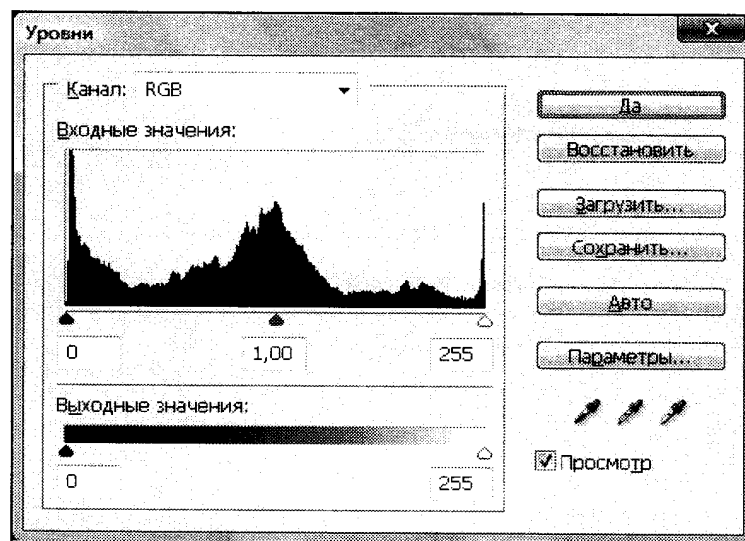




Рис. 4.19. Диалог **Levels (Уровни)** при нажатой клавише **[Alt]**



- ▶ Не отпуская клавишу **[Alt]**, щелкните мышью на кнопке **Восстановить (Reset)** диалога **Уровни (Levels)**.

Чтобы задать белую точку изображения:



- ▶ Нажмите кнопку  диалога **Уровни (Levels)**, указатель мыши приобретет вид пипетки, заполненной белой краской: .

- Щелкните мышью на точке обрабатываемого изображения, уровень яркости которой хотите приравнять к белой точке. График в диалоге **Уровни** (Levels) изменится в соответствии с новой белой точкой.
- Если вас не устроил результат, повторите предыдущее действие.



Если вы хотите отменить выбор белой точки:

- Нажмите клавишу . Кнопка **Отмена** (Cancel) диалога **Уровни** (Levels) сместится на кнопку **Восстановить** (Reset).
- Не отпуская клавишу , щелкните мышью на кнопке **Восстановить** (Reset) диалога **Уровни** (Levels).

Чтобы задать серую точку изображения:



- Нажмите кнопку  диалога **Уровни** (Levels), указатель мыши приобретет вид пипетки, заполненной серой краской: .
- Щелкните мышью на точке обрабатываемого изображения, уровень яркости которой хотите сделать средним уровнем яркости. График в диалоге **Уровни** (Levels) изменится в соответствии с новой серой точкой.
- Если вас не устроил результат, повторите предыдущее действие.

Если вы хотите отменить выбор серой точки:

- Нажмите клавишу . Кнопка **Отмена** (Cancel) диалога **Уровни** (Levels) сместится на кнопку **Восстановить** (Reset).
- Не отпуская клавишу , щелкните мышью на кнопке **Восстановить** (Reset) диалога **Уровни** (Levels).

В диалоге **Уровни** (Levels) предусмотрена возможность автоматически устанавливать параметры тоновой коррекции изображения. Все, что нужно для этого сделать – щелкнуть мышью на кнопке **Авто** (Auto) этого диалога. После чего автоматически определяются самая темная и самая светлая области изображения, они принимаются за черную и белую точку изображения, а яркости остальных точек изображения равномерно распределяются между ними.

Если вам не понравится полученный результат и вы захотите его отменить:

- Нажмите клавишу . Кнопка **Отмена** (Cancel) диалога **Уровни** (Levels) сместится на кнопку **Восстановить** (Reset).
- Не отпуская клавишу , щелкните мышью на кнопке **Восстановить** (Reset) диалога **Уровни** (Levels).

До сих пор мы узнавали, как с помощью диалога **Уровни** (Levels) расширить тоновый диапазон изображения – чтобы в нем присутствовали и светлые участки, вплоть до белого, и темные – до черного. Но встречается и обратная задача – «приглушить» изображение, убрать самые яркие участки, сделать самые светлые точки темнее, а темные – светлее. Для этого в диалоге **Уровни** (Levels) существует группа элементов

управления **Выходные значения** (Output Levels), состоящая из двух полей ввода и двух ползунковых регуляторов, черного и белого. Эти регуляторы располагаются под полосой уровней яркости: от минимальной – слева, до максимальной – справа.

Черный регулятор указывает на полосе уровней яркости минимальный уровень яркости для изображения. Передвигая этот регулятор вправо, вы делаете черную точку изображения уже не черной, а все более светлой. Точно так же, передвигая влево белый регулятор, вы делаете темнее белую точку изображения.

Минимальную и максимальную яркости изображения также можно задать, вводя непосредственные значения яркости в поля ввода **Выходные значения** (Output Levels). В левое поле вводится минимальный уровень яркости изображения, а в правое – максимальный.

- Настроив все параметры тоновой коррекции, примите изменения в изображении и закройте диалог **Levels** (Уровни), щелкнув мышью на кнопке **ОК**.

### Диалог Яркость/Контрастность (Brightness/Contrast)

Более простым средством тоновой коррекции в программе Adobe Photoshop служит диалог **Яркость/Контрастность** (Brightness/Contrast), Рис. 4.20.

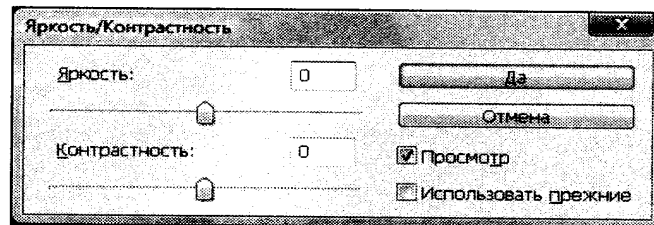


Рис. 4.20. Диалог **Яркость/Контрастность** (Brightness/Contrast)

- Чтобы открыть этот диалог, выберите команду меню **Изображение ♦ Коррекция ♦ Яркость/Контрастность** (Image ♦ Adjustments ♦ Brightness/Contrast).

Диалог содержит всего два ползунковых регулятора, продублированных полями ввода. Значения обоих полей могут изменяться в диапазоне от -100 до 100. Сдвигание ползункового регулятора вправо увеличивает значение в поле ввода, сдвигание регулятора влево – уменьшает.

- ✓ Если вы установите флажок **Просмотр** (Preview), то все изменения, произведенные в диалог **Яркость/Контрастность** (Brightness/Contrast), будут немедленно отображаться на самом рисунке.
- ✓ С помощью первого регулятора **Яркость** (Brightness) можно установить общую яркость изображения. Сдвигая регулятор вправо, вы увеличиваете яркость всех точек изображения, сдвигая влево – уменьшаете.
- ✓ Ползунковый регулятор **Контрастность** (Contrast) устанавливает контрастность изображения. Контрастность – это различие в яркости между самой яркой и самой темной точками изображения. Сдвигая регулятор вправо, вы контрастность увеличиваете, сдвигая влево – уменьшаете.

- Выставив уровень яркости и контрастности изображения, щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

## Диалог Цветовой баланс (Color Balance)

Важную роль в подготовке изображений для использования на web-страницах играет цветовая коррекция – изменение баланса цветов в изображении. Цветовая коррекция служит как для исправления дефектов изображения – например неправильных условий съемки фотографий, так и для реализации различных художественных замыслов – например, проведя небольшую цветовую коррекцию, можно превратить обычную земную пустыню в таинственный венерианский пейзаж.

Одним из средств цветовой коррекции программы Adobe Photoshop является диалог **Цветовой баланс** (Color Balance), Рис. 4.21.

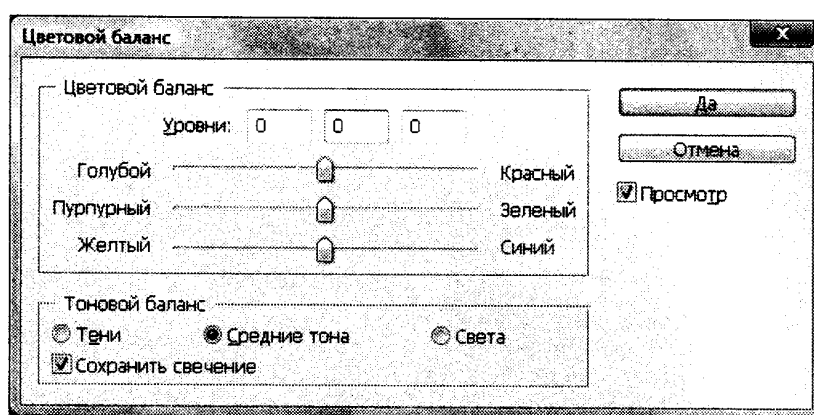


Рис. 4.21. Диалог **Цветовой баланс** (Color Balance)

- Чтобы открыть этот диалог, выберите команду меню **Изображение ♦ Коррекция ♦ Цветовой баланс** (Image ♦ Adjustments ♦ Color Balance).

Если вы установите флажок **Просмотр** (Preview), то все изменения, произведенные в диалог **Цветовой баланс** (Color Balance), будут немедленно отображаться на самом рисунке.

Диалог **Цветовой баланс** (Color Balance) разделен на две группы элементов управления. В группе **Цветовой баланс** (Color Balance) собраны элементы, управляющие непосредственно балансом цветов. Группа **Тоновый баланс** (Tone Balance) отвечает за то, к какому диапазону яркостей изображения применяется цветовая коррекция.

- ✓ В группе **Тоновый баланс** (Tone Balance) располагается переключатель с тремя положениями: **Тени** (Shadows), **Средние тона** (Midtones) и **Света** (Highlights). Установив переключатель в положение **Тени** (Shadows), вы получаете возможность изменять баланс цветов теней изображения. Установив его в положение **Средние тона** (Midtones), вы можете регулировать баланс цветов, точек изображения средней яркости, а переключив в положение **Света** (Highlights) – можете работать с цветами самых ярких точек изображения. Установка флажка **Сохранить свечение** (Preserve Luminosity) воспрепятствует изменению тонового диапазона изображения при корректировании его цветового баланса.



- ✓ Группа элементов управления **Цветовой баланс** (Color Balance) состоит из трех ползунковых регуляторов, перемещая которые, вы изменяете цветовой баланс изображения. Первый регулятор, **Голубой-красный** (Cyan-Red), отвечает за баланс между синим и красным цветами. Второй – **Пурпурный-зеленый** (Magenta-Green) – за баланс между пурпурным и зеленым цветами. И третий – **Желтый-синий** (Yellow-Blue) – за баланс между желтым и синим цветами. Перемещая регулятор в сторону одного цвета, вы увеличиваете долю этого цвета в изображении и, вместе с тем, уменьшаете долю цвета, находящегося с ним в паре.
- Завершив настройку баланса цветов изображения, щелкните мышью на кнопке **ОК**, чтобы принять изменения и закрыть диалог.

## Диалог Варианты (Variations)

Удобным и простым средством регулирования яркости / контрастности изображения совместно с цветовой коррекцией является диалог **Варианты** (Variations), Рис. 4.22.

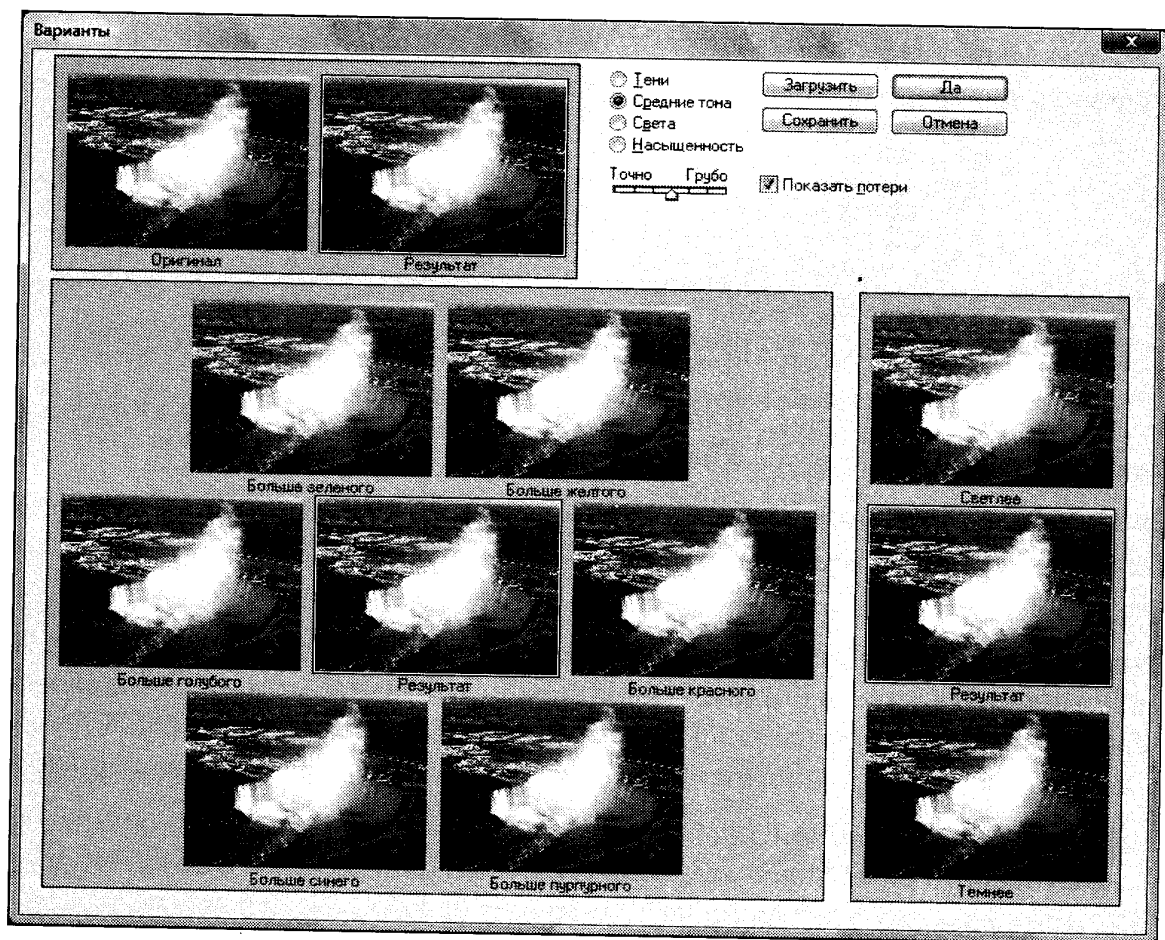


Рис. 4.22. Диалог **Варианты** (Variations)

- Чтобы открыть этот диалог, выберите команду меню **Изображение ♦ Коррекция ♦ Варианты** (Image ♦ Adjustments ♦ Variations).

В верхней части диалога **Варианты** (Variations) представлены две уменьшенные копии изображения: **Оригинал** (Original) и **Результат** (Current Pick).

- ✓ Картинка **Оригинал** (Original) показывает, как выглядело изображение до обработки. Щелкнув мышью на этой картинке, вы сбросите все настройки, произведенные в диалоге **Варианты** (Variations).
- ✓ Картинка **Результат** (Current Pick) показывает, как будет выглядеть изображение после принятия всех изменений, сделанных в диалоге **Variations** (Варианты).

В основной части диалога находится еще несколько уменьшенных версий изображения, с небольшими изменениями относительно его текущего состояния. Щелкнув мышью по одному из них, вы примете измененную версию изображения, сделав ее текущей. После этого все версии изображения опять обновятся.

Перечислим все картинки в основной части диалога по очереди:

- ✓ **Больше зеленого** (More Green). Версия с усилением зеленого цвета.
- ✓ **Больше желтого** (More Yellow). Версия с усилением желтого цвета.
- ✓ **Больше голубого** (More Cyan). Версия с усилением голубого цвета.
- ✓ **Больше красного** (More Red). Версия с усилением красного цвета.
- ✓ **Больше синего** (More Blue). Версия с усилением синего цвета.
- ✓ **Больше пурпурного** (More Magenta). Версия с усилением пурпурного цвета.
- ✓ **Светлее** (Lighter). Более светлая версия изображения.
- ✓ **Темнее** (Darker). Более темная версия изображения.

Справа от картинок **Оригинал** (Original) и **Результат** (Current Pick) находятся несколько элементов управления:

- ✓ Переключатели **Тени** (Shadows), **Средние тона** (Midtones) и **Света** (Highlights). Установив один из них, вы будете регулировать баланс цветов теней изображения, его средних тонов или самых светлых участков. При установке переключателя в положение **Насыщенность** (Saturation) количество изображений в центральной части диалога уменьшится. Этот режим предназначен для изменения насыщенности цветов изображения. Щелкая мышью на картинке **Меньше насыщенности** (Smaller Saturation), вы будете понижать насыщенность, щелкая мышью на картинке **Больше насыщенности** (More Saturation) – повышать.
- ✓ Ползунковый регулятор **Точно-грубо** (Fine-Coarse). Передвигая его влево, вы уменьшаете «дозировку» изменений изображения, делая его настройку более точной, но и более медленной. Передвигая же регулятор вправо, вы, напротив, увеличиваете «шаг», делая настройку более быстрой и грубой.
- ✓ Флажок **Показать потери** (Show Clipping). Если установить этот флажок, то на версиях изображения контрастными неоновыми цветами будут указываться области, которые после принятия версии уйдут из изображения, станут чистого белого или черного цвета.
- Закончив настройку цветового баланса и яркости изображения, примите изменения и закройте диалог, щелкнув мышью на кнопке **Да**.

## Редактирование изображений



Завершив тоновую и цветовую коррекцию изображения, можно приступить к редактированию его композиции. Рассмотрим по порядку важнейшие операции, наиболее часто выполняемые при работе над композицией.

### Кадрирование

Важным этапом в обработке изображений, особенно фотографий, является кадрирование – определение новых границ изображения отсечением его краев.

Кадрирование компьютерных изображений проводится в двух разных случаях: когда необходимо обрезать картинку под определенный формат и в художественных целях – отрезание второстепенных деталей, выстраивание композиции кадра и т. д.

В программе Adobe Photoshop для проведения кадрирования предназначен инструмент **Рамка** (Crop). Чтобы откадрировать с его помощью изображение:

- Щелкните мышью на кнопке **Рамка** (Crop)  на панели инструментов программы Adobe Photoshop. Указатель мыши примет следующий вид – .

На контекстной панели настроек можно установить некоторые параметры работы инструмента **Рамка** (Crop), Рис. 4.23. В поля ввода **Ширина** (Width) и **Высота** (Height) можно ввести размеры кадрируемой области. Высота и ширина области кадрирования будут пропорциональны заданным размерам, а итоговое изображение будет иметь точно такие же размеры. Чтобы сбросить установки параметров **Ширина** (Width) и **Высота** (Height), щелкните мышью на кнопке **Очистить** (Clear).

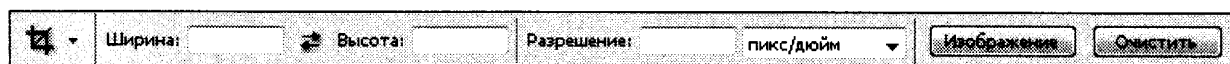


Рис. 4.23. Контекстная панель настроек инструмента **Рамка** (Crop)

- Переместите указатель мыши к той точке изображения, где будет один из углов кадрируемой области. Нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши к противоположному углу кадрируемой области.
- Отпустите левую кнопку мыши. Кадрируемая область будет окружена бегущей рамкой с маркерами, а оставшаяся часть изображения, которая будет обрезана, станет затененной (Рис. 4.24).

Пока края окончательно не обрезаны, кадрируемую область можно подредктировать.

Чтобы передвинуть кадрируемую область:

- Переместите указатель мыши на область кадрирования, затем нажмите левую кнопку мыши и, не отпуская ее, перетащите область в другое место.
- Отпустите левую кнопку мыши. Область кадрирования будет передвинута.

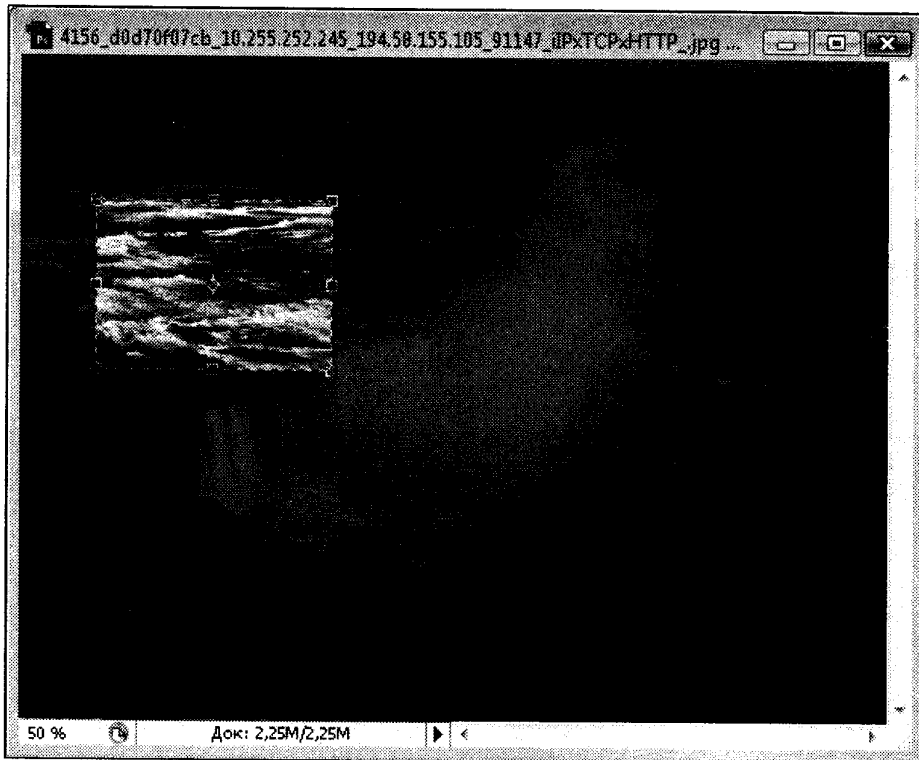


Рис. 4.24. Выделение кадрируемой области на изображении

Если вы хотите изменить размеры области кадрирования:

- Переместите указатель мыши на один из маркеров, находящихся по краю области кадрирования. Нажмите левую кнопку мыши и, не отпуская ее, переместите указатель мыши в другую позицию. Размеры области кадрирования будут изменяться.
- Отпустите левую кнопку мыши, размер области кадрирования будет изменен.
- Закончив настройку размеров и положения области кадрирования, нажмите клавишу **Enter**, либо щелкните мышью на кнопке . Изображение будет откадрировано.

## Поворот изображений

Часто возникает необходимость повернуть изображение на определенный угол, либо просто развернуть его. Для поворота изображений в программе Adobe Photoshop служит группа команд **Повернуть холст** (Rotate Canvas).

- ✓ Если вам необходимо повернуть изображение на 180°, следует выполнить команду меню **Изображение ♦ Повернуть холст ♦ 180°** (Image ♦ Rotate Canvas ♦ 180°).
- ✓ Чтобы повернуть изображение на 90° по часовой стрелке, следует выбрать команду меню **Изображение ♦ Повернуть холст ♦ 90° по часовой** (Image ♦ Rotate Canvas ♦ 90° CW).
- ✓ Чтобы повернуть изображение на 90° против часовой стрелки, следует выполнить команду меню **Изображение ♦ Повернуть холст ♦ 90° против часовой** (Image ♦ Rotate Canvas ♦ 90° CCW).

- ✓ Чтобы отразить изображение по горизонтали, следует выбрать команду меню **Изображение ♦ Повернуть холст ♦ Отразить холст по горизонтали** (Image ♦ Rotate Canvas ♦ Flip Canvas Horizontal).
- ✓ Чтобы развернуть изображение по горизонтали, следует выполнить команду меню **Изображение ♦ Повернуть холст ♦ Отразить холст по вертикали** (Image ♦ Rotate Canvas ♦ Flip Canvas Vertical).

Если вам понадобилось повернуть изображение на произвольный угол, не кратный 90°, сделайте следующее:

- Выберите команду меню **Изображение ♦ Повернуть холст ♦ Произвольно** (Image ♦ Rotate Canvas ♦ Arbitrary). Откроется диалог **Повернуть холст** (Rotate Canvas), представленный на Рис. 4.25.

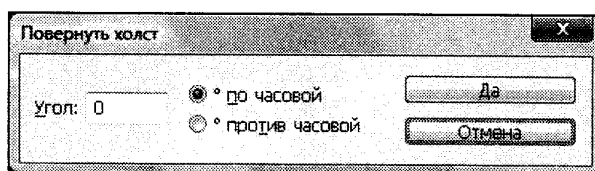


Рис. 4.25. Диалог **Повернуть холст** (Rotate Canvas)

- Щелкните мышью в поле ввода **Угол** (Angle) и введите угол, на который хотите повернуть изображение.
- Установите переключатель в положение **По часовой** (CW), чтобы поворот шел в сторону по часовой стрелке, или в положение **Против часовой** (CCW), чтобы поворачивать изображение против часовой стрелки.
- Щелкните мышью на кнопке **Да**, чтобы повернуть изображение и закрыть диалог.

## Преобразование цветных изображений в оттенки серого цвета

Иногда требуется преобразовать цветное изображение в изображение, состоящее из оттенков серого цвета. Проведем такое преобразование:

- Выберите команду меню **Изображение ♦ Коррекция ♦ Обесцветить** (Image ♦ Adjustments ♦ Decolorize). Изображение будет преобразовано.

## Изменение размеров изображений

Одной из последних стадий подготовки изображения к размещению его на web-странице является установка подходящего размера изображения.

Размеры изображения в программе Adobe Photoshop изменяются с помощью диалога **Размер изображения** (Image Size), Рис. 4.26.

- Чтобы открыть этот диалог, выполните команду меню **Изображение ♦ Размер изображения** (Image ♦ Image Size).

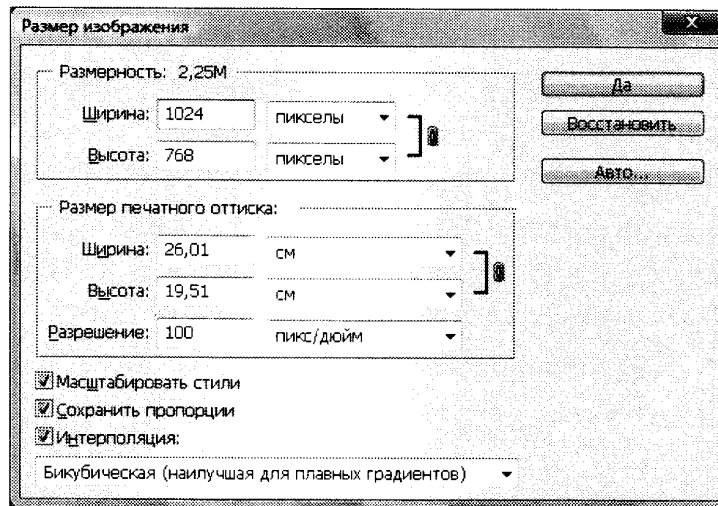


Рис. 4.26. Диалог **Размер изображения** (Image Size)

В группу элементов управления **Размерность** (Pixel Dimensions) сведены настройки, управляющие размером изображения в пикселях. В этой группе находятся два поля ввода, **Ширина** (Width) и **Высота** (Height), в которые вводятся желаемые высота и ширина изображения. Справа от этих полей находятся открывающиеся списки, состоящие из двух пунктов: **Пиксели** (pixels) и проценты. Если выбран пункт **Пиксели** (pixels), то в поле ввода размеры указываются в точках, если же выбран знак процента, — то в процентах от текущего размера изображения.

Если установлен флажок **Сохранить пропорции** (Constrain Proportions), то размеры изображения будут изменяться пропорционально: изменение ширины картинки будет вести к соответствующему изменению ее высоты, и наоборот.

В открывающемся списке **Интерполяция** (Resample Image) можно выбрать способ, каким будут пересчитываться пиксели изображения при изменении его размеров. При увеличении размеров изображения программа **Adobe Photoshop** рассчитывает цвета добавляемых пикселей изображения на основе цветов, уже существующих. Программа **Adobe Photoshop** поддерживает несколько способов такого перерасчета, называемого интерполяцией:

- ✓ **По соседним пикселям** (Nearest Neighbor). Самый быстрый и наименее точный способ интерполяции.
- ✓ **Билинейная** (Bilinear). Немного более точный способ интерполяции, но работающий несколько медленнее.
- ✓ **Bicubic** (Бикубическая). Бикубическая интерполяция. Самый точный и медленный способ, дающий наилучший результат.
- ✓ **Бикубическая глаже** (Bicubic Smoother). Способ аналогичен бикубической интерполяции, но в результате получается более сглаженное изображение.
- ✓ **Бикубическая четче** (Bicubic Sharper). Способ аналогичен бикубической интерполяции, но в результате получается более резкое изображение.
- Проведя все необходимые настройки, щелкните мышью на кнопке **Да**, чтобы изменить размер изображения и закрыть диалог.

## Увеличение резкости изображения

После различных манипуляций с изображением оно теряет часть резкости, становится несколько расплывчатым и нечетким. Для улучшения резкости изображений в программе Adobe Photoshop есть группа команд **Резкость** (Sharpen).

- ✓ Команда **Контурная резкость** (Outline Sharpen) повышает резкость изображения путем увеличения контрастности вдоль его краев. Запускается она выполнением команды меню **Фильтр ♦ Резкость ♦ Контурная резкость** (Filter ♦ Sharpen ♦ Outline Sharpen).
- ✓ В фильтре **Умная резкость** (Clever Sharpen) предусмотрены элементы управления резкостью, недоступные в фильтре **Контурная резкость** (Outline Sharpen). В нем можно задать алгоритм регулировки резкости или установить интенсивность резкости в затененных областях и в областях светов. Открыть окно фильтра **Умная резкость** (Clever Sharpen) можно, выбрав команду меню **Фильтр ♦ Резкость ♦ Умная резкость** (Filter ♦ Sharpen ♦ Clever Sharpen).
- ✓ Команда **Резкость** (Sharpen) увеличивает резкость изображения за счет усиления контраста между соседними точками изображения. Запускается она выполнением команды меню **Фильтр ♦ Резкость ♦ Резкость** (Filter ♦ Sharpen ♦ Sharpen).
- ✓ Команда **Резкость +** (Sharpen More) увеличивает резкость таким же способом, как и команда **Sharpen** (Резкость), но резкость увеличивается на большую величину. Запускается она выполнением команды меню **Фильтр ♦ Резкость ♦ Резкость +** (Filter ♦ Sharpen ♦ Sharpen More).
- ✓ Команда **Резкость на краях** (Sharpen Edges) увеличивает резкость не всего изображения, а только границ с резкими переходами цветов. Запускается она выполнением команды меню **Фильтр ♦ Резкость ♦ Резкость на краях** (Filter ♦ Sharpen ♦ Sharpen Edges).

В том случае, когда однократного применения одной из команд увеличения резкости недостаточно, ее можно применить повторно.

## Оптимизация изображений для размещения их в web

Когда изображение готово к размещению на web-странице, необходимо подготовить его к размещению в web. Для этого изображение необходимо сохранить в файле формата **GIF** или **JPEG**, чтобы при этом они занимали как можно меньше места, с минимальными потерями качества. Специально для такой оптимизации изображений в программе Adobe Photoshop предназначен диалог **Сохранить для Web и устройств** (Save For Web and devices), Рис. 4.27.

Открыть этот диалог можно, выбрав команду меню **File ♦ Сохранить для Web и устройств** (File ♦ Save For Web and devices).

Основную часть диалога занимает область предварительного просмотра изображения. Щелкая по ярлыкам над ним, вы можете выбрать вкладки с разными способами организации области предварительного просмотра.

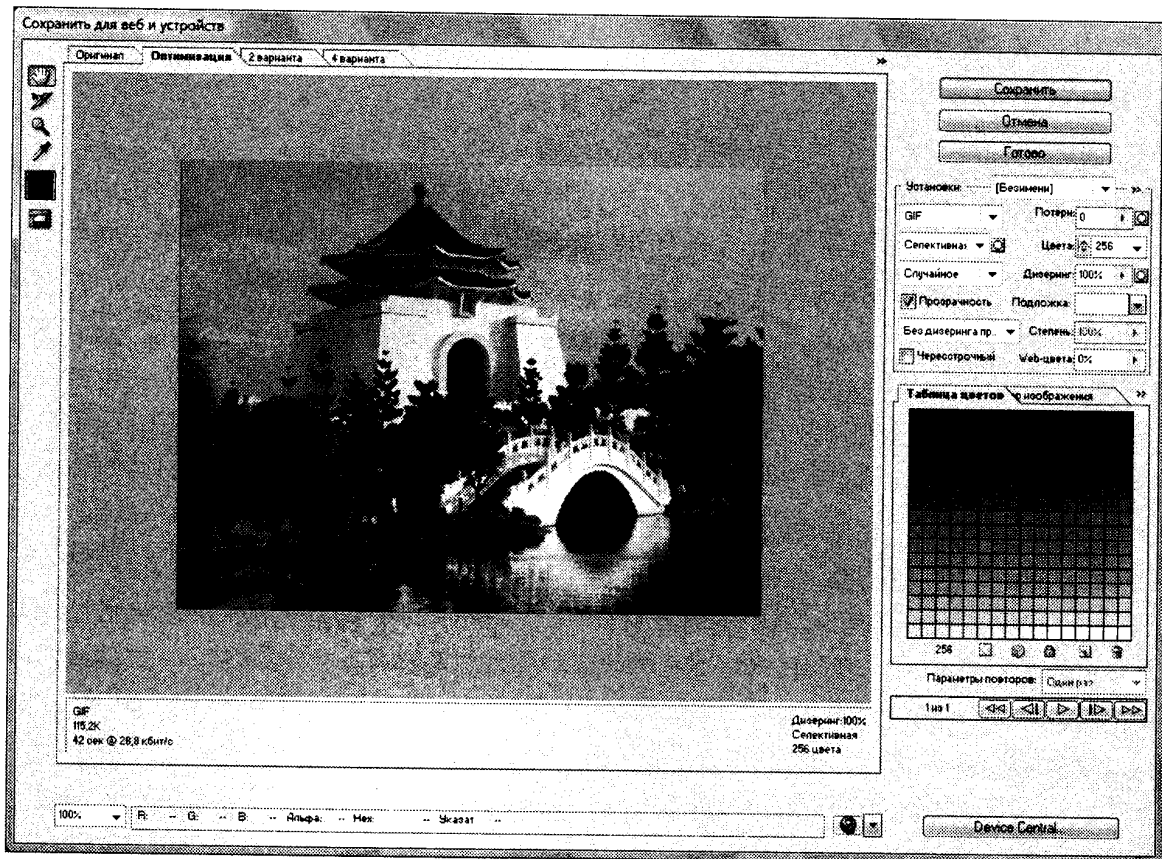


Рис. 4.27. Диалог **Сохранить для Web и устройств** (Save For Web and devices)

- ✓ Во вкладке **Оригинал** (Original) представлено исходное изображение до его оптимизации.
- ✓ Вкладка **Оптимизация** (Optimized), представленная на Рис. 4.27. Будет отображаться оптимизированное изображение.
- ✓ Вкладка **Два варианта** (2-Up), представленная на Рис. 4.28. В диалоге будет показано два варианта изображения: до оптимизации и после. Этот режим полезен для сравнения исходной картинке с тем, что получается после оптимизации.
- ✓ Вкладка **Четыре варианта** (4-Up), представленная на Рис. 4.29. Будет отображено исходное изображение и три его варианта с разными настройками оптимизации. В этом режиме удобно сравнивать влияние разных параметров оптимизации и разных форматов файлов на внешний вид картинке.

В том случае, когда в диалоге отображаются два или четыре изображения, чтобы настроить параметры оптимизации того или иного варианта изображения, следует щелкнуть на нем мышью. Сохраняться, в итоге, тоже будет только выбранный вариант изображения.

В правой части диалога сосредоточены настройки параметров оптимизации изображения. Количество и вид элементов настройки зависят от выбранного формата файла.

Формат файла, в который сохраняется оптимизированное изображение, выбирается в открывающемся списке **Формат оптимизированного файла** (Optimized file format).



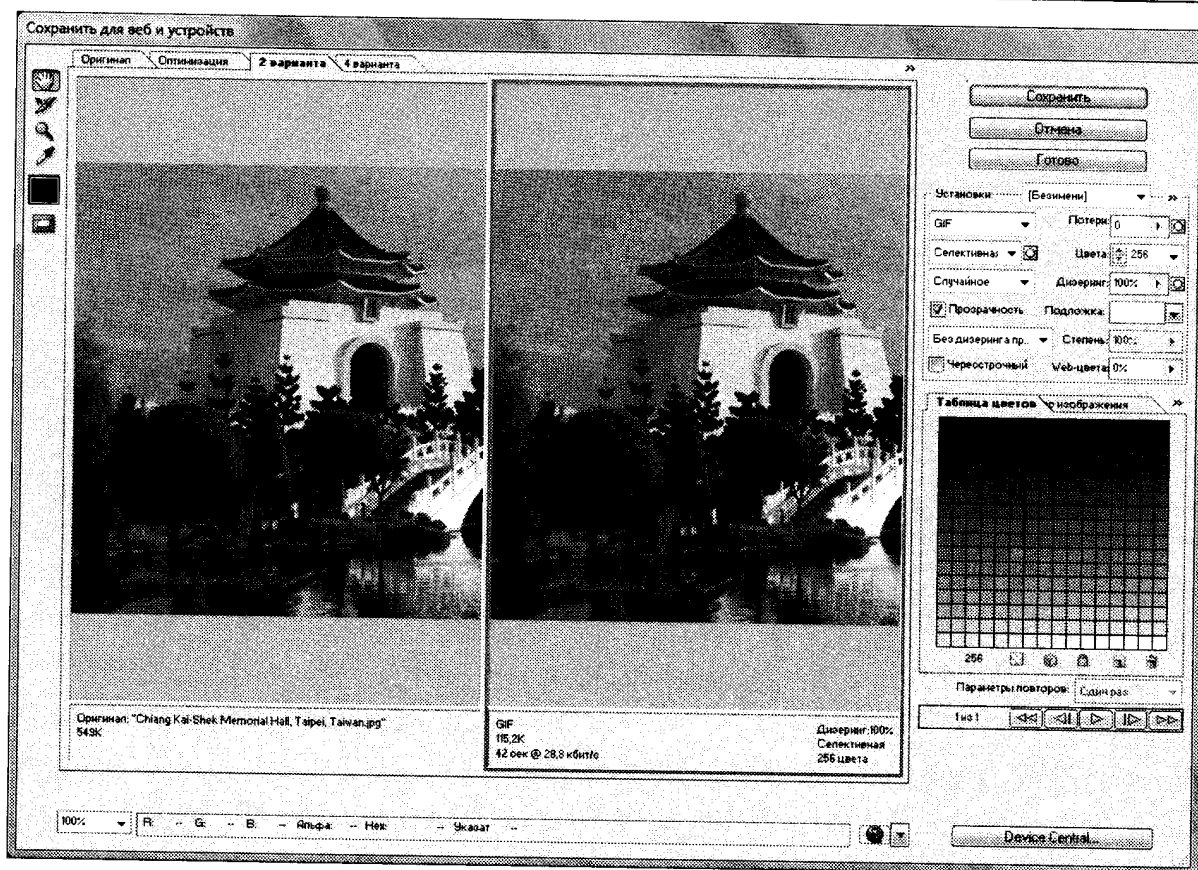


Рис. 4.28. Вкладка **Два варианта** (2-Up) диалога **Сохранить для веб и устройств** (Save For Web and devices)

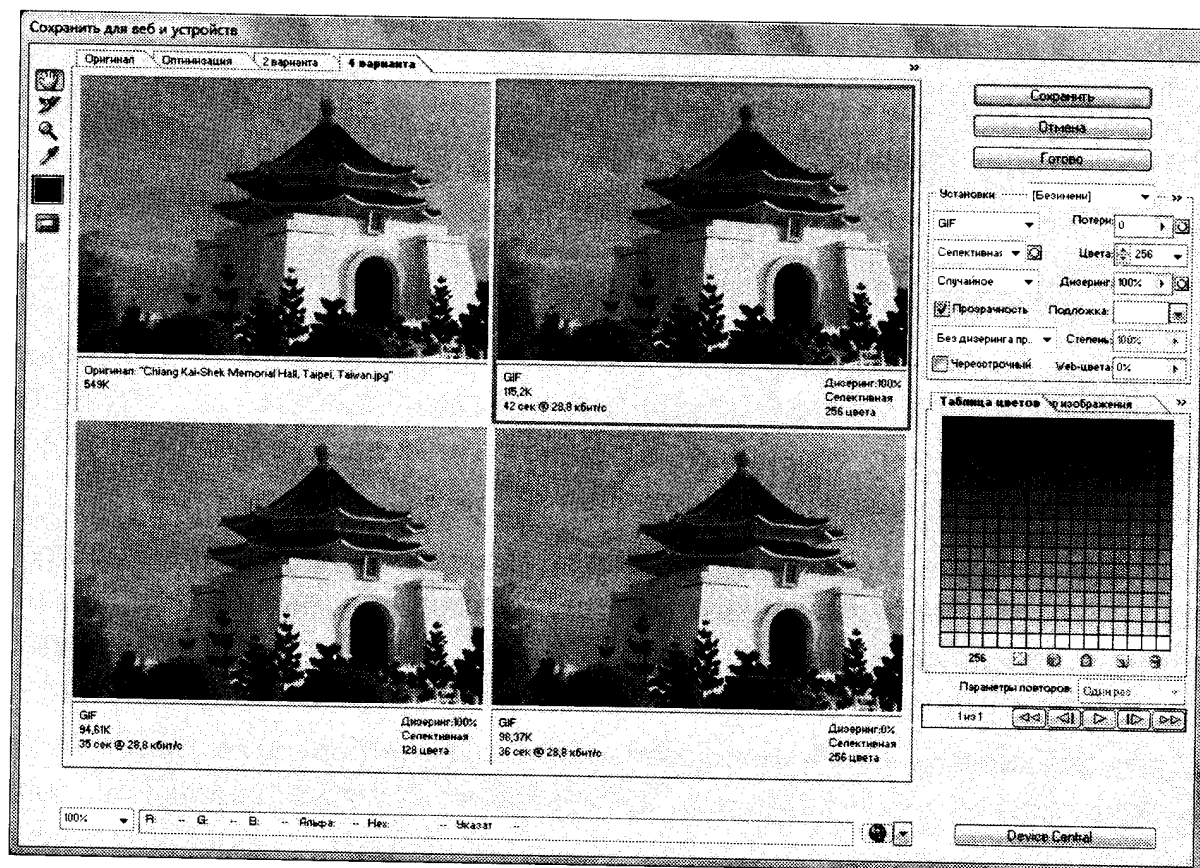


Рис. 4.29. Вкладка **Четыре варианта** (4-Up) диалога **Сохранить для веб и устройств** (Save For Web and devices)

Если выбран формат файла **JPEG** (Рис. 4.30), то доступны следующие параметры оптимизации:

- ✓ Флажок **Оптимизированное** (Optimized). Установка этого флажка включает дополнительную оптимизацию изображения, которая приводит к уменьшению файла без потери качества изображения.
- ✓ Поле ввода со счетчиком **Качество** (Quality). Отвечает за качество изображения. Значение может изменяться от 0 до 100. Чем больше число, тем выше качество изображения и тем больше места занимает файл.
- ✓ Поле ввода со счетчиком **Размытие** (Blur). Степень размытия может меняться от 0 до 2. Чем больше размытие, тем меньше места занимает файл, но тем расплывчатее становится изображение.
- ✓ Флажок **Постепенно** (Progressive). Установка этого флажка включает в файл несколько версий изображения с уменьшенным разрешением. При загрузке через web сначала будет открыта самая маленькая версия, потом версия размером побольше, и так далее, до самой большой. Визуально увеличивает скорость загрузки изображения, но немного увеличивает размер файла.

Для формата **GIF** (Рис. 4.31) предлагаются другие настройки:

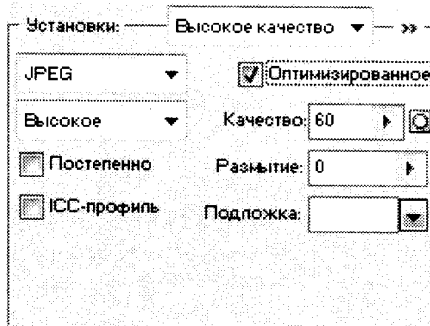


Рис. 4.30 Параметры настройки файла формата **JPEG**

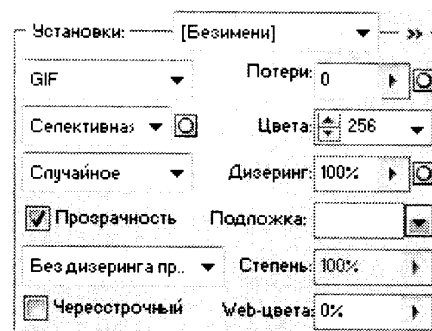


Рис. 4.31 Параметры настройки файла формата **GIF**

Прямо под открывающимся списком выбора форматов находится открывающийся список **Модель редукции цвета** (Color reduction algorithm). В этом списке можно выбрать способ, каким количество цветов изображения уменьшается до палитры **GIF**. Возможны следующие варианты:

- ✓ **Перцепционная** (Perceptual). Цвета выбираются с учетом особенностей человеческого восприятия.
- ✓ **Селективная** (Selective). Цвета выбираются также с учетом особенностей человеческого восприятия, но предпочтение отдается цветам из палитры безопасных web-цветов. Речь о том, что это за безопасные цвета, пойдет чуть позже.
- ✓ **Адаптивная** (Adaptive). Выбираются цвета, наиболее часто встречающиеся в изображении.

- ✓ **Ограниченная (Web) (Restrictive (Web))**. Используются только безопасные web-цвета.
- ✓ **Заказное (Customize)**. Используется палитра цветов, созданная или измененная пользователем.
- ✓ **Градации серого (Grayscale)**. Цвета изображения преобразовываются в оттенки серого цвета.
- ✓ **Черный и белый (Black & White)**. Палитра состоит из двух цветов – черного и белого.

Чуть ниже находится открывающийся список **Определение модели дизеринга прозрачности (Specify the dither algorithm)**. В этом списке выбирается способ, каким будут имитироваться цвета, отсутствующие в палитре итогового изображения. Возможны следующие варианты:

- ✓ **Без дизеринга прозрачности (No Dither)**. Цвета не имитируются, а просто замещаются ближайшим похожим цветом из палитры.
- ✓ **Случайный дизеринг прозрачности (Diffusion)**. Цвета имитируются смешением цветов нескольких соседних точек, смешанных в случайном порядке.
- ✓ **Регулярный дизеринг прозрачности (Pattern)**. Цвета имитируются смешением цветов соседних точек, выстроенных в определенном порядке.
- ✓ **Шумовой дизеринг прозрачности (Noise)**. Похож на режим **Случайный дизеринг прозрачности (Diffusion)**, но несколько другой способ размещения точек.

Перечислим остальные параметры формата **GIF**:

- ✓ **Чересстрочный (Interlaced)**. Установка этого флажка включает чересстрочный режим загрузки **GIF**-файла. Сначала загружается половина строк, через одну, затем вторая половина. Это создает видимость ускорения загрузки.
- ✓ Поле ввода со счетчиком **Потери (Lossy)**. Позволяет несколько уменьшить размер файла за счет уменьшения качества изображения. Величина в этом поле может изменяться от 0 до 100. 0 – отсутствие ухудшения качества, 100 – максимальное ухудшение качества и минимальный размер файла.
- ✓ Поле ввода со счетчиком **Colors (Цвета)**. В этом поле выбирается, какое количество цветов будет в палитре итогового изображения. Чем меньше цветов, тем меньше места занимает файл и тем хуже выглядит изображение. Количество цветов может быть от 2 до 256.
- ✓ Поле ввода со счетчиком **Дизеринг (Dither)**. Определяет степень имитации цветов, не вошедших в палитру. Значение в поле может быть от 0 до 100%. 0% – отсутствие сглаживания, 100% – максимальное сглаживание.
- ✓ Поле ввода со счетчиком **Web-цвета (Web-Snap)**. Степень соответствия цветовой палитры изображения палитре безопасных web-цветов. Может изменяться от 0 до 100%. 0% – отсутствие соответствия, 100% – палитра полностью состоит из безопасных web-цветов.

Закончив настройку параметров оптимизации изображения, сохраните его:

- Щелкните мышью на кнопке **Save** (Сохранить), откроется диалог **Сохранить оптимизированный как** (Save Optimized As), представленный на Рис. 4.32.

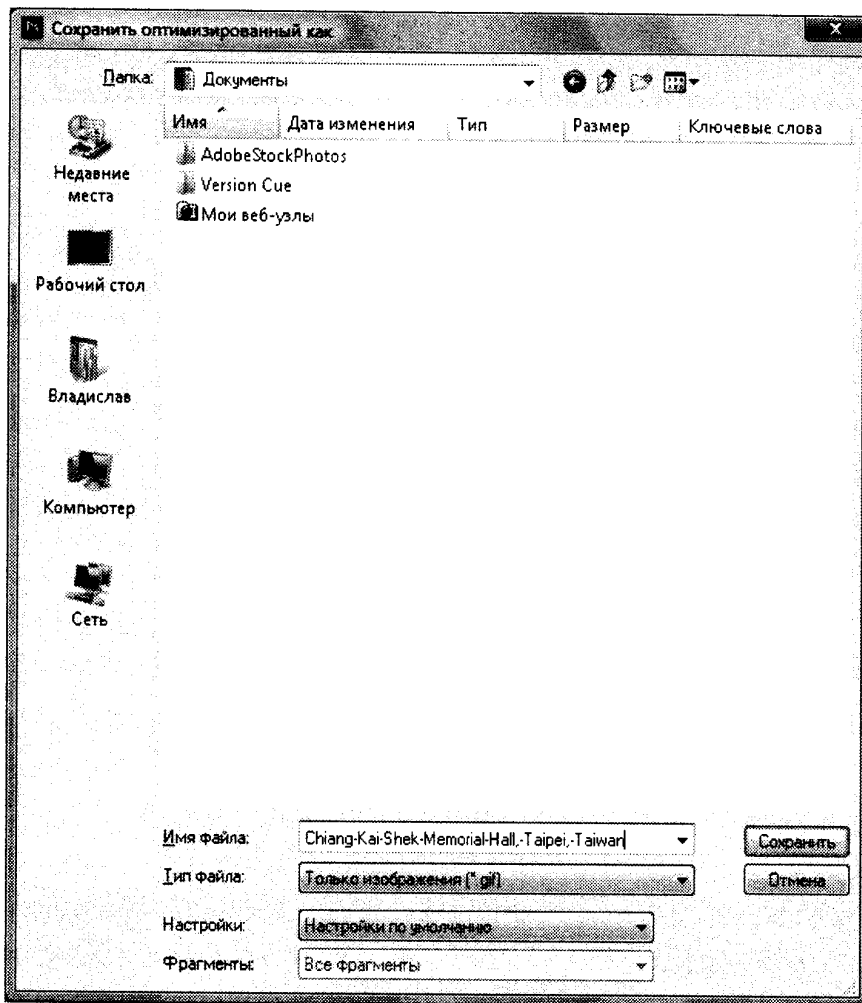


Рис. 4.32. Диалог сохранения оптимизированного файла

- Перейдите к папке, в которой вы хотите сохранить файл.
- Щелкните мышью в поле ввода **Имя файла** (Filename) и введите название, под которым вы хотите сохранить файл изображения.
- Щелкните мышью на кнопке **Сохранить** (Save), чтобы сохранить файл и закрыть диалоги **Сохранить оптимизированный как** (Save Optimized As) и **Сохранить для Web и устройств** (Save For Web and devices).

## Оптимизация кода HTML для web

Размещение web-страниц в сети Интернет предъявляет к страницам ряд требований, которые необходимо учитывать при их создании.

Существуют два основных момента, которые необходимо всегда иметь в виду.

- ✓ Первый – пропускная способность Интернет-соединений отнюдь не бесконечна, и чем меньше места будет занимать ваша страница, тем быстрее она будет загружена

пользователем и тем выше вероятность, что он дождется ее загрузки. Кроме того, за данные, прошедшие через Интернет-канал, часто приходится платить. Поэтому важнейшей характеристикой web-страницы является ее размер, желательно его минимизировать.

- ✓ И второй важный момент – компьютеры у всех пользователей разные. Люди пользуются разными мониторами, работают в разных операционных системах, просматривают ресурсы WWW с помощью разных браузеров. Все эти особенности желательно учитывать при разработке web-страниц.

## Оптимизация размеров web-страниц

Любая web-страница состоит из двух составляющих: текстового содержимого – HTML-кода и текстов страницы, и нетекстового содержимого – графики, звука, видео и т. д. Объем, занимаемый текстовым содержимым страницы, обычно невелик, да и уменьшить его сложно. Правда, существуют различные методики уменьшения объема кода страницы, но при достаточно высокой трудоемкости отдача от них невелика.

Совсем по-другому обстоит дело с нетекстовым содержимым: оно обычно достаточно велико по объему, и его оптимизация может дать хорошие результаты.

Самым популярным нетекстовым содержимым страниц является графика – она используется в виде элементов оформления, иллюстраций, коллекций изображений и т. д. Сформулируем несколько рекомендаций по использованию графики на ваших web-страницах.

- ✓ Не злоупотребляйте графикой в оформлении страниц. Все хорошо в меру. Основным содержанием вашей web-страницы, если это не рекламный сайт и не галерея изображений, является текст. Оформление страницы служит для более удобного и красивого представления текстового содержимого и ни для чего иного.
- ✓ Оптимизируйте изображения. Подготавливая изображения для размещения в web, пользуйтесь диалогом **Сохранить для Web и устройств** (Save For Web and devices) программы Adobe Photoshop. Подбирайте параметры оптимизации так, чтобы изображение занимало минимально возможный объем и при этом сохраняло достойное качество. Не бойтесь экспериментировать, попробуйте несколько вариантов настроек оптимизации и выберите лучший.
- ✓ Не размещайте изображения слишком большого размера. Помните, что чем больше размер изображения, тем больше оно занимает места, поэтому старайтесь размещать изображения минимально необходимого размера.
- ✓ Используйте уменьшенные версии изображений для предварительного просмотра. Если вы хотите расположить на странице много изображений, например вашу фотогалерею, то лучше разместите на ней уменьшенные версии изображений, щелкая по которым посетитель сможет открыть изображения большого размера. Это ускорит загрузку страницы и позволит посетителю просмотреть только те изображения, которые его заинтересуют.

По поводу размещения видео и звука на ваших страницах можно привести практически те же рекомендации, что и для размещения графики: не злоупотребляйте такими объектами; старайтесь уменьшить их размер так, чтобы при минимальном размере обеспечивалось удовлетворительное качество.

## Создание универсальных web-страниц

Как уже было сказано выше, компьютеры у всех пользователей WWW разные. Разные у пользователей размеры мониторов, разрешение экранов, размеры окна браузера. При создании web-страниц это важно учитывать.

- ✓ По статистике, разрешение экранов у большинства пользователей компьютеров 1024x768 точек и выше, но есть люди, на компьютерах которых выставлено разрешение не больше 800x600 точек. Количеством пользователей с меньшим разрешением экрана можно пренебречь. Желательно создавать web-страницы так, чтобы они помещались в web-браузерах, запущенных на компьютерах с разрешением экрана 800x600 точек. Но, вместе с тем, нельзя забывать и о пользователях с большими экранами. Разрешить это противоречие позволяет концепция резинового дизайна, рассмотренная в главе 3. Оформление страницы создается так, чтобы все элементы оформления занимали в ширину не более 750 точек, 50 точек отводится на границы окна браузера, полосы прокрутки и т. д. Если ширина окна браузера больше 750 точек, то содержимое страницы раздвигается, занимая весь свободный объем полезной информацией.
- ✓ Еще лет 10 назад большинство компьютеров не могли отображать на экране больше 256 цветов. В связи с этим была разработана специальная безопасная палитра web-цветов. Она состояла из 216 цветов, которые могли быть отображены на любом компьютере с цветным экраном. Сейчас таких компьютеров осталось очень мало, и безопасные web-цвета давно потеряли свою актуальность. Но, тем не менее, если вы хотите, чтобы с вашими web-страницами не испытывали проблем люди, пользующиеся такими компьютерами, используйте безопасную палитру, хотя это может вас сильно ограничить при создании графики.
- ✓ В настоящее время практически не осталось компьютеров, мониторы которых отображают только оттенки серого цвета. Но есть много людей с ослабленным зрением, плохо воспринимающие оттенки цвета, также множество людей страдает разными формами дальтонизма. Чтобы эти категории посетителей вашего сайта могли без труда знакомиться с содержимым ваших web-страниц, необходимо, чтобы страницы легко воспринимались, будучи преобразованными в режим оттенков серого цвета. Цвета, легко отличимые друг от друга в цветном режиме, при преобразовании в оттенки серого могут сливаться друг с другом.
- ✓ Как уже говорилось в главе 3, все изображения, размещаемые на ваших web-страницах, необходимо снабжать альтернативными описаниями. Это важно для людей, отключающих загрузку графики с web-страниц, и для людей с ослабленным зрением, пользующихся речевыми браузерами, озвучивающими содержание web-страниц.

## Заключение

Этой главой завершается рассказ о создании статических web-страниц. Вы знаете, как создавать такие страницы, знакомы с различными тонкостями их оформления, умеете делать страницы, насыщенные разнообразнейшим содержанием, так, чтобы они хорошо выглядели, быстро загружались и были доступны максимальному количеству людей.

Вы многого достигли, но вскоре достигнете еще большего. Начинается новый этап освоения web-технологий – создание динамических web-сайтов. Этому посвящены следующие главы данной книги. По сути своей, динамические сайты являются расширением структуры и возможностей сайтов статических. Уже приобретенные вами знания, умения и опыт станут основой, которую вы дополните сведениями о технологиях создания динамических сайтов.

## ГЛАВА 5.

# Web-сайт с динамическими страницами



Пришла пора оживить ваши web-страницы, добавить в них динамики и гибкости; этой главой начинается изучение технологий создания динамических web-сайтов. В ней мы разберемся, какие бывают динамические сайты, как они выполняются на сервере и в браузере пользователя. Основное внимание в этой главе будет уделено написанию сценариев на языке JavaScript и внедрению их в HTML-страницы. Вы узнаете об основных командах и конструкциях этого языка, познакомитесь с тем, как сценарии JavaScript взаимодействуют с кодом HTML-страниц. Полученные знания мы применим на практике, создав действующую программу на языке JavaScript – инженерный калькулятор.

## Как работают статические и динамические web-сайты

В первой главе мы уже знакомились с особенностями выполнения статических и динамических web-страниц на web-сервере и в браузере пользователя. Рассмотрим этот процесс подробнее.

Статические web-страницы лежат в каталогах сервера в виде уже готовых HTML-файлов. Работа сервера в этом случае заключается лишь в том, чтобы в ответ на запросы браузера передавать ему необходимые файлы. Браузер, в свою очередь, отображает страницу, опираясь на правила языка HTML, и обеспечивает реакцию на действия пользователя, щелкающего по гиперссылкам. Все просто и надежно, но возможности такого сайта ограничены. Реакция браузера на действия пользователя ограничивается переходом по ссылкам, все страницы сайта должны быть созданы заранее, и, чтобы изменить содержание сайта, необходимо переписать код страниц.

Ограничения статических web-страниц преодолеваются в динамических web-сайтах. Технологии создания таких сайтов делятся на две группы: серверные и клиентские технологии.

## Серверные технологии создания динамических web-сайтов

Серверные технологии работают следующим образом: браузер посылает web-серверу запрос, в ответ на который сервер запускает программу, называемую сценарием или скриптом. Этот сценарий, в зависимости от параметров запроса, «на лету» форми-



рует HTML-страницу, которая и передается браузеру. С помощью сценариев можно также генерировать изображения, анимации, создавать письма электронной почты и делать многое другое.

Серверные сценарии можно писать на множестве различных языков программирования, но чаще всего для этой цели используются языки, специально для этого предназначенные: PERL (Practical Extraction and Report Language – Практичный язык для создания выборок и отчетов) и PHP (расшифровывается как самоповторяющаяся аббревиатура PHP: Hypertext Preprocessor – PHP: препроцессор гипертекста).

К серверным сценариям прибегают в тех случаях, когда необходимо сохранять какую-либо информацию о посетителе на сервере, периодически обновлять содержимое сайта, не перелопачивая вручную код страниц, выдавать пользователю не всю информацию, а только соответствующую его запросу.

Серверные технологии необходимы при создании поисковых систем, форумов, Интернет-магазинов, новостных сайтов, почтовых служб типа **mail.ru** и **gmail.com** и во многих других случаях.

Одним из самых ярких применений серверных технологий является концепция «web-приложений» – программ, аналогичных по своей функциональности приложениям, запускаемым на компьютере пользователя, но работающим на web-сервере. Доступ к таким приложениям осуществляется с помощью обычного web-браузера. Плюсом такой системы является то, что пользователь может работать с программой на любом компьютере, подключенном к сети Интернет. Причем, поскольку данные пользователя тоже хранятся на сервере, нет нужды переносить их с компьютера на компьютер. Главные минусы web-приложений – необходимость быстрого Интернет-канала и меньшее удобство работы по сравнению с обычными программами. Наиболее популярным видом web-приложений являются разнообразные web-интерфейсы для работы с электронной почтой, которые есть на сайтах большинства почтовых служб.

При использовании серверных технологий никаких дополнительных требований к web-браузеру пользователя не предъявляется, он может быть практически любым, вся нагрузка по созданию страниц и обработке данных ложится на сервер, который должен обладать достаточной мощностью, чтобы успевать обрабатывать запросы всех пользователей и иметь поддержку соответствующих языков сценариев.

Поведение web-браузера при работе с сайтами, созданными при помощи серверных технологий, ничем не отличается от работы с обычными статическими сайтами. Браузер по-прежнему работает с уже готовыми HTML-страницами, разница лишь в том, что содержимое страниц формируется для каждого посетителя сайта отдельно.

## **Клиентские технологии создания динамических web-страниц**

В отличие от серверных технологий, клиентские технологии создания динамических web-страниц позволяют влиять непосредственно на работу браузера. Основу клиентских технологий составляют сценарии на языке JavaScript, Jscript или VBScript,

внедренные прямо в HTML-страницу, либо подгружаемые дополнительно. С помощью таких сценариев можно создавать различные динамические эффекты, делать web-страницы, изменяющиеся в ответ на действия пользователя, встраивать в них несложные игры и программы. Кроме того, современные web-браузеры поддерживают механизм так называемых «плагинов» – небольших дополнений к браузеру, позволяющих включать в web-страницу дополнительные динамические элементы – модули Flash, программы на языке JAVA, трехмерные объекты на языке VRML (Virtual Reality Modeling Language – Язык моделирования виртуальной реальности) и т. д.

С помощью клиентских сценариев создаются различные «всплывающие» меню; кнопки, изменяющие внешний вид при наведении на них указателя мыши, и т. д. Чтобы клиентские технологии правильно работали, необходимо, чтобы они поддерживались браузером посетителя web-страницы. При этом вся вычислительная нагрузка по их обработке ложится на его компьютер.

Наилучших результатов часто можно достигнуть, комбинируя как серверные, так и клиентские технологии на одном сайте. На сервере может генерироваться содержимое страниц, а динамическое меню сайта может реализовываться с помощью клиентских сценариев. В этой главе мы рассмотрим клиентские технологии создания динамических web-страниц с использованием языка JavaScript. Этот язык выбран нами по той причине, что он поддерживается максимальным количеством браузеров.

## Создание клиентских сценариев на языке JavaScript

Технология совместного использования HTML вместе с клиентскими сценариями называется DHTML (Dynamic HTML – динамический HTML). Обе ее составляющие – код HTML и клиентские сценарии – тесно связаны друг с другом. Сценарии, внедренные в HTML-страницу, могут влиять практически на все ее элементы, создавать новые окна браузера с HTML-кодом и изображениями, открывать диалоги с предупреждениями и вопросами.

### Объектная модель HTML-документа

Элементы HTML-документа называются объектами. Сценарии JavaScript манипулируют объектами HTML-страницы. Всю HTML-страницу можно представить как иерархию объектов. Можно представить следующий наглядный пример такой иерархии. Главный объект – квартира. Внутри квартиры три комнаты – это дочерние объекты квартиры, также их можно назвать ее потомками. А квартира для комнат является родительским объектом. Содержимое комнат (например, столы, стулья, шкафы) будет, в свою очередь, состоять из дочерних объектов для комнат. Дочерние объекты комнат являются дочерними объектами и для объекта квартиры, и наоборот, объект квартиры является родительским объектом и для объектов содержимого комнат.

Полное описание содержимого квартиры в терминах объектов будет называться объектной моделью, DOM (Document Object Model – объектная модель документа). Указание на определенный объект в комнате будет выглядеть следующим образом:

**квартира.комната[0].стол.** Имена дочерних объектов указываются после родительских, через точку. Поскольку комнат в квартире несколько, в квадратных скобках указывается номер конкретной комнаты. Такие группы объектов, состоящие из нескольких однородных элементов, называются коллекциями. Элементы коллекции могут отличаться друг от друга именами или нумерацией.

Документ HTML имеет достаточно большую иерархию объектов. Например, объект **window** отвечает за текущее окно браузера. Его дочерним объектом является объект HTML-документа в целом – **document**. У объекта **document** есть ряд дочерних коллекций объектов:

- ✓ **forms** – коллекция форм. Что такое формы, вы узнаете в следующей главе;
- ✓ **anchors** – коллекция якорей;
- ✓ **images** – коллекция изображений;
- ✓ **links** – коллекция ссылок;
- ✓ **plugins** – коллекция подключаемых модулей.

С помощью объектов можно узнать или изменить параметры большинства элементов HTML-страницы. В объектной модели JavaScript есть также несколько объектов, не касающихся напрямую документа HTML, например:

- ✓ **history** – объект, дающий доступ к истории посещенных ссылок;
- ✓ **location** – объект, содержащий текущий URL;
- ✓ **screen** – объект, дающий доступ к характеристикам экрана;
- ✓ **Date** – объект, работающий с датой и временем;
- ✓ **Array** – объект для работы с массивами переменных;
- ✓ **Math** – объект математических функций.

Обратиться к одному из элементов коллекции объектов можно, зная номер этого элемента на странице. Номера присваиваются объектам автоматически в процессе открытия страницы браузером. Нумерация идет с 0 по возрастанию. Если рассмотреть страницу, код которой показан в Листинге 5.1, то в ней тегу, отображающему рисунок **photo1.jpg**, будет присвоен номер 0, тегу с рисунком **photo2.jpg** – номер 1, а тегу с рисунком **photo3.jpg** – номер 2. И места этих тегов в иерархии объектов страницы будут **document.images[0]**, **document.images[1]** и **document.images[2]** соответственно.

#### *Листинг 5.1. Нумерация элементов web-страницы*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<title>Нумерация элементов</title>
</head>
<body>
   <br />
   <br />
   <br />
</body>
</html>
```

Для удобства обращения к элементам web-страницы им можно присваивать имена. Имя тегу HTML задается при помощи атрибута **NAME**, например, так: **<IMG NAME="rabbit" SRC="rabbit.gif">**. После этого, вместо безликого номера элемента в коллекции объектов можно использовать осмысленное имя: **document.rabbit**.

## События

Сценарии на web-странице выполняются не постоянно, а только в случае наступления определенных событий, например нажатия кнопок, наведения указателя мыши на определенный участок страницы, прохождения определенного интервала времени или другого действия. При наступлении заданного события запускается определенный сценарий JavaScript, называемый обработчиком события. После его завершения система опять переходит в режим ожидания событий, практически не потребляя ресурсов компьютера.

Хотя чаще всего события назначают различным кнопкам, событие можно «повесить» практически на любой элемент HTML-страницы. Приведем полный список возможных событий, с которыми можно связать сценарии:

- ✓ **onLoad** – окончание открытия HTML-документа;
- ✓ **onUnload** – выгрузка HTML-документа;
- ✓ **OnClick** – щелчок мышью на поверхности элемента;
- ✓ **OnDbClick** – двойной щелчок мышью на поверхности элемента;
- ✓ **OnMouseDown** – нажатие кнопки мыши;
- ✓ **OnMouseOver** – перемещение мыши на элемент из-за его пределов;
- ✓ **OnMouseMove** – перемещение мыши над элементом;
- ✓ **OnMouseOut** – перемещение мыши с элемента за его пределы;
- ✓ **OnFocus** – элемент получает фокус ввода;
- ✓ **OnBlur** – элемент теряет фокус ввода;
- ✓ **OnKeyPress** – пользователь нажимает и отпускает клавишу;

- ✓ **OnKeyDown** – пользователь нажимает клавишу над элементом;
- ✓ **OnKeyUp** – пользователь отпускает клавишу над элементом;
- ✓ **OnSubmit** – данные из формы переданы web-серверу;
- ✓ **OnReset** – форма очищена;
- ✓ **OnSelect** – пользователь выбирает текст в текстовом окне;
- ✓ **OnChange** – элемент теряет фокус ввода после изменения значения элемента.

## Базовые понятия языка JavaScript

Чтобы вы могли самостоятельно создавать сценарии, вам необходимо разобраться с основами языка JavaScript. Базовые понятия этого языка, такие как переменные, операторы, функции, являются общими для большинства языков программирования и пригодятся вам при изучении серверных языков сценариев.

### Сценарии

Любой сценарий представляет собой набор команд, последовательно исполняемых программой, выполняющей сценарий. В данном случае web-браузером. Традиционно изучение языков программирования начинается с написания программы, выводящей на экран монитора приветствие: «Hello, world». Не будем отступать от этой традиции и мы. В Листинге 5.2 приведена HTML-страница, содержащая сценарий, выводящий на экран диалог с надписью «Hello, world» (Рис. 5.1).

#### Листинг 5.2. Первый сценарий JavaScript

```
<html>
<body>
  <script language=»JavaScript»>
    window.alert(«Hello, world!»)
  </script>
</body>
</html>
```



Рис. 5.1. Диалог, вызванный сценарием JavaScript

Договоримся о терминологии. Особенности записи команды и ее параметров в сценарии называются синтаксисом. В описании синтаксиса команды необязательные ее параметры заключаются в квадратные скобки – **[необязательный\_параметр]**.

## Переменные

Результаты промежуточных вычислений, обрабатываемые текстовые строки и другая информация, с которой работают программы, хранится в переменных. Переменные можно сравнить с грифельной доской, на которой вы можете записать некоторую информацию, затем стереть ее и записать что-нибудь другое. Чтобы прочитать содержимое переменной или изменить его, нужно знать только название переменной.

Переменные могут содержать значения трех различных типов:

- ✓ Числовые. Числа могут целыми и дробными, например: 34 и 2.1234.
- ✓ Логические. Переменные могут принимать одно из двух логических значений: **True** (Истина) или **False** (Ложь).
- ✓ Строки. Переменные в JavaScript могут содержать строки из нулевого и большего количества знаков. При присвоении переменной строкового значения строка заключается в двойные или одинарные кавычки, например так: **x="строка"**. Чтобы включить кавычку в строковую переменную, необходимо поставить перед ней знак «\». Выражение **x="\ Фраза в кавычках\ "** запишет в переменную **x** значение "Фраза в кавычках".

В языке JavaScript названия переменных имеют следующие особенности:

- ✓ Название может состоять из символов латинского алфавита от «a» до «z», как строчных, так и прописных, цифр от 0 до 9 и символов подчеркивания «\_».
- ✓ Название переменной не может начинаться с цифры. Название переменной **\_3abc** – правильное, а **3\_abc** – уже нет.
- ✓ JavaScript различает строчные и прописные буквы в названии переменной: **Abc** и **ABC** – две разные переменные.

В командах JavaScript переменные могут использоваться в качестве параметров наряду с числами и строками текста.

## Операторы

Основные математические операции, манипуляции со строками и переменными производятся с помощью операторов. Основной оператор – оператор присваивания, **=**. Результат вычислений, проведенных справа от оператора **=**, записывается в переменную, имя которой стоит слева от него. Например, выражение **x=6** означает, что в переменную **x** записывается число **6**. Другой пример: **a=b** – это значит, что в переменную **a** записывается значение переменной **b**.

Математические вычисления проводятся при помощи стандартных математических операторов сложения, вычитания, деления, умножения и скобок: «**+**», «**-**», «**/**», «**\***», «**( )**», вычисления проводятся в согласии с правилами арифметики. Например так: **2\*4+1**.

Результатом выполнения этого выражения будет число 9. А в результате выполнения выражения **m=(3+1)\*2**, переменная **m** примет значение 8.

Вычисления могут производиться и над значениями переменных. Например, если значение переменной **exp** равняется 25, то в результате выполнения выражения **res=exp/5** в переменную **res** будет записано число 5.

Оператор сложения используется также для объединения между собой текстовых строк. Результатом выражения «**амбар**» + «**цумян**» будет строка «амбарцумян».

Кроме стандартных математических действий, в языке JavaScript есть несколько дополнительных математических операций.

- ✓ **%** – модуль. Вычисляет остаток от целочисленного деления одного выражения на другое. Пример – **12%5**, результатом этого выражения будет 2.
- ✓ **+=** – добавление некоторого значения к переменной. Выражение **x+=y** добавляет к значению переменной **x** значение переменной **y** и записывает результат в переменную **x**. Это выражение является более короткой записью выражения **x=x+y**.
- ✓ **-=** – вычитание некоторого значения из переменной. Выражение **-=y** вычитает из значения переменной **x** значение переменной **y** и записывает результат в переменную **x**. Это выражение является более короткой записью выражения **x=x-y**.
- ✓ **\*=** – умножение значения переменной на некоторую величину. Выражение **x\*=y** умножает значение переменной **x** на значение переменной **y** и записывает результат в переменную **x**. Это выражение является более короткой записью выражения **x=x\*y**.
- ✓ **/=** – деление значения переменной на некоторую величину. Выражение **x/=y** делит значение переменной **x** на значение переменной **y** и записывает результат в переменную **x**. Это выражение является более короткой записью выражения **x=x/y**.
- ✓ **++** – инкремент. Увеличивает значение переменной на единицу. Записывается так: **x++**. В результате значение переменной **x** увеличится на 1.
- ✓ **--** – декремент. Уменьшает значение переменной на единицу. Записывается так: **x--**. В результате значение переменной **x** уменьшится на 1.

## Логические операторы

Результатом логических, или Булевых, выражений, может являться одно из двух значений: **True** (истина) или **False** (Ложь). Простые логические выражения состоят из двух значений, или выражений, объединенных оператором сравнения: **1e\_значение оператор\_сравнения 2e\_значение**. Существуют следующие операторы сравнения:

- ✓ **==** – равенство. Если оба значения равны, то результат выражения равен **True** (истина), иначе **False** (Ложь).
- ✓ **!=** – неравенство. Если значения не равны, то результатом выражения будет **True** (истина).
- ✓ **<** – меньше. Результат **True** (истина), если первое выражение меньше второго.

- ✓ **>** – больше. Результат **True** (истина), если 1-е выражение больше второго.
- ✓ **<=** – меньше либо равно. Результат **True** (истина), если 1-е выражение меньше либо равно второму.
- ✓ **>=** – больше либо равно. Результат **True** (истина), если 1-е выражение больше либо равно второму.

Приведем несколько примеров логических выражений. **2<3** – результат **True** (истина), поскольку 2 меньше 3; **2\*2==4** – **True** (истина); **34>=45** – **False** (Ложь). Несколько простых логических выражений можно объединить в более сложное выражение при помощи логических операторов. Логическое выражение будет выглядеть так: **1е\_логическое\_значение логический\_оператор 2е\_логическое\_значение**. Исключение составляет логический оператор **Not** (Не), который размещается перед единственным логическим значением.

Логические операторы бывают следующими:

- ✓ **!** – **Not** (Не). Меняет результат логического выражения на обратный, т. е. значение **True** (истина) меняет на **False** (Ложь), и наоборот.
- ✓ **&&** – **And** (И). Результатом выражения будет **True** (истина) в том случае, если значение **True** (истина) имеют оба логические значения.
- ✓ **||** – **OR** (Или). Результатом выражения будет **True** (истина), если значение **True** (истина) имеет хотя бы одно из логических значений.

Результатом выражения **2>5||5>2** будет **True** (истина), поскольку одно из выражений имеет результат **True** (истина). Результатом выражения **3>0 && 3>5** будет **False** (Ложь), поскольку одно из выражений имеет результат **False** (Ложь).

## Условные переходы

При написании программ часто возникают ситуации, в которых необходимо, чтобы при одних значениях определенной переменной выполнялись одни команды, а при других значениях – другие. В таких случаях используются операторы условного перехода.

Можно привести следующий пример условного оператора из реальной жизни: «Если пойдет дождь, то я останусь дома, в противном случае я пойду гулять». В рассмотренном условном операторе условие выполнения команд определяет переменная «погода», если ее значение будет «дождь», то выполняются команды «остаться дома», при любых других значениях этой переменной выполняется группа команд «пойти гулять».

В языке JavaScript условные переходы создаются при помощи конструкции **IF ELSE**, см. Листинг 5.3. Если результат логического выражения **условие** равен **True** (истина), то выполняются команды **последовательность\_команд\_1**, в противном случае выполняются команды **последовательность\_команд\_2**. То есть, если условие выполняется, то исполняются команды, находящиеся в фигурных скобках непосредственно после оператора **IF**, если же условие не выполнено, – то команды, идущие за оператором **ELSE**. Если вместо последовательности команд нужно исполнить всего одну команду, то фигурные скобки использовать необязательно.



*Листинг 5.3. Условный переход*


---

```

if (условие)
    {последовательность_команд_1}
else
    {последовательность_команд_2}

```

---

Пример реального условного перехода в сценарии JavaScript приведен в Листинге 5.4. Если значение переменной **x** равняется 3, то на экран выводится диалоговое окно, сообщающее об этом. Если же значение этой переменной иное, то выводится другое диалоговое окно, сообщающее о том, что **x** не равен 3.

*Листинг 5.4. Пример реального условного перехода*


---

```

If (x==3)
    {window.alert("x=3")}
else
    {window.alert("x<>3")}

```

---

**Циклы**

При написании программ часто возникает необходимость повторять одно действие несколько раз подряд, возможно с небольшими изменениями. В таких случаях используются циклы. Цикл состоит из двух основных частей: тела и заголовка. В заголовке цикла задается условие, при выполнении которого исполнение цикла закончится. Тело цикла – команды, заключенные в фигурные скобки и выполняемые по кругу, пока условие выхода из цикла не будет выполнено. Если тело цикла состоит всего из одной команды, фигурные скобки использовать необязательно. Каждое выполнение цикла называется итерацией.

**Циклы FOR**

В языке JavaScript предусмотрено два вида циклов. Первый из них цикл **FOR**, его структура показана в Листинге 5.5. Цикл **FOR** является «циклом со счетчиком». Так называются циклы, в которых есть переменная-счетчик, значение которой изменяется при каждой итерации цикла. При достижении счетчиком определенного значения выполнение цикла заканчивается. Выражение **нач\_знач** задает начальное значение переменной-счетчика. Выражение **условие** – логическое выражение, становящееся ложным при определенном значении переменной-счетчика. Когда условие перестает выполняться, цикл завершается. Выражение **приращение** изменяет переменную-счетчик при каждой итерации цикла.

*Листинг 5.5. Структура цикла FOR*


---

```

for (нач_знач; условие; приращение)
{

```

---

тело\_цикла

}

Код реального документа HTML с циклом **FOR** приведен в Листинг 5.6. На Рис. 5.2 показано, что отобразит web-браузер при открытии данной страницы.

### Листинг 5.6. Пример цикла FOR

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Цикл FOR</title>
</head>
<body>
  <script language="JavaScript">
    for (i=1;i<=10;i++){
      document.write(«Это строка номер « + i + «<br>»)
    }
  </script>
</body>
</html>
```

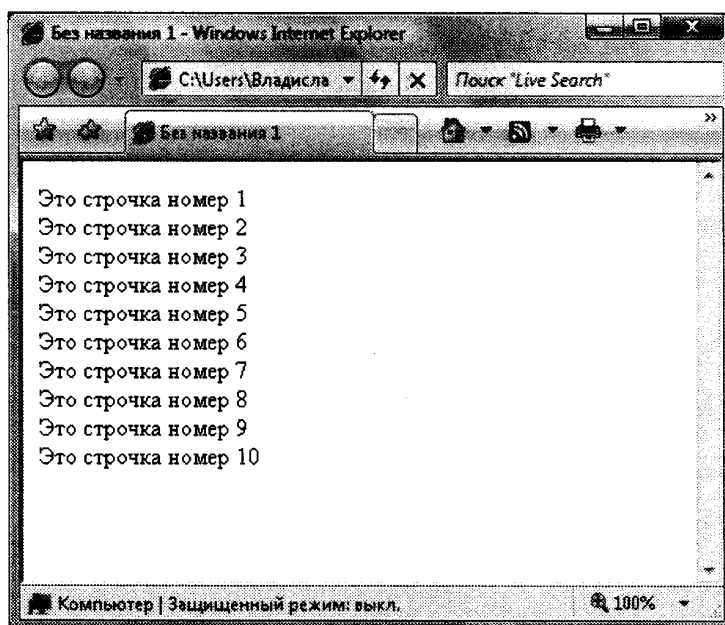


Рис. 5.2. Отображение работы цикла **FOR** на странице web-браузера

## Циклы WHILE

Другой вид циклов в языке JavaScript – циклы **WHILE**. В отличие от цикла **FOR**, в заголовке этого вида циклов осуществляется только проверка истинности условия. Структура цикла **WHILE** приведена в Листинге 5.7. Пока значение логического выражения **условие** равняется **True** (истина), выполнение цикла будет продолжаться. Если в качестве условия поставить выражение, истинное при любых обстоятельствах, например **2=2**, или просто **True**, то цикл будет выполняться бесконечно, или пока не будет принудительно прерван.

### *Листинг 5.7. Структура цикла FOR*

---

```
while (условие)
{
    тело_цикла
}
```

---

В Листинге 5.8 приведен пример цикла **WHILE**, выполняющего работу, аналогичную циклу **FOR** из Листинге 5.6. Обратите внимание, что задание начального значения переменной цикла и ее приращение осуществляется вне заголовка цикла.

### *Листинг 5.8. Пример цикла WHILE*

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Цикл WHILE</title>
</head>
<body>
    <script language="JavaScript">
        i=1;
        while(i<=10){
            document.write(«Это строка номер « + i + «<br>»)
            i++;
        }
    </script>
</body>
</html>
```

---

## Команды, управляющие выполнением циклов

Для управления работой циклов используются команды **break** и **continue**. Команда **break** прерывает выполнение цикла и передает управление первой команде после цикла. Добавим к циклу **WHILE** из Листинга 5.8 условный переход, выполняющий команду **break**, при достижении переменной **i** значения 5. Результат приведен в Листинге 5.9. Как изменения отразились на получаемой web-странице, видно на Рис. 5.3. При достижении переменной **i** значения 5 выполнение цикла принудительно прерывается.

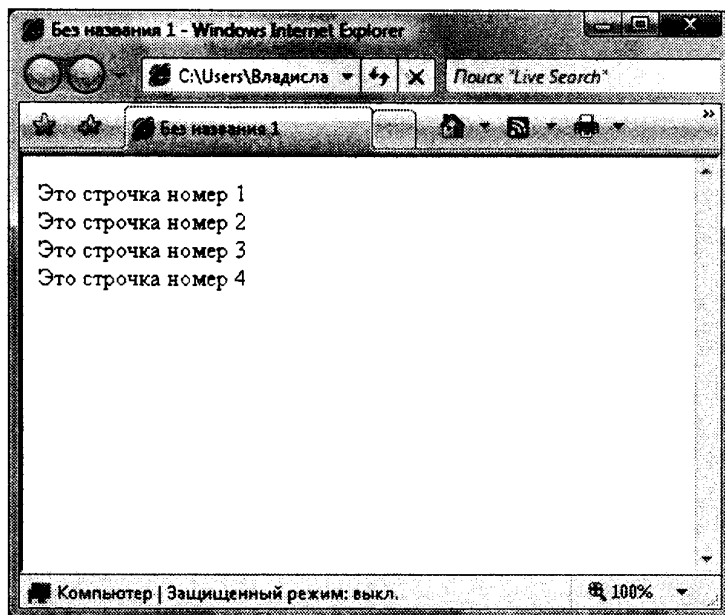


Рис. 5.3. Применение команды **break**

### Листинг 5.9. Пример использования команды **break**

```
<html>
<head>
  <title>Цикл WHILE с командой break</title>
</head>
<body>
  <script language=»JavaScript»>
    i=1;
    while(i<=10){
      if (i==5)
        break
      document.write(«Это строка номер « + i + «<br>»)
      i++;
    }
  </script>

```

```

    }
  </script>
</body>
</html>

```

Команда **continue** вызывает выполнение следующей итерации цикла, не заканчивая текущую. Добавим в код страницы в Листинге 5.6 условный переход, выполняющий команду **continue**, когда значение **i** достигнет значения 5. Полученный код показан в Листинге 5.10. В результате открытия такой страницы браузер отобразит страницу, показанную на Рис. 5.4. На странице будет пропущена строка 5, поскольку команда **continue** запустила следующую итерацию цикла, не заканчивая текущую.

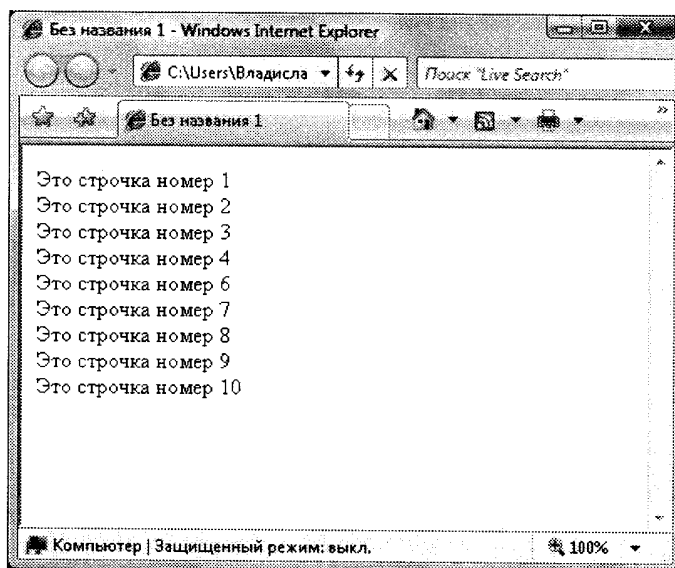


Рис. 5.4. Пример работы команды **continue**

Листинг 5.10. Пример использования команды **continue**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Цикл FOR</title>
</head>
<body>
  <script language="JavaScript">
    for (i=1;i<=10;i++){
      if (i == 5)
        continue

```

```

document.write("Это строка номер " + i + "<br>")
}
</script>
</body>
</html>

```

## Функции

Последовательность из нескольких команд сценария можно объединить между собой, создав функцию. После этого команды функции можно выполнить, вызвав функцию по названию. В функции можно свести неоднократно используемые блоки кода, чтобы вместо повторения кода просто вызывать функцию. Кроме того, в функции можно вынести последовательности команд, дающие некий законченный результат, – это позволяет улучшить структуру программы.

Функции создаются по образцу, показанному в Листинге 5.11. **Имя\_функции** – название, по которому функция будет вызываться в дальнейшем. **Параметры\_функции** – переменные, значение которых передается в функцию. Внутри функции команды могут оперировать этими переменными.

### *Листинг 5.11. Структура функции*

```

function имя_функции (параметры_функции)
{
    код_функции
}

```

В качестве результата своей работы функция может возвращать некоторое значение, которое можно присваивать переменным, использовать в вычислениях и т. д. Вызов функции происходит следующим образом: **имя\_функции (параметры\_функции)**. Можно присвоить значение функции переменной: **переменная = имя\_функции (параметры\_функции)**. Значение функции присваивается с помощью оператора **return**. Используется он так: **return значение\_функции**.

В Листинге 5.12 показано практическое использование функции на примере функции, возводящей число **a** в целую положительную степень **b**. Код JavaScript страницы последовательно возводит двойку в степени от 0 до 10, а результаты работы можно увидеть на Рис. 5.5.

### *Листинг 5.12. Пример функции*

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

```

```
<head>
  <title>Использование функций</title>
  <script language=»JavaScript»>
    function powerb (a, b){
      var i, atemp;
      atemp=a
      if (b>1){
        for (i=2;i<=b;i++)
          atemp*=a
      }
      if(b==0)
        atemp=1
      return atemp
    }
  </script>
</head>
<body>
  <script language=»JavaScript»>
    i=0;
    while(i<=10){
      document.write(«2 в степени « + i + « равняется « + powerb(2,
i) + «<br>»)
      i++;
    }
  </script>
</body>
</html>
```

---

Обратите внимание на команду **var**. Эта команда создает переменные и задает их область действия. Если переменная создана командой **var** внутри функции, то и доступ к ней можно будет получить только внутри этой функции. Если ее создать в основной части сценария, то переменная будет доступна из любой части этого сценария, в том числе и из всех его функций. Явное задание переменных не является обязательным в языке JavaScript и используется в основном для того, чтобы разграничить зону действия переменных. Например, если в основной части сценария есть

переменная **i**, а внутри функции используется переменная с таким же именем, то, если не задать в функции эту переменную явно, в функции будет использоваться переменная из основного сценария. В качестве эксперимента вы можете удалить команду **var** из документа в Листинге 5.12 и посмотреть, что получится в результате. При создании переменных командой **var**, можно задать их начальное значение, например: **var i=0, m="переменная"**.

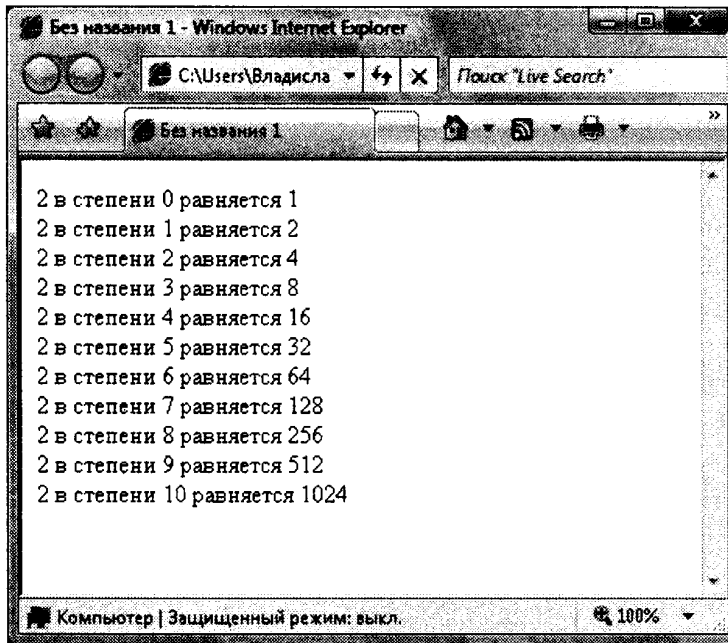


Рис. 5.5. Пример работы функции JavaScript

Имена функций в JavaScript должны соответствовать тем же требованиям, что и имена переменных:

- ✓ Название может состоять из символов латинского алфавита от «a» до «z», как строчных, так и прописных, цифр от 0 до 9 и символов подчеркивания.
- ✓ Название функции не может начинаться с цифры. Название функции **\_3abc** – правильное, а **3\_abc** – уже нет.
- ✓ JavaScript различает строчные и прописные буквы в названии функции: **Abc** и **ABC** – две разные функции.

## Объекты

Пришло время поближе познакомиться с объектами. Любой объект представляет собой контейнер, в который «сложено» некоторое количество переменных и функций, работающих с этими переменными. Проводя аналогию с обычным миром, представим объектом книжный шкаф. Переменными этого объекта будут сами книги, их количество и расположение на полке. Функциями – доставание книг с полки для прочтения, составление каталога, возврат книг обратно на полку. Переменные объекта называются его свойствами, функции – методами.

При работе сценария может быть создано несколько копий объектов одного типа, и каждая из копий называется «экземпляром объекта». После создания экземпляра



объекта остальной код программы может обращаться к его методам и переменным. Чтобы создать новый экземпляр объекта, обычно используется команда **new**. Синтаксис команды **new**: **имя\_объекта = new тип\_объекта ([параметры])**. При обращении к свойству или методу объекта их имена пишутся через точку после названия объекта: **имя\_объекта.свойство**. Рассмотрим подробнее некоторые объекты документа HTML и языка JavaScript.

## Объект Array

Позволяет создавать и работать с массивами. Массив – это группа переменных, объединенных под одним именем. Эти переменные называются элементами массива. Элементы массива пронумерованы от нуля и до некоего **N**, определяющего номер максимального элемента в массиве. Доступ к элементам массива осуществляется через указание названия массива и номера элемента в квадратных скобках: **название\_массива[номер\_элемента]**.

Новые массивы создаются при помощи команды **new**: **название\_массива = new Array ([длина\_массива])**. С помощью необязательного параметра **длина\_массива** можно задать количество элементов в массиве. Нумероваться эти элементы будут от нуля до числа, на единицу меньшего значения выражения **длина\_массива**. В Листинге 5.13 приведен пример использования массивов. В этом примере создается массив из пяти элементов, и трем первым его элементам присваиваются значения.

### *Листинг 5.13. Пример использования массива*

```
myArray = new Array (5)
myArray[0] = "мешок с цементом"
myArray[1] = "мешок с песком"
myArray[2] = "ведро с обойным клеем"
```

Если при создании массива не задавать количество его элементов, то будет создан массив нулевой длины. Но при задании значения некоторому элементу этого массива массив расширяется до этого номера. В программе из Листинга 5.14 создается массив нулевой длины, а затем создается элемент этого массива под номером 99. Массив при этом автоматически расширяется до 100 элементов, от 0 до 99. С помощью свойства **length** объекта **Array** можно узнать количество элементов в массиве.

### *Листинг 5.14. Пример использования массива*

```
myArray = new Array ()
myArray[99] = "сапог"
```

## Объект Math

В этом объекте собраны различные математические функции и константы. Константы сохранены в свойствах объекта, доступ к математическим функциям осуществляется при помощи его методов.

Свойства объекта **Math**:

- ✓ **E**. Значение числа «е».
- ✓ **LN2**. Значение натурального логарифма числа 2.
- ✓ **LN10**. Значение натурального логарифма числа 10.
- ✓ **LOG2E**. Значение логарифма числа «е» по основанию 2.
- ✓ **LOG10E**. Значение десятичного логарифма числа «е».
- ✓ **PI**. Значение числа **пи**.
- ✓ **SQRT1\_2**. Значение квадратного корня из 0.5.
- ✓ **SQRT2**. Значение квадратного корня из 2.

Методы объекта **Math**:

- ✓ **abs**. Возвращает модуль аргумента. Синтаксис: **Math.abs(аргумент)**, **аргумент** – произвольное число.
- ✓ **acos**. Возвращает арккосинус аргумента. Синтаксис: **Math.acos(аргумент)**, **аргумент** – число от -1 до 1, если число выходит за эти границы, метод возвращает результат 0.
- ✓ **asin**. Возвращает арксинус аргумента. Синтаксис: **Math.asin(аргумент)**, **аргумент** – число от -1 до 1, если число выходит за эти границы, метод возвращает результат 0.
- ✓ **atan**. Возвращает арктангенс аргумента. Синтаксис: **Math.atan(аргумент)**, **аргумент** – число от -1 до 1, если число выходит за эти границы, метод возвращает результат 0.
- ✓ **ceil**. Возвращает значение аргумента, округленное в большую сторону. Синтаксис: **Math.ceil(аргумент)**, **аргумент** – произвольное число.
- ✓ **cos**. Возвращает косинус аргумента. Синтаксис: **Math.cos(аргумент)**, **аргумент** – произвольное число.
- ✓ **exp**. Возвращает экспоненту от аргумента. Синтаксис: **Math.exp(аргумент)**, **аргумент** – произвольное число.
- ✓ **floor**. Возвращает значение аргумента, округленное в меньшую сторону. Синтаксис: **Math.floor(аргумент)**, **аргумент** – произвольное число.
- ✓ **log**. Возвращает натуральный логарифм аргумента. Синтаксис: **Math.log(аргумент)**, **аргумент** – любое положительное число.
- ✓ **max**. Возвращает значение большего из двух аргументов. Синтаксис: **Math.max(аргумент1, аргумент2)**, **аргумент1**, **аргумент2** – произвольные числа.
- ✓ **min**. Возвращает значение меньшего из двух аргументов. Синтаксис: **Math.min(аргумент1, аргумент2)**, **аргумент1**, **аргумент2** – произвольные числа.

- ✓ **pow**. Возводит число в произвольную степень. Синтаксис: **Math.pow(основание, степень)**. Число **основание** возводится в степень **степень**, **основание** и **степень** – произвольные числа.
- ✓ **random**. Возвращает случайное число в диапазоне от 0 до 1. Синтаксис: **Math.random(аргумент)**, **аргумент** – произвольное число.
- ✓ **round**. Возвращает значение аргумента, округленное по правилам округления. Синтаксис: **Math.round(аргумент)**, **аргумент** – произвольное число.
- ✓ **sin**. Возвращает синус аргумента. Синтаксис: **Math.sin(аргумент)**, **аргумент** – произвольное число.
- ✓ **sqrt**. Возвращает квадратный корень аргумента. Синтаксис: **Math.sqrt(аргумент)**, **аргумент** – неотрицательное число. В том случае, когда **аргумент** меньше нуля, метод возвращает значение 0.
- ✓ **tan**. Возвращает тангенс аргумента. Синтаксис: **Math.tan(аргумент)**, **аргумент** – произвольное число.

## Объект window

Этот объект описывает открытое окно браузера и создается браузером при открытии нового документа HTML. Все объекты HTML-документа являются дочерними объектами объекта **window**. Для создания нового окна используется метод **open** следующим образом: **название\_окна=window.open (“URL”, “имя\_окна”[, “настройки\_окна”])**.

- ✓ **название\_окна** – название нового экземпляра окна.
- ✓ **URL** – адрес HTML-документа, открываемого в окне.
- ✓ **имя\_окна** – название окна, используя которое можно обращаться к этому окну из тегов HTML.
- ✓ **настройки\_окна** – дополнительные параметры, задаваемые для создаваемого окна. Параметры задаются в кавычках по парам: **название\_параметра=значение\_параметра**. Параметры перечисляются через запятую, например, так: **“width=640, height=480,resize=no”**. Если при создании окна дополнительные параметры не задавать, то будут установлены параметры по умолчанию.

При создании нового окна ему можно задать такие параметры:

- ✓ **toolbar**. Панель инструментов. Может принимать значения **yes** и **no**. При значении **yes** у создаваемого окна будет панель инструментов.
- ✓ **location**. Адресная строка. Может принимать значения **yes** и **no**. При значении **yes** у создаваемого окна будет адресная строка.
- ✓ **status**. Строка статуса. Может принимать значения **yes** и **no**. При значении **yes** у создаваемого окна будет статусная строка.
- ✓ **menubars**. Строка меню. Может принимать значения **yes** и **no**. При значении **yes** у создаваемого окна будет строка меню.

- ✓ **scrollbars**. Полосы прокрутки. Может принимать значения **yes** и **no**. При значении **yes** у создаваемого окна будут полосы прокрутки в том случае, когда содержимое не будет помещаться в окне.
- ✓ **resizable**. Возможность изменения размеров окна. Может принимать значения **yes** и **no**. При значении **yes** размеры созданного окна можно будет изменять.
- ✓ **width**. Ширина окна. Ширина задается в пикселях.
- ✓ **height**. Высота окна. Высота тоже задается в пикселях.

Команда **window.open("index.html", "win1", "width=640, height=480, resize=no, scrollbars=no")** создаст окно браузера размером 640x480 пикселей, без полос прокрутки и возможности изменения размера, содержащее HTML-документ **index.html**.

При обращении к методам, свойствам и дочерним объектам текущего окна вместо названия экземпляра объекта **window** можно писать **window**, либо вообще ничего не писать. Записи **window.alert (сообщение)** и просто **alert (сообщение)** вызывают один и тот же метод.

Методы объекта **window**:

- ✓ **alert**. Создает диалог с заданным текстом и кнопкой **OK** (Рис. 5.6). Синтаксис метода: **window.alert(сообщение), сообщение** – текст, показываемый в диалоге.

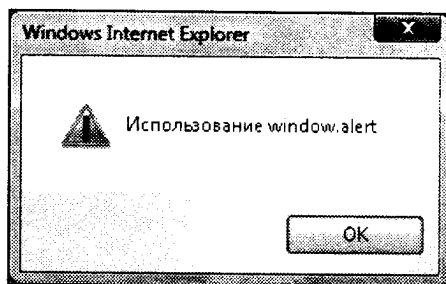


Рис. 5.6. Использование метода **window.alert**

- ✓ **close**. Закрывает окно. Синтаксис: **название\_окна.close ()**, название\_окна – название экземпляра объекта **window**. Командой **window.close ()** или просто **close ()** закрывается текущее окно.
- ✓ **confirm**. Создает диалог с заданным текстом и кнопками **OK** и **Отмена (Cancel)**, представленный на Рис. 5.7. Если пользователь щелкнет на кнопке **OK** этого диалога, то метод вернет значение **True (Истина)**, если пользователь щелкнет на кнопке **Отмена (Cancel)**, то метод вернет значение **False (Ложь)**. Синтаксис метода: **window.confirm (сообщение), сообщение** – текст, показываемый в диалоге.

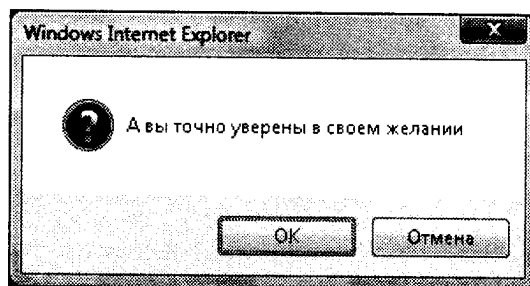


Рис. 5.7. Использование метода **confirm**

- ✓ **open**. Открывает новое окно и создает новый экземпляр объекта **window**. Подробное описание этого метода уже давалось чуть выше в этой главе.
- ✓ **prompt**. Создает диалог с заданным текстом и полем ввода (Рис. 5.8). Синтаксис метода: **window.prompt (сообщение [,значение\_по\_умолчанию])**, **сообщение** – текст, показываемый в диалоге, **значение\_по\_умолчанию** – величина, являющаяся значением по умолчанию для поля ввода. Метод возвращает значение, введенное пользователем в поле ввода.



Рис. 5.8. Использование метода **prompt**

Свойства объекта **window**:

- ✓ **defaultStatus**. Содержание строки статуса окна браузера по умолчанию, когда она не занята другими сообщениями, создаваемыми самим браузером или задаваемыми свойством **status**.
- ✓ **status**. Содержание статусной строки браузера. Является более приоритетным по сравнению со свойством **defaultStatus**.

## Объект **document**

Объект, описывающий HTML-документ в целом. Все элементы HTML-страницы являются его дочерними объектами.

Основные методы объекта **document**:

- ✓ **write**. Записывает произвольный текст в документ HTML. Этот метод позволяет генерировать с помощью JavaScript HTML-документы целиком, записывая в них все теги и текст, включая сценарии JavaScript. Синтаксис: **document.write (текст)**, **текст** – текст, записываемый в документ HTML.
- ✓ **writeln**. Аналогичен методу **write**, с той лишь разницей, что после текста в документ HTML записывается символ перевода строки. Синтаксис: **document.writeln (текст)**, **текст** – текст, записываемый в документ HTML.

Основные свойства объекта **document**:

- ✓ **AlinkColor**. Цвет активных ссылок на странице.
- ✓ **LinkColor**. Цвет неактивных ссылок на странице.
- ✓ **VlinkColor**. Цвет посещенных ссылок на странице.
- ✓ **form**. Все формы документа.
- ✓ **BgColor**. Цвет фона страницы.

- ✓ **FgColor**. Цвет текста страницы.
- ✓ **LastModified**. Время последней модификации документа.
- ✓ **title**. Содержание заголовка страницы.
- ✓ **URL**. Полный URL документа.

Свойства **alinkColor**, **linkColor**, **vlinkColor**, **BgColor** и **FgColor** задают цвета документа. Цвета записываются в виде строковой переменной, содержащей в текстовом виде три шестнадцатеричных числа, задающих интенсивность красной, зеленой и голубой составляющих цвета. Например, так: «FFFFFF» – это белый цвет, или так: «000000» – это черный цвет. Некоторые основные цвета можно задать, записав в свойство название этого цвета, например, черному цвету соответствует значение «black». Приведем список названий некоторых основных цветов:

- ✓ **black**. Черный.
- ✓ **blue**. Голубой.
- ✓ **brown**. Коричневый.
- ✓ **cyan**. Синий.
- ✓ **gray**. Серый.
- ✓ **green**. Зеленый.
- ✓ **magenta**. Малиновый.
- ✓ **orange**. Оранжевый.
- ✓ **pink**. Розовый.
- ✓ **red**. Красный.
- ✓ **turquoise**. Бирюзовый.
- ✓ **white**. Белый.
- ✓ **yellow**. Желтый.

### Коллекция **images**

Коллекции представляют собой массивы, элементами которых являются объекты. Элементами коллекции **images** являются экземпляры объекта **image**. Объект **image** является объектом изображения в документе HTML. Коллекция **images** является дочерним объектом объекта **document** и содержит объекты всех изображений HTML-страницы. Свойство **length** коллекции **images** позволяет узнать общее количество изображений на странице. Обращение к объектам конкретных изображений происходит как обращение к элементам массива **document.images[номер\_изображения]**.

Объект **image** позволяет управлять практически всеми параметрами изображения на странице. У объекта **image** есть следующие свойства:

- ✓ **border**. Ширина границы изображения в пикселях.
- ✓ **complete**. Логическая переменная, показывающая, загрузилось ли изображение. Когда изображение полностью загружено, принимает значение **True** (Истина).
- ✓ **height**. Высота изображения в пикселях.
- ✓ **hspace**. Расстояние между краем изображения и текстом, его окружающим, по горизонтали.
- ✓ **src**. Адрес, по которому располагается графический файл.
- ✓ **vspace**. Расстояние между краем изображения и текстом, его окружающим, по вертикали.
- ✓ **width**. Ширина изображения в пикселях.

Изменяя параметры изображений на странице, можно создавать различные динамические эффекты. Следующий код, например, заменяет одну картинку на странице на другую: `document.images[0].src="image.jpg"`.

## Взаимодействие JavaScript и HTML

Код JavaScript-сценариев, при использовании их на HTML-страницах, обычно делится на две части: определение функций выносится в заголовок страницы, а вызываются они из тела страницы при наступлении различных событий. При размещении в коде страницы сценарии заключаются в контейнер `<script></script>`. С помощью атрибута **language** можно явно указать, на каком языке написан сценарий, заключенный в контейнер: `<script language="JavaScript">`.

В целях дополнительной совместимости со старыми браузерами, которые не знают, что такое сценарии, перед кодом сценария часто вставляют: `<!--`, а после него: `-->`. Эта комбинация символов обозначает комментарий HTML – часть кода страницы, не предназначенная для рассмотрения браузером. Браузеры, не знающие о сценариях, благополучно пропускают этот участок, а браузеры, знакомые со сценариями, знают об этом трюке и обрабатывают содержимое таких комментариев. В Листинге 5.15 показано, как происходит внедрение кода JavaScript на HTML-страницу.

### *Листинг 5.15. Внедрение JavaScript на HTML-страницу*

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
<http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>В эту страницу внедрен код JavaScript </title>
  <script language="JavaScript">
  <!--
```

```
function myFunc(parameters) {  
    ...  
}  
-->  
</script>  
</head>  
<body>  
    <script language=»JavaScript»>  
    <!--  
    myFunc(parameters)  
    -->  
</body>  
</html>
```

## Вынесение функций в отдельный файл

Если одни и те же функции используются на нескольких страницах сайта, то целесообразно вынести определения функций в отдельный файл. Файл сценариев JavaScript обычно носит расширение **js**. В файл сценариев просто записываются определения функций, одной за другой. Чтобы использовать на странице HTML функции, определенные в файле сценариев, этот файл подключается к HTML-странице с помощью атрибута **src** тега **<script>**: **<script language=»JavaScript» src=»main.js»></script>**. В приведенном теге к HTML-странице были подключены функции из файла **main.js**.

## События и сценарии JavaScript

Как уже говорилось выше, сценарии JavaScript на HTML-страницах обычно связываются с наступлением различных событий. Реакцию на события можно добавлять практически ко всем тегам HTML-документа с помощью соответствующих атрибутов тега. В значении атрибута могут использоваться практически любые выражения JavaScript. Если необходимо на одно событие «повесить» несколько выражений, то они записываются через точку с запятой.

### Событие **onClick**

Самым часто используемым событием является событие **onClick** – щелчок мышью. В следующем примере мы добавим к тегу изображения **<img>** реакцию на щелчок мышью: **<img src=»image.jpg» onClick= «openWin(»балалайка»)»>**. После добавления атрибута **onClick** щелчок мышью на изображении будет приводить к выполнению функции **openWin** с аргументом «балалайка».



По правилам языка HTML значение атрибута **onClick** заключено в кавычки. Строковое значение, передаваемое функции, тоже должно быть заключено в кавычки. Поэтому, чтобы не возникло путаницы и программа работала правильно, аргумент функции заключен в одинарные кавычки. Этому правила следует придерживаться и в дальнейшем: если строка вызова функции заключена в кавычки, то с аргументами функции следует использовать кавычки другого типа.

## Открытие нового окна браузера по щелчку мыши

Одним из самых частых случаев использования сценариев JavaScript является открытие новых окон браузера с заданными параметрами при щелчке на кнопке или на ссылке. В Листинге 5.16 показано, как это можно делать. При щелчке мышью по ссылке «**щелкни меня**» будет выполняться функция **openwin**, открывающая новое окно браузера, лишенное практически всех элементов управления. Аргументами этой функции является название открываемого документа и размеры открываемого окна. Функция возвращает значение **False** (Ложь), которое передается тегу **<a>**. Получив от события **onClick** результат **False** (Ложь), тег **<a>** в ответ на щелчок мышью не открывает прописанную в нем ссылку.

Это сделано для совместимости с браузерами, не умеющими выполнять сценарии JavaScript, и браузерами, в которых выполнение сценариев отключено. Работает эта система таким образом: если браузер понимает сценарии, то срабатывает событие **onClick** и открывается окно заданного размера с заданными свойствами. При этом работа тега **<a>** блокируется. Если же браузер не выполняет сценарии, то он игнорирует событие **onClick** и открывает окно с документом, прописанным в атрибуте **href** тега. Правда, окно это будет произвольного размера и с настройками по умолчанию, но зато пользователь в любом случае получит доступ к нужной ему информации.

*Листинг 5.16. Использование JavaScript для открытия новых окон браузера*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
«http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd»
<html xmlns=»http://www.w3.org/1999/xhtml»>
<head>
  <script language=»JavaScript» >
  <!--
  function openwin(w,par) {
    window.open (w,»dpw»,par+»status=no, toolbar=no, menubar=no,
resizable = no, scrollbars=no, titlebar=no, location=no»);
    return false;
  }
  -->
</script>
```

```

<title>Открытие окон с помощью JavaScript</title>
</head>
<body>
  <a href=»newwin.html» target=»_blank» onClick=»return
  openwin('newwin.html', 'width=670,height=360') »>Щелкни меня</a>
</body>
</html>

```

## Общий список возможных событий

Перечислим еще раз все возможные события, в ответ на которые можно запускать сценарии JavaScript:

- ✓ **onLoad** – окончание открытия HTML-документа.
- ✓ **onUnload** – выгрузка HTML-документа.
- ✓ **OnClick** – щелчок мышью по элементу.
- ✓ **OnDbClick** – двойной щелчок мышью по элементу.
- ✓ **OnMouseDown** – нажатие кнопки мыши.
- ✓ **OnMouseOver** – перемещение мыши на элемент из-за его пределов.
- ✓ **OnMouseMove** – перемещение мыши над элементом.
- ✓ **OnMouseOut** – перемещение мыши с элемента за его пределы.
- ✓ **OnFocus** – элемент получает фокус ввода.
- ✓ **OnBlur** – элемент теряет фокус ввода.
- ✓ **OnKeyPress** – пользователь нажимает и отпускает клавишу.
- ✓ **OnKeyDown** – пользователь нажимает клавишу над элементом.
- ✓ **OnKeyUp** – пользователь отпускает клавишу над элементом.
- ✓ **OnSubmit** – данные из формы переданы web-серверу.
- ✓ **OnReset** – форма очищена.
- ✓ **OnSelect** – пользователь выбирает текст в текстовом окне.
- ✓ **OnChange** – элемент теряет фокус ввода, после изменения значения элемента.

Для удобства работы с объектом определенного тега HTML можно присвоить ему имя. Это делается при помощи атрибута **name** тега, например так: ****. После назначения имени тегу изображения к его объекту можно обращаться не только как к элементу коллекции **images: document.images[N]**, где **N** – номер изображения в документе, но и по имени: **document.image1**. Имя объекта можно подставить также вместо номера в коллекцию: **document.images["image1"]**.

## Пример сценария JavaScript в HTML-документе

Для закрепления полученных знаний о языке JavaScript давайте сделаем небольшое приложение, выполняющее функцию калькулятора. Чтобы сделать удобный и красивый калькулятор, нам понадобится такой элемент HTML-страницы, как форма.

### Формы

Форма – это элемент страницы, позволяющий отправлять некие данные на сервер или передавать их сценариям JavaScript. Форма заключается в контейнер `<form></form>`. Составляющими частями формы являются поля ввода и кнопки, с помощью которых пользователь вводит данные. Подробнее о формах разговор пойдет в следующей главе, сейчас же рассмотрим ее элементы, необходимые нам для создания калькулятора.

Кнопки и поля ввода добавляются на HTML-страницу при помощи тега `<input>`. Обязательным атрибутом тега `<input>` является атрибут **name** – он задает название поля ввода. В зависимости от значения атрибута **type** меняется тип поля ввода.

Стандарт HTML поддерживает достаточно много типов полей ввода, и более подробно мы рассмотрим их в следующей главе. Сейчас же нам понадобятся только два типа: **text** и **button**.

Задав значение атрибута **type** равным значению «ТЕХТ», вы получите текстовое поле ввода (Рис. 5.9). С помощью этого поля вы будете вводить числа в программу калькулятора.

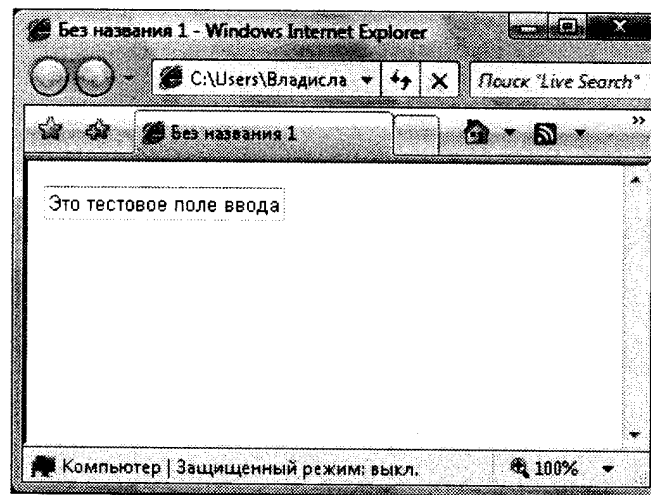


Рис. 5.9. Текстовое поле ввода в документе HTML

Для текстового поля ввода можно задать следующие атрибуты:

- ✓ **size**. Этот атрибут устанавливает размер поля ввода текста.
- ✓ **maxlength**. Устанавливает максимальное количество символов, которые можно ввести в поле ввода текста или пароля.
- ✓ **value**. Устанавливает значение поля по умолчанию.

Сделав значение атрибута **type** равным значению «**button**», вы создадите кнопку (Рис. 5.10). Атрибут **value** задает текст, отображаемый на кнопке.

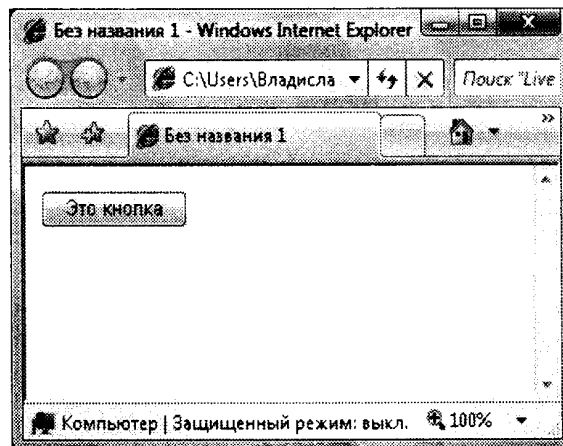


Рис. 5.10. Кнопка в документе HTML

## Принцип работы программы

Нажатиями на кнопки мы будем производить вычисления над числами. За основу калькулятора возьмем стандартный калькулятор Windows (Рис. 5.11). Чтобы было интереснее, используем инженерный его вариант с некоторыми упрощениями. Чтобы получить представление о том, что получится в результате, следует запустить калькулятор и немного поработать с ним.

- Чтобы запустить калькулятор, в главном меню Windows выберите команду **Все программы** ♦ **Стандартные** ♦ **Калькулятор**.
- Чтобы перевести калькулятор Windows в инженерный вид, выберите команду меню калькулятора **Вид** ♦ **Инженерный**.

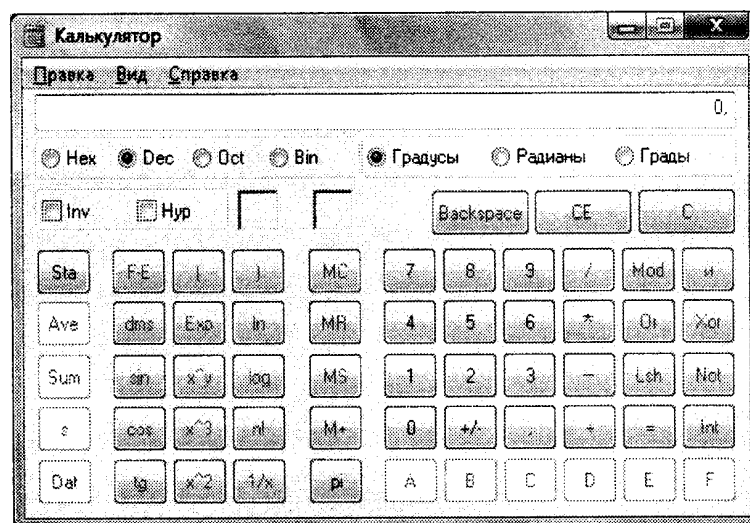


Рис. 5.11. Стандартный калькулятор Windows

После того, как ознакомление с калькулятором Windows будет закончено, займемся созданием аналогичного калькулятора на странице HTML.

## Основные части калькулятора

Наш калькулятор будет состоять из нескольких основных частей: области ввода чисел, области основных арифметических действий, области математических функций и области работы с памятью калькулятора. Для упрощения программы числа будут вводиться непосредственно в текстовом поле ввода, кнопок с цифрами делать не будем. Весь калькулятор будет оформлен в виде HTML-таблицы, с группами элементов в отдельных ячейках. Полученная в итоге страница будет выглядеть так, как показано на Рис. 5.12.

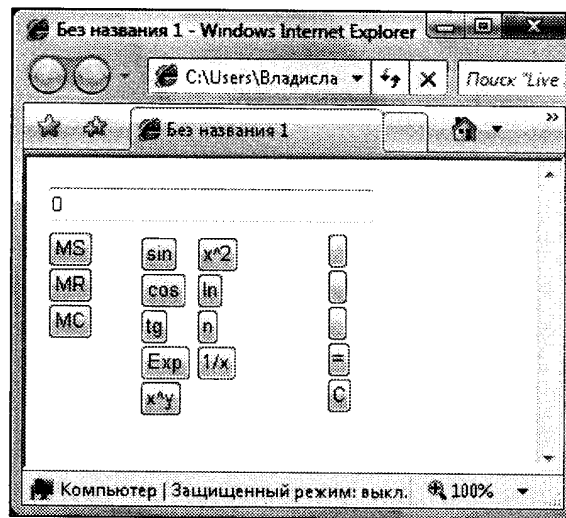


Рис. 5.12. Калькулятор, написанный на JavaScript

Костяк калькулятора составляет форма, в которую вставлена таблица с кнопками и полем ввода (Листинг 5.17). Необходимо связать действия с нажатием на кнопки калькулятора. Для этого к кнопкам добавляется реакция на щелчок мышью событие **onClick**.

Листинг 5.17. Форма с элементами управления калькулятором

```
<form action=»» name=»calc» id=»calc»>
  <table cellspacing=»2» cellpadding=»2» border=»0»>
    <tr>
      <td colspan=»3»><input type=»text» name=»numbers» value=»0»
size=»30» /><br /></td>
    </tr><tr>
      <td valign=»top»>
        <input type=»button» name=»memosave» value=»MS» /><br />
        <input type=»button» name=»memorecall» value=»MR» /><br />
        <input type=»button» name=»memoclear» value=»MC» /></td>
      <td valign=»top»>
        <table cellspacing=»3» cellpadding=»0» border=»0»>
          <tr valign=»top»><td>
```

```



```

Форме присвоено имя **calc**, поле ввода носит название **numbers**. Поле является дочерним объектом объекта **form**, а тот, в свою очередь, является потомком объекта **document**. Значение поля **value** является свойством объекта поля. Поэтому обращение к значению поля записывается как **document.название\_формы.название\_поля.value**, а в данном случае, как **document.calc.numbers.value**.

## Вычисление функций одного аргумента

Проще всего вычислять функции одного аргумента, например натуральный логарифм или возведение в квадрат. Достаточно в ответ на щелчок мышки присвоить полю значение, рассчитанное соответствующим методом объекта **Math**: **onClick = «document.calc.numbers.value = Math.log (document.calc.numbers.value)»**.

В объекте **Math** нет метода расчета факториалов, поэтому напишем свою функцию для их расчета, см. Листинг 5.18. Обратите внимание на начало функции. Здесь проверяется, является ли число, от которого берется факториал, целым. Если число равняется ему же, но округленному до целого, значит, оно и было целым, если же нет, то на экран выводится предупреждающий диалог. Если проверка пройдена, то производится вычисление факториала.

*Листинг 5.18. Функция расчета факториалов*


---

```
function factorial(n) {
  if(Math.round(n)-n!=0){
    window.alert('факториалы можно брать только от целых чисел')
    return 0
  }
  else{
    if ((n == 0) || (n == 1))
      return 1
    else {
      result = (n * factorial(n-1) )
      return result
    }
  }
}
```

---

**Вычислений функций с двумя аргументами**

Сложнее дело обстоит с операциями, в которых производятся действия с двумя аргументами, например умножение или деление. Выполнение таких операций проводится в два этапа. Сначала пользователь вводит в поле ввода первое число, затем щелкает мышью на кнопке выбранной математической операции. В ответ на щелчок кнопка вызывает функцию, которая сохраняет значение поля ввода и выбранную операцию в специальные переменные. Затем пользователь вводит второе число и щелкает мышью на кнопке =. Кнопка вызывает функцию, которая берет из переменных первое число и название операции, из поля ввода берет второе число, проводит вычисления и возвращает результат. Как выглядят эти функции, показано в Листинге 5.19. Функция **storeop** сохраняет первый аргумент математической операции **variable** и тип операции **op** в переменных **x** и **oper**. Функция **total** производит вычисление результата математической операции с первым аргументом – **x** и вторым аргументом – **y**. Тип операции выбирается в зависимости от значения переменной **oper**.

*Листинг 5.19. Вычисление функций с двумя аргументами*


---

```
var x=0, oper=»»
function storeop(variable, op){
  x=variable
  oper=op
  return true
}
function total(y)
{
```

```

if (oper=='pow')
    return Math.pow(x,y)
if (oper=='div')
    return x/y
if (oper=='mult')
    return x*y
if (oper=='plus')
    return x+y
if (oper=='min')
    return x-y
if (oper=='')
    return 0
oper=''
}

```

Щелчок на одной из кнопок математических операций с двумя аргументами приводит к выполнению функции **storeop** и очищению поля ввода: `<input type="button" name="mult" value="" onClick="storeop(document.calc.numbers.value,'mult'); document.calc.numbers.value="">`. Щелчок на кнопке = вызывает функцию **total**, результаты ее выполнения записываются в поле ввода: `<input type="button" name="equals" value="" onClick="document.calc.numbers.value = total(document.calc.numbers.value)">`.

## Работа с памятью

Последняя часть калькулятора – память. За эту часть отвечают две функции – **vsave** и **vrecall** и одна переменная – **tvalue**, см. Листинг 5.20. Функция **vsave** сохраняет значение переданного ей аргумента в переменной **tvalue**. Функция **vrecall** возвращает значение этой переменной.

*Листинг 5.20. Функции для работы с памятью калькулятора*

```

var tvalue=0;
function vsave(v)
{
    tvalue=v
    return true
}
function vrecall()
{
    return tvalue
}

```



При щелчке на кнопке записи в память **MS** вызывается функция **vsave**, ей передается текущее состояние поля ввода: `<input type="button" name="memosave" value="MS" onClick="vsave(document.calc.numbers.value)">`. Щелчок на кнопке восстановления из памяти, **MR**, записывает в поле ввода значение, возвращаемое функцией **vrecall**: `<input type="button" name="memorecall" value="MR" onClick="document.calc.numbers.value = vrecall()">`. Щелчок на кнопке очищения памяти **MC** вызывает функцию **vsave** с нулем в качестве аргумента: `<input type="button" name="memoclear" value="MC" onClick="vsave(0)">`.

## Общий код программы

Теперь, после того как рассмотрены все составляющие части калькулятора, можно записать его код целиком (см. Листинг 5.21).

*Листинг 5.21. Код калькулятора в документе HTML*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Калькулятор</title>
<script language="JavaScript">
<!--
var x=0, oper="»», tvalue=0;
function factorial(n) {
if(Math.round(n)-n!=0) {
    window.alert('факториалы можно брать только от целых чисел')
    return 0
}
else{
    if ((n == 0) || (n == 1))
        return 1
    else {
        result = (n * factorial(n-1) )
        return result
    }
}
}
function storeop(variable, op){
x=variable
```

```
oper=op
return true
}
function total(y)
{
if (oper=='pow')
    return Math.pow(x,y)
if (oper=='div')
    return x/y
if (oper=='mult')
    return x*y
if (oper=='plus')
    return x+y
if (oper=='min')
    return x-y
if (oper=='')
    return 0
oper=''
}
function vsave(v) {
tvalue=v
return true
}
function vrecall(){
return tvalue
}
-->
</script>
</head>
<body>
<h3>Калькулятор</h3>
<form action=»» name=»calc» id=»calc»>
    <table cellspacing=»2» cellpadding=»2» border=»0»>
    <tr>
        <td colspan=»3»><input type=»text» name=»numbers» value=»0»
size=»30» /><br /></td>
```

```

</tr><tr>
  <td valign=»top»><input type=»button» name=»memosave»
value=»MS» onClick = «vsave (document.calc.numbers.value)»
/><br />
  <input type=»button» name=»morecall» value=»MR» onClick =
«document.calc.numbers.value=vrecall()» /><br />
  <input type=»button» name=»memoclear» value=»MC»
onClick=»vsave(0)» />
</td><td valign=»top»>
  <table cellspacing=»3» cellpadding=»0» border=»0»>
    <tr valign=»top»><td>
      <input type=»button» name=»sin» value=»sin» onClick = «docu-
ment.calc.numbers.value = Math.sin(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»cos» value=»cos» onClick = «docu-
ment.calc.numbers.value = Math.cos(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»tg» value=»tg» onClick = «docu-
ment.calc.numbers.value = Math.tan(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»Exp» value=»Exp» onClick = «docu-
ment.calc.numbers.value = Math.exp(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»pow» value=»x^y» onClick =
«storeop(document.calc.numbers.value,'pow'); document.calc.num-
bers.value=' '» /><br />
    </td><td>
      <input type=»button» name=»sqr» value=»x^2» onClick = «docu-
ment.calc.numbers.value *= document.calc.numbers.value» /><br
/>
      <input type=»button» name=»ln» value=»ln» onClick = «docu-
ment.calc.numbers.value = Math.log(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»fact» value=»n!» onClick = «docu-
ment.calc.numbers.value = factorial(document.calc.numbers.
value)» /><br />
      <input type=»button» name=»inv» value=»1/x» onClick = «docu-
ment.calc.numbers.value = 1/document.calc.numbers.value» /><br
/>
    </td></tr>
  </table>

```

```
</td><td valign=»top»>
  <input type=»button» name=»divide» value=»/» onClick =
«storeop(document.calc.numbers.value,'div'); document.calc.num-
bers.value = ''» /><br />
  <input type=»button» name=»mult» value=»*» onClick =
«storeop(document.calc.numbers.value,'mult'); document.calc.
numbers.value=''» /><br />
  <input type=»button» name=»minus» value=»-» onClick =
«storeop(document.calc.numbers.value,'min'); document.calc.num-
bers.value=''» /><br />
  <input type=»button» name=»equals» value=»=» onClick = «docu-
ment.calc.numbers.value = total(document.calc.numbers.value)»
/><br />
<input type=»button» name=»clear» value=»C» onClick = «document.
calc.numbers.value = ''» /><br />
  </td></tr>
</table>
</form>
</body>
</html>
```

---

## Заключение

Итак, вы создали свою первую динамическую web-страницу, которая в ответ на действия посетителя выполняет сценарии JavaScript, проводит расчеты и выводит полученный результат на экран. Поздравляем вас, вы сделали большой шаг от статических технологий к динамическим. Теперь перед вами открыты замечательные возможности по созданию web-страниц, дружественных посетителю, реагирующих на каждый его запрос, подстраивающихся под его нужды.

## ГЛАВА 6.

# Размещение сайта в сети Интернет



Познав суть технических тайнств создания сайтов для сети Интернет, вы, несомненно, почувствуете настойчивое желание обнародовать свои творения, представить продукт своего труда в естественной среде обитания пользователей Всемирной сети – на Интернет-сервере. В этой главе вы узнаете, как можно выложить web-сайт в Интернет, как заявить всему Интернет-сообществу о существовании сайта, как отвоевать свое место под солнцем у конкурентов и что делать с завоеванными территориями, чтобы не утратить свои владения.

## Варианты размещения web-сайта в сети Интернет

Итак, ваш сайт готов, настала пора познакомить с ним обитателей сети Интернет. Как это сделать? Существует множество способов выложить ваш сайт в сети Интернет.

### Виртуальный хостинг

Вашему сайту выделяется каталог на web-сервере, из которого сайт и запускается. Название хостинг происходит от слова хост (от англ. Host – главный, хозяин). Хостом называется сервер, на котором размещается каталог с сайтом. Таких каталогов на сервере может быть несколько сотен и даже тысяч, но работать эта система будет так, будто для сайта выделен отдельный сервер, поэтому к слову хостинг часто добавляется слово «виртуальный».

Преимуществом такой системы является доступность. В зависимости от условий, предоставляемого дискового объема, ежемесячного Интернет-трафика и наличия различных дополнительных услуг, виртуальный хостинг стоит от пяти долларов в месяц. Некоторые хостинг-провайдеры, т. е. фирмы, занимающиеся предоставлением хостинга, предоставляют виртуальный хостинг и вовсе бесплатно.

Недостатки виртуального хостинга являются продолжением его достоинств. Вычислительную мощность сервера и пропускную способность канала, по которому сервер подключен к Интернету, вашему сайту придется делить еще с несколькими сотнями конкурентов. Поэтому скорость выполнения сценариев на вашем сайте и быстрота доступа к страницам могут желать лучшего. В таких случаях лучше воспользоваться одним из других способов размещения сайта в Интернете.

Другим недостатком виртуального хостинга является зависимость от программного обеспечения, установленного на сервере. Вы не сможете поменять web-сервер, установленный в системе, подключить поддержку дополнительных языков серверных сценариев и т. д.

## Выделенный сервер

Этот вариант позволяет избавиться от большинства недостатков виртуального хостинга. Для сайта выделяется отдельный сервер, на который устанавливается программное обеспечение, по выбору владельца сайта.

Основными недостатками этого способа размещения сайтов являются достаточно высокая цена такого решения и необходимость использовать только то оборудование, которое предоставляет хостинг-провайдер. Кроме того, свобода в выборе программного обеспечения сервера приносит с собой необходимость это программное обеспечение устанавливать и настраивать, так как хостинг-провайдер несет в данном случае ответственность только за аппаратную часть сервера.

Выделенные серверы используются, как правило, для размещения достаточно крупных коммерческих проектов, окупающих затраты на поддержание и администрирование сервера. Еще одним возможным случаем использования выделенного сервера является размещение на нем нескольких web-проектов, работающих одновременно. Использование отдельного сервера позволяет более гибко контролировать работу каждого из проектов по сравнению с использованием виртуального хостинга.

## Совместное размещение

Совместное размещение (на сленге – «co-location» – совместное размещение) – это подключение сервера, принадлежащего заказчику, к Интернет-каналу хостинг-провайдера. Содержимое сервера, как программное, так и аппаратное, полностью определяется потребностями владельца сайта или сайтов, размещенных на сервере. Хостинг-провайдер предоставляет только выход в Интернет, электропитание сервера и физическое пространство для размещения сервера.

Совместное размещение позволяет удовлетворить потребности любого, самого взыскательного владельца сайта. Но этот вариант приносит с собой и некоторое количество проблем, связанных с тем, что обслуживание сервера полностью ложится на плечи владельца сайта, за работоспособность оборудования и программ хостинг-провайдер ответственности не несет.

Используется совместное размещение практически в тех же случаях, что и выделенный сервер: для крупных проектов с высокой посещаемостью либо для размещения нескольких web-сайтов одного владельца.

## Размещение сайта на своем компьютере

Одним из вариантов обеспечения доступа к сайту из Интернета является размещение сайта на своем компьютере, подключенном к Сети.

Преимущества такого решения ясны. Нет необходимости платить за оборудование хостинг-провайдера и за размещение сервера, только за Интернет-трафик. Можно устанавливать любое программное и аппаратное обеспечение без каких-либо ограничений. Сервер находится всегда под рукой, а не у хостинг-провайдера, и т. д.

Минусов у такого варианта тоже достаточно много. Необходимо быстрое и постоянное подключение к Интернету с выделенным IP-адресом. Требуется постоянно держать компьютер включенным и выделять часть ресурсов компьютера на обслуживание сервера и т. д.

На своем личном компьютере имеет смысл размещать только web-сайты, для которых не будет критичным периодическое пропадание из сети, при отключении электричества и перезагрузках. Также не стоит размещать таким образом сайты с большой посещаемостью и непростыми серверными сценариями, иначе нагрузка на компьютер будет слишком велика.

## Особенности бесплатного хостинга

Самым простым и популярным способом размещения web-сайтов в сети был и остается виртуальный хостинг. Это решение подходит для большинства случаев, именно этот вариант мы и рассмотрим подробнее.

Как уже отмечалось ранее, виртуальный хостинг бывает платным и бесплатным. Разумеется, ничего полностью бесплатного не существует, и бесплатный хостинг на самом деле таким не является. Услуги бесплатных хостинг-провайдеров приходится оплачивать не деньгами, а чем-то другим.

Чаще всего за бесплатный хостинг приходится платить размещением рекламы на своем сайте. К тому же, хостинг-провайдер не отвечает ни за что, качество услуг может быть неудовлетворительным.

Существует и другой вариант, когда бесплатный хостинг идет в дополнение к какой-нибудь другой услуге или товару. Например, большинство Интернет-провайдеров в числе прочих услуг предлагают своим клиентам бесплатно разместить web-страницу. Этого, кстати, делать не стоит, поскольку в том случае, если вы поменяете Интернет-провайдера, вы потеряете и свою web-страницу.

Кроме того, большинство хостинг-провайдеров предлагает на своих бесплатных тарифных планах очень маленький набор дополнительных услуг, маленький размер дискового пространства, ограничение на размер и типы файлов. Эти ограничения вводятся, в том числе, и с надеждой на то, что пользователь перейдет со временем на платный тарифный план, не меняя хостинг-провайдера.

Бесплатный хостинг идеально подходит для размещения ваших первых web-сайтов, поскольку многочисленные эксперименты с сайтами и первые «блины комом», которых вы вряд ли избежите, не будут вам стоить ничего. В дальнейшем, когда эксперименты закончатся, вы сможете перейти на платный хостинг, либо вообще поставить выделенный сервер, но для некоторых проектов можно так и ограничиться бесплатным хостингом. Если вы планируете впоследствии поменять хостинг-провайдера, то выбирайте хостинг с поддержкой сторонних доменов второго уровня, чтобы поменять сервер с минимальными проблемами.

## Рекомендации по использованию бесплатного хостинга

Чтобы в дальнейшем у вас не возникало затруднений, четко оговорим область применения бесплатного хостинга. Бесплатный хостинг можно применять для:

- ✓ Экспериментов. Это как раз то, чем мы с вами будем заниматься в этой главе.
- ✓ Домашних страниц. Если вы создаете страничку для неформальных целей, чтобы познакомить Интернет-сообщество с собой, своими увлечениями и пристрастиями и т. д., нет необходимости за это платить деньги. Если человек захотел узнать о вас побольше, он, скорее всего, не будет очень расстраиваться и не уйдет рассерженно, если ваш сайт будет грузиться не слишком быстро, а в углу примостится рекламный баннер.
- ✓ Некоммерческих проектов. Если проект не приносит денег, то желательно, чтобы и затраты были как можно меньше, поэтому бесплатный хостинг в таком случае является хорошим решением. Хотя, если посещаемость вашего ресурса будет достаточно велика, можно вместо рекламы, вставляемой бесплатным хостингом, повесить на сайте рекламный баннер и оплачивать из вырученных денег услуги платного хостинга.
- ✓ Фан-ресурсов на начальной стадии работы. Допустим, вы создали сайт, посвященный вашему увлечению, ваше увлечение разделяют другие люди, и им будет интересно посещать ваш сайт. Но пока о нем еще никто не знает, и на уровень окупаемости, например за счет показа рекламы, сайт выйдет нескоро. В этом случае можно сначала разместить сайт на сервере бесплатного хостинг-провайдера, а затем, после повышения популярности ресурса, перейти на платный хостинг. Хотя лучшим вариантом будет все-таки и начинать с платного хостинга.

Бесплатный хостинг не следует использовать для:

- ✓ Солидных коммерческих и официальных сайтов – представительств фирм, общественных организаций, государственных учреждений. Доменный адрес, принадлежащий бесплатному хостинг-провайдеру, наверняка подорвет доверие к солидной фирме, которая экономит несколько долларов на своем сайте. Кроме того, хозяева бесплатных хостингов, как правило, не очень хорошо относятся к размещению коммерческих сайтов на их территории.
- ✓ Крупных проектов. Бесплатный сервер просто захлебнется от наплыва посетителей и одновременного запуска кучи серверных сценариев.
- ✓ Файловых архивов. Хостинг-провайдеры не жалуют и всячески противодействуют такому способу использования бесплатного web-пространства. Ограничивают максимальный размер файлов, запрещают закачивать файлы с определенными разрешениями и т. д.

## Требования, предъявляемые к серверу бесплатного хостинга

Итак, для начала нам нужен бесплатный хостинг. Давайте разберемся, какие основные требования необходимо предъявлять к бесплатным хостингам, на что следует обратить внимание.



## Скорость работы сервера

Это требование понятно. Чем быстрее сервер будет реагировать на запросы пользователя, тем ниже вероятность, что пользователь закроет web-сайт, не дождавшись загрузки. Скорость работы определяется экспериментальным путем. Либо создайте и проверьте работу тестовой страницы на сервере, либо походите по страницам, уже размещенным на этом бесплатном хостинге.

## Ограничения на типы и размеры файлов

Многие бесплатные хостинг-провайдеры устанавливают ограничение на размер и тип размещаемых файлов. Чаще всего web-страницы содержат достаточно небольшие HTML-файлы и картинки, поэтому в таких ограничениях нет ничего страшного.

Но не исключен вариант, что вам понадобится разместить на своем сайте несколько музыкальных композиций в формате **mp3**, видеоклипы или что-нибудь столь же объемное. В таком случае при выборе бесплатного хостинга обратите внимание на условия размещения файлов.

## Место, выделяемое под сайт

Среднестатистический web-сайт, особенно недавно начавший работу, занимает довольно мало места, и двадцати мегабайт минимального объема, предлагаемого бесплатными хостинг-провайдерами, вам скорее всего хватит с хорошим запасом. К тому же на многих бесплатных хостингах предоставляется возможность впоследствии увеличить предоставляемый объем дискового пространства. Поэтому этот параметр не слишком важен. Смотрите только, чтобы вам не предоставили объем порядка одного-двух мегабайт, этого места вам скорее всего не хватит.

## Возможность работы с CGI и PHP

Если вы хотите разместить на бесплатном хостинге динамический web-сайт, использующий серверные сценарии, то от хостинга вам потребуется поддержка технологий CGI и языка PHP. К сожалению, далеко не все бесплатные хостинг-провайдеры поддерживают такую возможность.

Чаще всего хостингом предоставляется некий предустановленный набор сценариев, создающих гостевую книгу, чат, набор счетчиков и т.д., без возможности добавить свои сценарии. Если вы размещаете на сервере домашнюю страницу, то предустановленные сценарии вам еще могут быть полезны, если же вы делаете какой-то более-менее навороченный динамический сайт, то вам, безусловно, понадобится полноценная поддержка серверных сценариев.

## Реклама на странице

Бесплатность размещения web-сайтов хостинг-провайдер окупает, как правило, размещением своей рекламы на страницах сайтов, хотя некоторые хостеры получают прибыль с бесплатных страниц другими способами, например, привлекая новых клиентов на свой платный хостинг или повышая с помощью бесплатных страниц популярность своего доменного имени.

Реклама, размещаемая хостинг-провайдерами, делится на два вида: рекламные баннеры на страницах сайтов и динамические рекламные вставки, рор-уп'ы (от англ. рор-уп – выскочить), закрывающие часть страницы, а затем автоматически исчезающие с экрана. Выбирайте из двух зол ту, что кажется вам меньшей. Баннер можно органично вписать в дизайн вашей страницы, зато динамические вставки, исчезнув, не отвлекают посетителя и не уменьшают рабочий объем страницы.

### Адрес сайта

Большинство бесплатных хостингов предлагают своим пользователям доменные имена третьего уровня **http://www.название\_страницы.имя\_хостинг\_провайдера.ru**, например: **http://www.vasya.narod.ru** или **http://www.rot.front.ru**. Другим вариантом является представление сайта пользователя, как папки на сервере хостинг-провайдера, например **http://www.chat.ru/~username**. Такое имя выглядит хуже, создается впечатление, будто ваш сайт не самостоятельный проект, а просто один из разделов сервера, да и запомнить такое название затруднительно.

Следует отметить, что адрес сайта, указывающий на один из популярных серверов бесплатных страниц, негативно сказывается на рейтинге страницы. Дело в том, что большинство страниц, размещаемых на серверах бесплатных страниц, например, **narod.ru** или **by.ru**, – это слепленные по одному шаблону, скучные и некрасивые страницы, все содержание которых сводится к выражению типа «здесь был Вася». Чтобы доменное имя сервера не работало против вас, выберите хостинг-провайдера с не таким раскрученным именем. Но здесь вас поджидают другие подводные камни. Если доменное имя не популярно, то, возможно, для этого есть объективные причины. Например, хостинг не очень хорош сам по себе или недавно открыт и еще не пережил «болезни роста», негативно сказывающиеся на стабильности и удобстве работы.

Некоторые бесплатные хостинг-провайдеры предлагают услугу по размещению сайта пользователя под произвольным доменом второго уровня. Но для этого данный домен необходимо приобрести, заплатив некоторую сумму денег. Преимущество такого решения заключается в том, что даже перейдя впоследствии на платный хостинг, вы сохраните свое доменное имя, а недостаток – за него надо платить деньги. А учитывая тот факт, что заказывая платный хостинг у некоторых компаний, домен второго уровня вы получаете бесплатно, имеет смысл подумать о платном хостинге. Если вы уже готовы платить за свой сайт, то почему бы не заплатить и за хостинг, избавившись от всех недостатков бесплатного сервера.

### Способ обновления сайта

Некоторые бесплатные хостинг-провайдеры предлагают единственный способ обновления файлов на сайте – через web-форму, с помощью браузера. Безусловно, таким способом можно добавить 1-2 файла на сайт, но если вам придется таким образом обновлять несколько сотен файлов, вы недобрым словом помянете разработчиков этого инструмента. Единственно правильный способ работы с файлами на сервере – по протоколу FTP (File Transfer Protocol – Протокол передачи файлов), когда вы можете обращаться с файлами сайта, как с файлами в папке на вашем

жестком диске, т.е. просто и функционально. Обязательно обращайте внимание на этот параметр при выборе хостинга, благо возможность FTP-доступа предоставляет большинство хостинг-провайдеров.

## Сравнение различных серверов бесплатного хостинга

Найти сервера бесплатного хостинга можно, просто введя в поисковую систему **Yandex** запрос «бесплатный хостинг». Сейчас же мы перечислим и отметим основные особенности нескольких популярных бесплатных хостинг-провайдеров (Табл. 6.1).

Табл. 6.1. Сравнительная характеристика бесплатных хостинг-провайдеров

Хостер	Дисковое пространство	Адрес сайта	Серверные сценарии	Способы обновления	Тип рекламы
<b>www.narod.ru</b>	Неогр.	<b>www.назв._сайта.narod.ru</b>	Только предустановленные	Web-форма, FTP	Всплывающее окно
<b>by.ru</b>	Неогр.	<b>www.назв._сайта.by.ru</b> , поддержка произвольных доменов второго и третьего уровней	Поддержка CGI и PHP	Web-форма, FTP	Баннер
<b>www.webservis.ru</b>	Неогр.	<b>Назв._сайта.ai.ru</b> <b>Назв._сайта.bip.ru</b> <b>Назв._сайта.dem.ru</b> <b>Назв._сайта.fud.ru</b> <b>Назв._сайта.hobi.ru</b>	Поддержка CGI и PHP, набор предустановленных сценариев	Web-форма, FTP	Баннер
<b>www.pochta.ru</b>	20 Мб	<b>www.назв._сайта.pochta.ru</b> , возможны варианты: <b>fromru.com</b> , <b>front.ru</b> , <b>hotbox.ru</b> , <b>krovatka.net</b> , <b>land.ru</b> , <b>mail15.com</b> , <b>mail333.com</b> , <b>pisem.net</b> , <b>pochtamt.ru</b> , <b>pop3.ru</b> , <b>rbcmail.ru</b> , <b>smtp.ru</b>	Нет	Web-форма, FTP	Без рекламы
<b>www.holm.ru</b>	Неогр.	<b>www.назв._сайта.h15.ru</b>	Поддержка CGI, PHP	FTP	Баннер
<b>www.hut.ru</b>	100 Мб, с возможностью дальнейшего увеличения	<b>www.назв._сайта.hut.ru</b> , возможна поддержка произвольных доменов второго уровня	Поддержка CGI-сценариев на языке PERL, PHP	FTP	Баннер вверху страницы

Хостер	Дисковое пространство	Адрес сайта	Серверные сценарии	Способы обновления	Тип рекламы
<b>www.mail.ru</b>	50 Мб, возможно увеличение	<b>www.название_сайта.mail.ru</b>	Сценарии не поддерживаются	Web-форма, FTP	Баннер вверху всех страниц, размером более 2 Кбайт
<b>www.chat.ru</b>	10 Мб	<b>www.назв._сайта.chat.ru</b> , либо <b>www.chat.ru/~название_сайта</b>	Только предоставленные сценарии	Web-форма, FTP	Баннер

## Регистрация на сервере бесплатного хостинга

Допустим, вы выбрали себе бесплатный хостинг по вкусу, теперь необходимо на нем зарегистрироваться. Процедура регистрации зависит от сервера хостинга, но в общем и целом они похожи. Рассмотрим эту процедуру на примере сервера <http://www.narod.ru>.

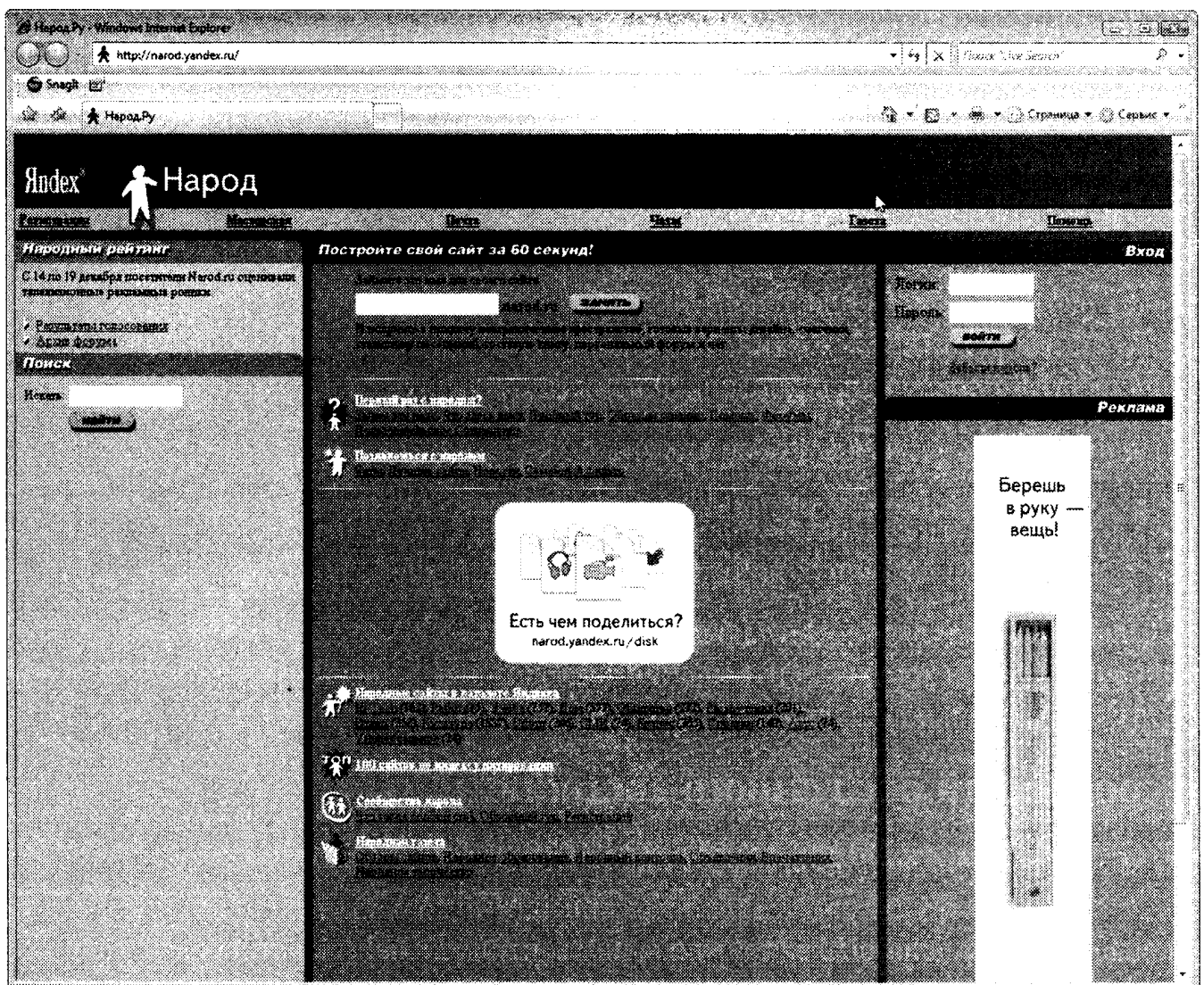


Рис. 6.1. Главная страница сайта хостинг-провайдера **Narod.ru**

- В адресной строке браузера введите адрес сайта хостинг-провайдера **http://www.narod.ru**, после чего откроется главная страница (Рис. 6.1). На этой странице вы найдете ссылку на страницу регистрации новых пользователей хостинга.
- Щелкните мышью на ссылке **Регистрация** в верхней левой части страницы. Откроется страница с регистрационной формой (Рис. 6.2).

Рис. 6.2. Страница регистрации нового пользователя

- Щелкните мышью на ссылке **Пользовательского соглашения** в нижней части страницы. Откроется страница, на которой будут приведены основные условия предоставления хостинга, ограничения на размещение информации и файлов и правила поведения, которым вы должны следовать, пользуясь услугами этого хостинг-провайдера.

Если вам подходят условия соглашения, закрывайте эту страницу и приступайте к заполнению регистрационной формы, если же какие-то пункты вам категорически не подходят, значит, этот хостинг не для вас, выберите другого хостинг-провайдера.

В регистрационной форме звездочками помечаются поля, обязательные к заполнению.

- Введите в поля ввода свое имя, желаемое название домена третьего уровня, адрес электронной почты, номер мобильного телефона. Эти данные будут использованы при размещении сайта в каталоге хостинг-провайдера. Наконец, укажите свой адрес электронной почты и пароль доступа.
- Щелкните мышью на кнопке **Зарегистрировать**. Если все данные введены правильно и выбранное вами доменное имя сайта еще не занято, вы будете зарегистрированы (Рис. 6.3). Если же в форме были ошибки либо выбранное доменное имя уже кем-то занято, вас попросят изменить часть введенных сведений.

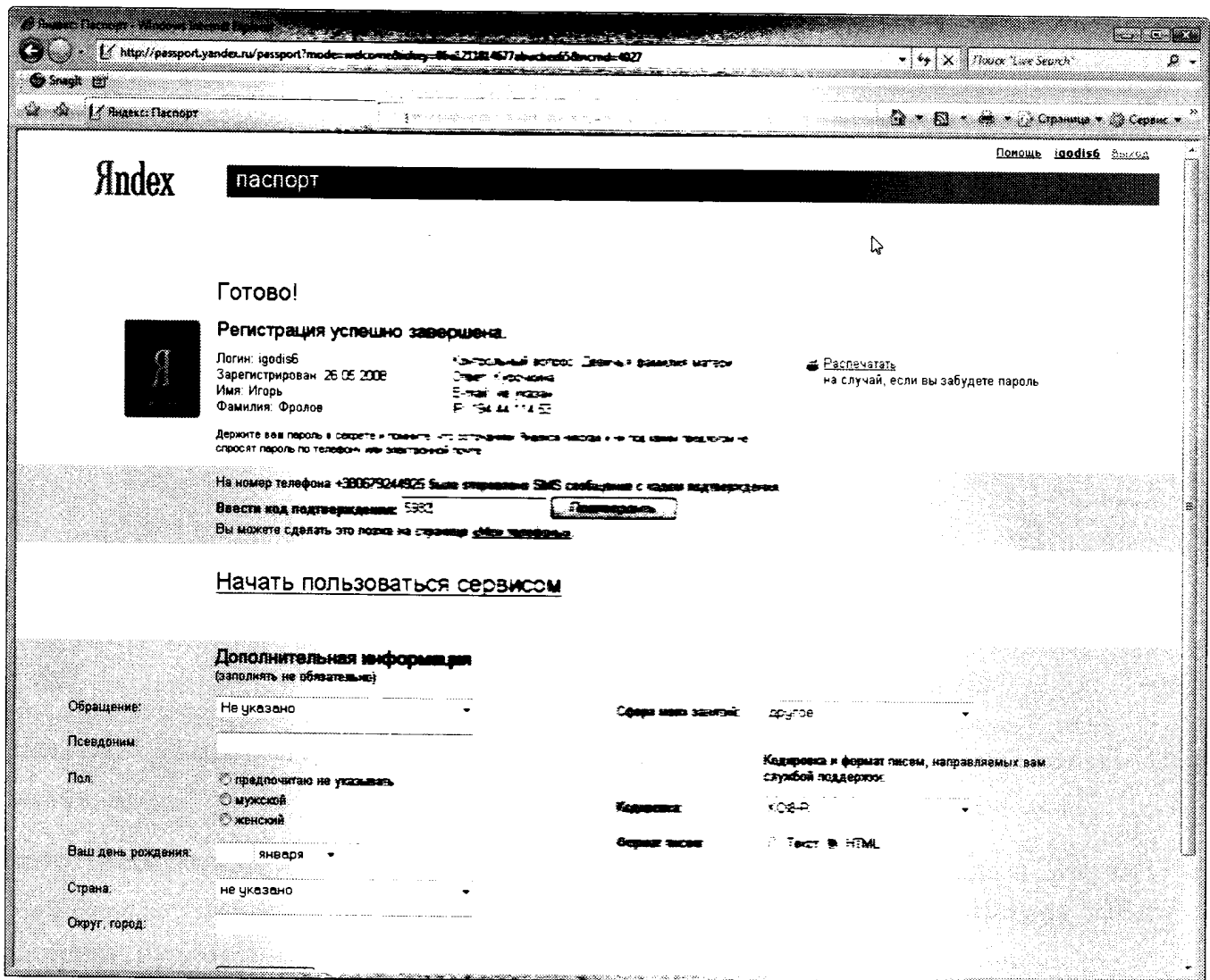


Рис. 6.3. Страница, сообщающая об успешной регистрации

Через некоторое время после выполнения процедуры регистрации, на ваш мобильный телефон придет SMS-сообщение с кодом регистрации.

- Введите значение кода регистрации в поле ввода **Ввести код подтверждения:** страницы сообщения об успешной регистрации и нажмите кнопку **Подтвердить**. Появится следующая страница с сообщением об окончании регистрации.
- На этой странице щелкните мышью на ссылке **Начать пользоваться сервисом**.

Вы увидите страницу **Мастерская** с группой элементов управления **Редактирование и управление** (Рис. 6.4). Щелкая мышью на необходимых ссылках, вы получаете доступ к разделам, позволяющим редактировать дизайн и содержание страниц сайта, загружать файлы на сервер. Войти на страницу **Мастерская** можно также из браузера:

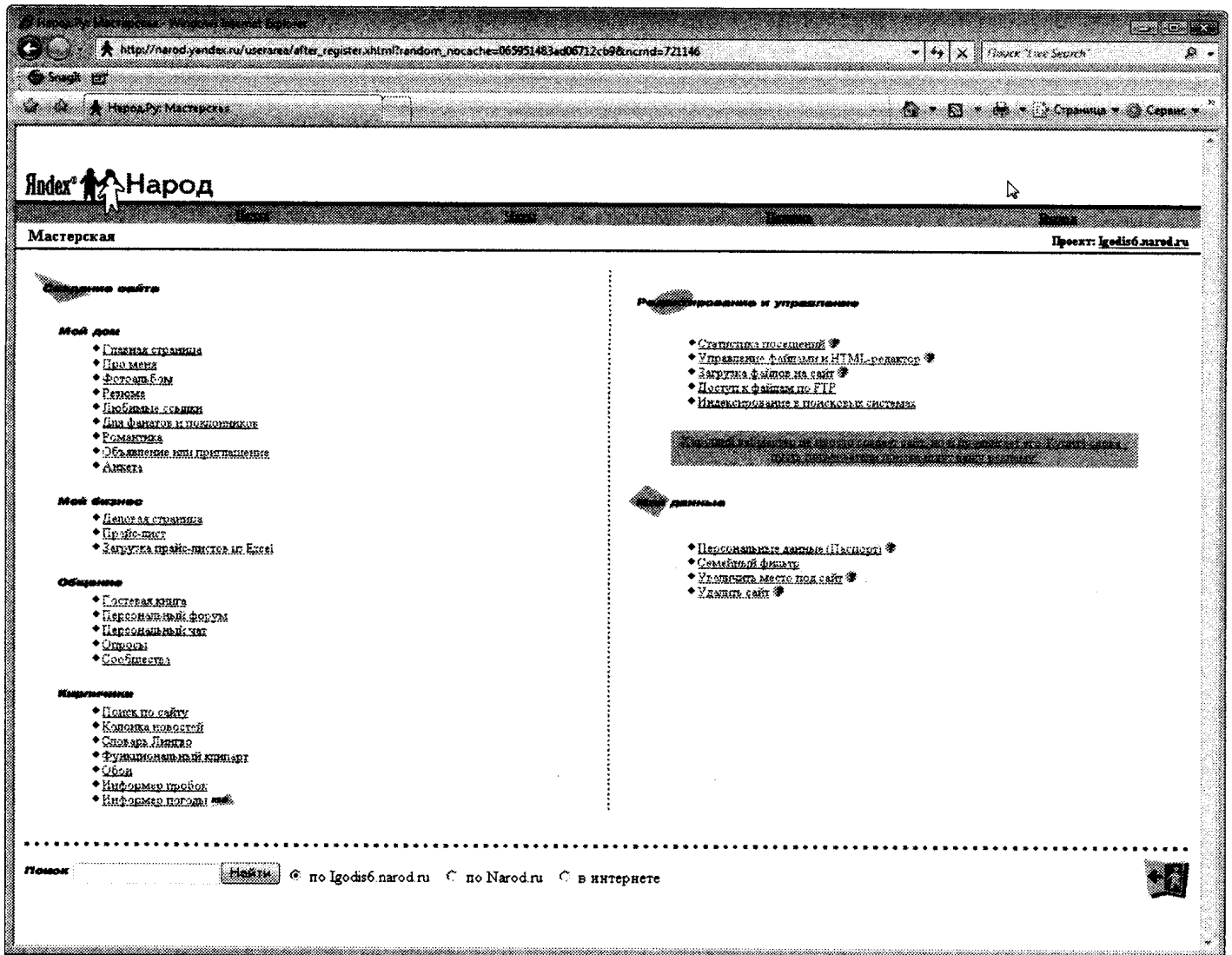


Рис. 6.4. Страница **Мастерская**

- Введите в адресную строку браузера адрес **http://narod.yandex.ru/userarea/after\_register.html**.
- В открывшемся окне входа в персональную часть введите логин и пароль, которые вы использовали при регистрации.

Вход на страницу **Мастерская** произойдет после нажатия кнопки **Войти**.

О том, каким образом загрузить свои собственные файлы на сервер, вы узнаете в следующем разделе.

## Загрузка сайта на сервер

Итак, вы создали учетную запись на сервере бесплатного хостинга, и теперь вам необходимо выложить на него свой сайт. Для выполнения этой задачи существует несколько способов.

## Использование web-интерфейса

Первый и самый простой способ – воспользоваться web-интерфейсом. Такую услугу предлагает большинство хостинг-провайдеров. У разных хостингов web-интерфейсы тоже разные, но в общих чертах они достаточно похожи. Рассмотрим web-интерфейс управления файлами, предлагаемый хостингом <http://www.narod.ru>. Получить доступ к web-интерфейсу можно через страничку **Мастерская** (Рис. 6.4).

- Перед загрузкой на сайт подготовьте и отредактируйте файлы, которые находятся на жестком диске вашего компьютера и предназначены для загрузки на сервер провайдера.
- Щелкните мышью на ссылке **Загрузка файлов на сайт** странички **Мастерская**. Откроется страница **Загрузка файлов** (Рис. 6.5).

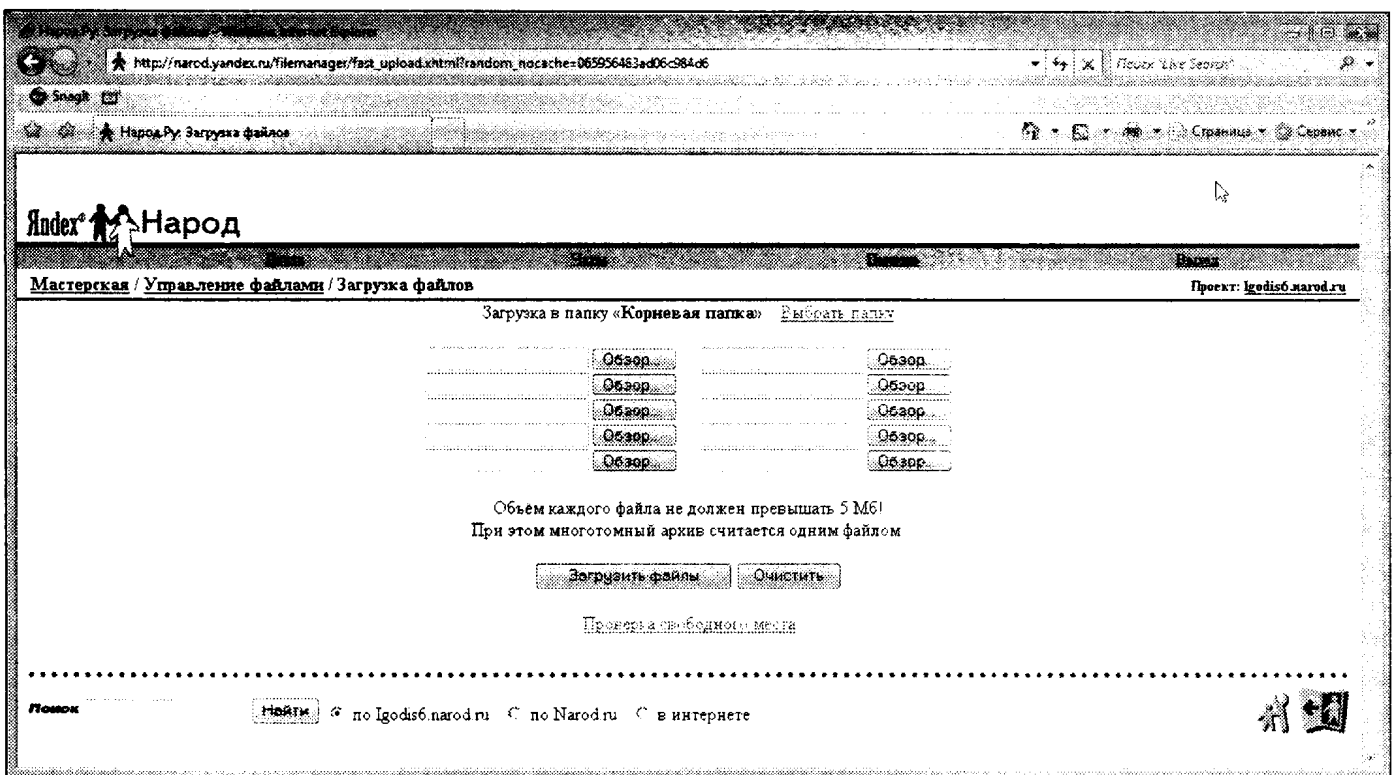


Рис. 6.5. Страница **Загрузка файлов**

Эта страница состоит из десяти полей ввода с кнопками **Обзор** справа от них и кнопкой **Загрузить файлы**. Щелкнув мышью на кнопке **Обзор**, вы откроете диалог выбора файлов. Здесь следует выбрать файл, который вы хотите отправить на сервер, и щелкнуть мышью на кнопке **Открыть**. Путь к файлу добавится в поле ввода.

- Чтобы отправить файл на сервер, щелкните мышью на кнопке **Загрузить файлы**, файл будет загружен на сервер.
- Если вам необходимо загрузить сразу несколько файлов, то перед нажатием кнопки **Загрузить файлы** заполните несколько полей **файл #N** и повторяйте процедуру до загрузки всех нужных файлов. Причем учтите, что файлы загружаются в текущий каталог.



- ▶ Чтобы загрузить файлы в другой каталог вашего виртуального сервера, предварительно перейдите в него, щелкнув мышью на ссылке **Выбрать папку**, и затем – на названии выбираемого каталога.

Очевидно, что **web**-интерфейс управления файлами лучше всего подходит для закачки на сервер 1-2 файлов. Эта система является незаменимой, если вам необходимо произвести некоторые незначительные изменения на сайте, немного подправить код страниц или произвести еще какую-нибудь косметическую правку.

## Использование протокола FTP

Если же вам необходимо загрузить на сервер несколько сотен файлов, лежащих к тому же в разных папках, работа с **web**-интерфейсом превращается в сущий ад. В таком случае незаменимым является доступ к серверу через протокол **FTP**. Для работы с сервером по протоколу **FTP** нужна программа, которая называется **FTP-клиент**, например **FileZilla**. Следует отметить, что **FTP-клиентов** существует великое множество, на любой вкус и любые потребности. Установив такую программу на своем компьютере, вы сможете с максимальным удобством и оперативностью загружать значительное количество файлов на сервер. Каждая такая программа предоставляет возможность выполнить сценарий загрузки файлов, который несложно освоить самостоятельно. Поэтому в этой главе мы не рассматриваем детально работу с конкретным **FTP-клиентом**.

## Заключение

Итак, мы с вами успешно прошагали по пути освоения технических методов создания сайта, освоили приемы выхода вашего продукта на просторы Всемирной сети.

Этой главой заканчивается познание премудростей **web**-мастеринга и дальше начинается реальная жизнь в сети Интернет.

# Практическое руководство

# Web-дизайн для начинающих



Эта книга содержит все необходимое для создания прекрасного web-сайта. Начав с создания простейших web-страниц, вы шаг за шагом построите собственный web-сайт, обладающий всеми качествами профессионального Интернет-портала. Вы познакомитесь с языком HTML и скриптами Java, украсите web-сайт графикой и наполните динамическим содержанием, а также научитесь работать в программе SharePoint Designer 2007 – профессиональном приложении для создания web-сайтов. Кроме того, вы узнаете, как разместить ваше творение в Интернете совершенно бесплатно.

## Рассматриваемые темы:

- Устройство Всемирной паутины
- Первые web-страницы своими руками
- Приемы, без которых нельзя обойтись
- Web-дизайн и графика
- Web-сайт с динамическими страницами
- Размещение сайта в сети Интернет

9 785893 924305  
ООО «Книжный Лабиринт»  
Самойлов Е.Э.  
Web-дизайн для  
начинающих. Быстрый старт  
цена: 187,00 р 08.05.2009  
т-магазин  
180475 /пеньки»®  
3st.ru

Телефон для товароведов:  
(495) 720-07-65  
E-mail: opt@triumph.ru