

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

А.В. Белозубов, Д.Г. Николаев

Основы работы с HTML-редактором Adobe Dreamweaver CS3

Учебно-методическое пособие

Санкт-Петербург

2007

УДК 681.3

Белозубов А.В., Николаев Д.Г. Основы работы с HTML-редактором Adobe Dreamweaver CS3. Учебно-методическое пособие. – СПбГУ ИТМО, 2007. – 112 с.

Рецензенты: Л.С. Лисицына, к.т.н., доцент, зав. каф. КОТ СПбГУ ИТМО
А.А. Бобцов, д.т.н., профессор каф. СУиИ СПбГУ ИТМО

Учебно-методическое пособие предназначено для использования в учебном курсе “Информатика” по ряду специальностей и направлений подготовки студентов университета, а также для поддержки курсов повышения квалификации работников образования по программе «Основы ИКТ для применения в образовательной деятельности» по заказу Комитета по образованию Санкт-Петербурга.

ISBN 5-7577-0092-0

© Санкт-Петербургский государственный
университет информационных технологий,
механики и оптики, 2007

Оглавление

1. Введение	5
2. HTML-редактор Adobe Dreamweaver.....	9
2.1 Идеология и рабочее пространство программы.....	9
2.2 Подготовка редактора Dreamweaver к работе	11
3. Управление сайтом средствами редактора Dreamweaver.....	15
3.1 Подготовка редактора к работе с web-сервером	15
3.2 Работа с локальной и удаленной версией сайта.....	17
3.3 Работа с картой сайта.....	21
4. Реализация структуры сайта	23
4.1 Соглашение об именах файлов	23
4.2 Создание заглавной страницы.....	23
4.3 Создание навигационной панели.....	28
4.4 Подготовка шаблонов	29
4.5 Подготовка библиотеки элементов	34
5. Форматирование HTML-страниц: логическое и визуальное	36
5.1 Элементы структурной разметки.....	36
5.2 Визуальное форматирование текста.....	38
5.3 Создание гипертекстовых ссылок	39
5.4 Создание списков	41
5.5 Создание и форматирование таблиц	42
5.6 Создание и использование стилей CSS.....	46
5.7 Импорт текста в формат HTML	51
6. Вставка изображений и других элементов.....	54
6.1 Использование графических материалов.....	54
6.2 Создание карты ссылок на изображениях	55
6.3 Вставка изменяющегося изображения	56
6.4 Определение интерактивных свойств объектов.....	57
6.5 Вставка Flash-кнопок	61
6.6 Использование подключаемых модулей.....	63
7. Создание тестовых заданий с использованием дополнения CourseBuilder	65
7.1 Вставка элементов CourseBuilder	65
7.2 Создание тестового задания в закрытой форме	67

7.3	Создание тестового задания в открытой форме.....	72
7.4	Создание тестового задания на установление соответствия	75
7.5	Создание тестового задания на установление правильной последовательности	79
7.6	Создание последовательности тестовых заданий.....	80
7.7	Организация подсчета и вывода результатов.....	82
Приложение А. Краткая справка по тегам языка HTML		86
	Структура документа HTML.....	86
	Теги языка HTML.....	86
Приложение Б. Задания для закрепления материала.....		102
Литература.....		109

1.

Введение

Сегодня во все области человеческой деятельности вмешивается *компьютерная сеть Internet*. В большей или меньшей степени сеть входит в жизнь каждого: в повседневную жизнь (почта и новости, отдых и развлечения), а главное – в профессиональную деятельность. Наиболее распространенной на данный момент службой Internet является WWW (от англ. *World Wide Web* – Всемирная Паутина), которая предназначена для публикации информации на всеобщее обозрение.

В основу всемирной паутины положена идеология и технология гипертекста, технически реализованная при помощи языка HTML. HTML (от англ. *HyperText Markup Language*, язык гипертекстовой разметки) представляет собой язык, разработанный специально для создания web-документов. HTML-документы представляют собой обычные текстовые ASCII-файлы. Помимо текстового содержания, они включают в свой состав специальные управляющие конструкции языка – так называемые *теги*. При отображении документа в браузере сами теги на экран не выводятся. Они определяют структурные единицы внутри документа, задают параметры форматирования, включают в состав документа новые элементы (встроенные изображения, Java-апплеты и т.п.) и, что самое важное, позволяют создавать гипертекстовые ссылки на другие документы и ресурсы сети Internet.

Стандарт языка HTML, как и все остальные стандарты для Web, был разработан под руководством World Wide Web консорциума (W3C). Последние версии этого и других стандартов, а также проекты новых предложений можно найти на сайте консорциума по адресу <http://www.W3.org>. Следует заметить, что на практике поддержка различными браузерами официального стандарта языка оставляет желать лучшего. Это выражается с одной стороны в неполной или некорректной поддержке отдельных тегов и во введении своих "фирменных" тегов с другой. Такое нестрогое следование стандарту со стороны производителей браузеров значительно усложняет процесс создания качественных web-страниц.

Для создания и редактирования HTML-документов достаточно любого текстового редактора с минимальными возможностями. По мнению сторонников чистоты языка, ручное написание HTML-кода в простом текстовом редакторе является единственно правильным. Однако, в большинстве практических случаев написание исходного кода "вручную" не используется. Это связано с тем, что подобный подход требует от разработчика наличия глубоких знаний языка HTML и практических навыков разработки сайтов. Если говорить о профессиональных web-разработчиках, то они используют редактирование вручную, только в редких и сложных случаях. Ручное редактирование приводит к значительному увеличению временных затрат.

На практике обычно используют средства редактирования, разработанные специально для написания HTML. Подобные программные продукты позволяют повысить эффективность и значительно сократить время разработки, особенно если речь идет о средних и крупных сайтах. Специализированные

HTML-редакторы также работают с исходным кодом web-страницы, но располагают различными вспомогательными инструментами, ускоряющими и упрощающими процесс редактирования документов и дополнительными средствами для выполнения повторяющихся операций. Кроме того, наиболее мощные из них, включают в свой состав средства автоматизации процесса разработки (работа с шаблонами, стилями и библиотеками элементов) и средства размещения файлов сайта на web-сервере. Но при всех удобствах подобных средств разработки, они требуют от автора достаточно глубоких знаний языка HTML. К данной группе редакторов относится большое количество программных продуктов, в том числе и распространяемых бесплатно.

Второй большой класс инструментов для web-разработчиков составляют HTML-редакторы WYSIWYG (от англ. *what you see is what you get*, что видишь, то и получаешь) или, как их еще называют, средства визуального редактирования. Такие программные продукты имеют графические интерфейсы, делающие написание HTML-кода подобным работе с программой редактирования текстов и разметки страниц. Изначально основной целью подобных инструментов было освобождение автора от необходимости непосредственной работы с тегами HTML. Современные визуальные HTML-редакторы имеют средства, позволяющие избавить разработчика от выполнения рутинных действий. При всех своих преимуществах подобным инструментам свойственен существенный недостаток – они не создают чистого HTML-кода, добавляя в него избыточные, а иногда и фирменные теги. В некоторых случаях, для достижения нужного эффекта они используют излишне сложные методы, а отдельные из них могут сгенерировать даже неверный код. Подобные проблемы решаются выбором инструмента, генерирующего максимально чистый и правильный код. Но даже в этом случае нельзя рассчитывать на то, что для создания действительно качественных страниц удастся избежать ручной доводки, а, следовательно, хотя бы общего знакомства с языком HTML. Следует заметить, что серьезные, мощные пакеты подобного класса являются коммерческим программным обеспечением, имеющим достаточно высокую стоимость.

На основе сделанного выше обзора инструментов редактирования HTML можно сделать вывод, что для разработчиков, не имеющих большого опыта создания web-страниц и не располагающих глубокими знаниями языка HTML, наилучшим выбором является один из WYSIWYG-редакторов. Авторов образовательных ресурсов в большинстве случаев можно отнести именно к этой группе. Использование визуальных редакторов позволит им без лишних временных затрат на глубокое изучение HTML и web-дизайна создавать качественные образовательные Internet-ресурсы.

Осталось лишь остановить свой выбор на одном из конкретных программных продуктов данной категории. Критерии выбора, и соответственно его результат, могут быть различными. В данном учебном пособии предлагается использовать редактор Dreamweaver фирмы Adobe. Этот редактор в настоящее время рассматривается как промышленный стандарт в области

web-мастеринга. Основным критерием такого выбора стало то, что редактор генерирует достаточно чистый код, соответствующий стандарту языка HTML. Dreamweaver представляет собой полноценную среду разработки как отдельных web-страниц, так и сайтов любого масштаба. Ядром пакета является мощный HTML-редактор, позволяющий осуществлять как разработку документа в визуальном режиме, так и его ручное кодирование. Dreamweaver располагает мощными средствами автоматизации разработки и управления web-сайтом. В его состав входит множество полезных вспомогательных инструментов, упрощающих и ускоряющих разработку. Кроме того, Dreamweaver чрезвычайно гибок в настройке, что позволяет организовать индивидуальную среду, удобную для конкретного разработчика. В данном пособии рассматриваются вопросы создания web-страниц, разработки и управления сайтами с использованием именно этого редактора.

Глава 2 посвящена основам работы с редактором Adobe Dreamweaver. Особое внимание в этой главе уделяется подготовке редактора к работе с русскоязычными документами.

Вопросам проектирования и организации сайта посвящена глава 3. Кроме богатых возможностей по редактированию HTML-страниц редактор Dreamweaver предоставляет инструменты для управления сайтом, о настройке и использовании которых рассказывается в этой главе.

Вопросы реализации разработанной структуры сайта – правила назначения имен файлов, создание страниц, работа с фреймами, разработка навигационной панели – рассматриваются в главе 4. В этой главе также уделяется большое внимание средствам автоматизации, которые можно использовать при создании и поддержании сайта, – шаблонам и библиотеке элементов.

Логическое и визуальное форматирование HTML-документа рассматривается в главе 5. Рассказывается об особом оформлении информации в виде списков и таблиц. Большое внимание уделяется работе с каскадными таблицами стилей CSS. В конце главы рассматриваются вопросы преобразования текста в формат HTML из различных форматов (в частности из формата широко распространенного редактора Microsoft Word).

Глава 6 посвящена использованию на страницах графических материалов, а также Internet-технологиям, которые позволяют организовать интерактивное взаимодействие пользователя с ресурсами. HTML-редактор Dreamweaver имеет специальные средства для управления поведением любых объектов, расположенных на HTML-странице. О них, а также о готовых JavaScript-программах, которые можно использовать при создании своих ресурсов, рассказывается в этой главе.

Для создания тестовых заданий на web-страницах фирма Adobe предлагает специальный программный продукт CourseBuilder for Dreamweaver. В главе 7 рассматриваются вопросы связанные с разработкой тестовых заданий в различной форме и их реализацией с помощью программы CourseBuilder.

Данное учебное пособие не предусматривает изучение идеологии и синтаксиса языка HTML. В приложении приводится краткая справка по тегам языка HTML. Кроме того, вопросам теории языка, непосредственно тегам и их

атрибутам, а также поддержке тегов со стороны различных браузеров посвящено большое количество литературных источников, некоторые из которых приведены в списке литературы, и разнообразных ресурсов сети Internet.

2. HTML-редактор Adobe Dreamweaver

Создатели программы Dreamweaver постарались сделать удобный и простой в использовании инструмент, облегчающий процесс создания HTML-страниц, их редактирования и управления большим количеством связанных друг с другом документов. В данном учебном пособии рассматриваются основы работы с пакетом Adobe Dreamweaver CS3. К сожалению, ввиду ограниченности своего объема, пособие не охватывает всех возможностей и настроек программы. Основное внимание уделяется инструментам и приемам работы, позволяющим создавать качественные образовательные ресурсы. Вы научитесь создавать и форматировать web-страницы, использовать интерактивные элементы, познакомитесь со средствами автоматизации разработки и управления файлами сайта.

2.1 Идеология и рабочее пространство программы

Внешний вид программы является стандартным для операционной системы Windows. После запуска Dreamweaver открывается окно с возможностью выбора дальнейших действий пользователя.

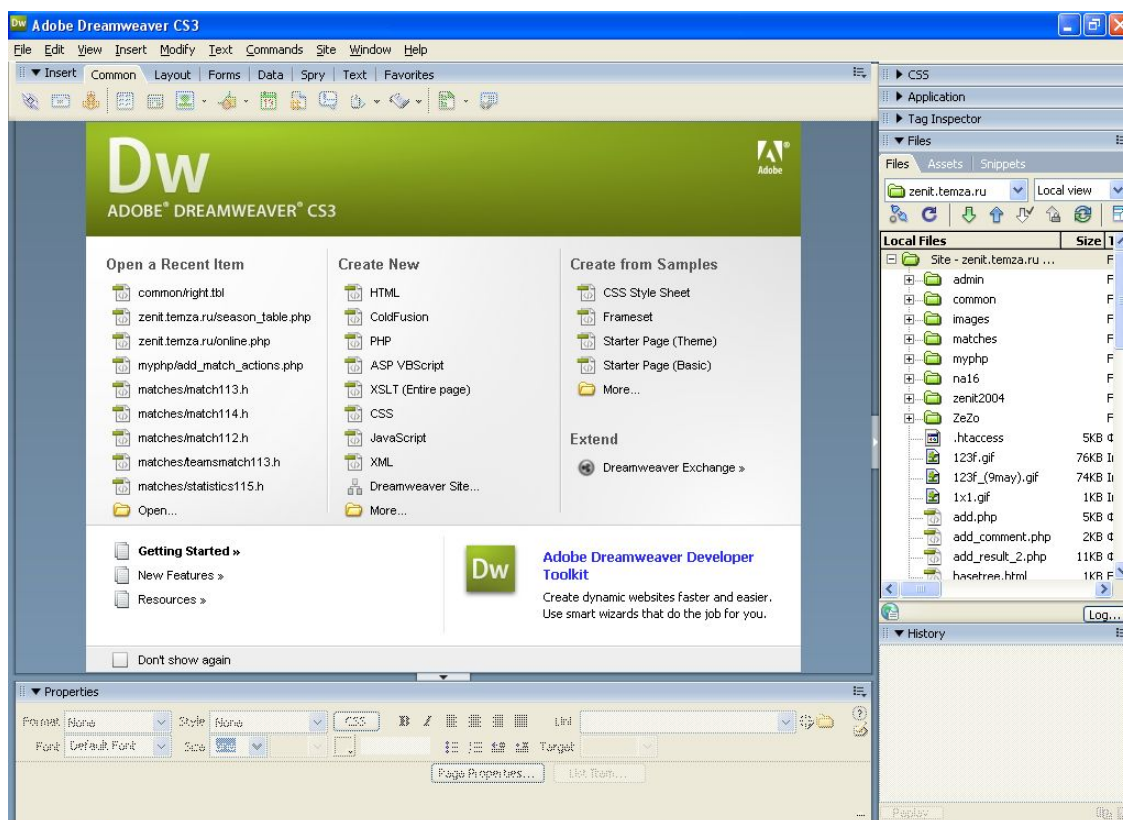


Рис. 2.1. Главное окно программы Dreamweaver

Open a Recent Item – открыть документ, выбрав его из списка ранее созданных (размер списка ограничен).

Create New – создать новый документ (HTML, PHP, CSS, JavaScript и т.д.)

Create from Samples – создать новый документ на основе готовых примеров страниц.

пустым HTML-документом. Каждый новый HTML-документ открывается в новой закладке. Для более удобной работы с программой предусмотрены удобная система меню и набор палитр инструментов.

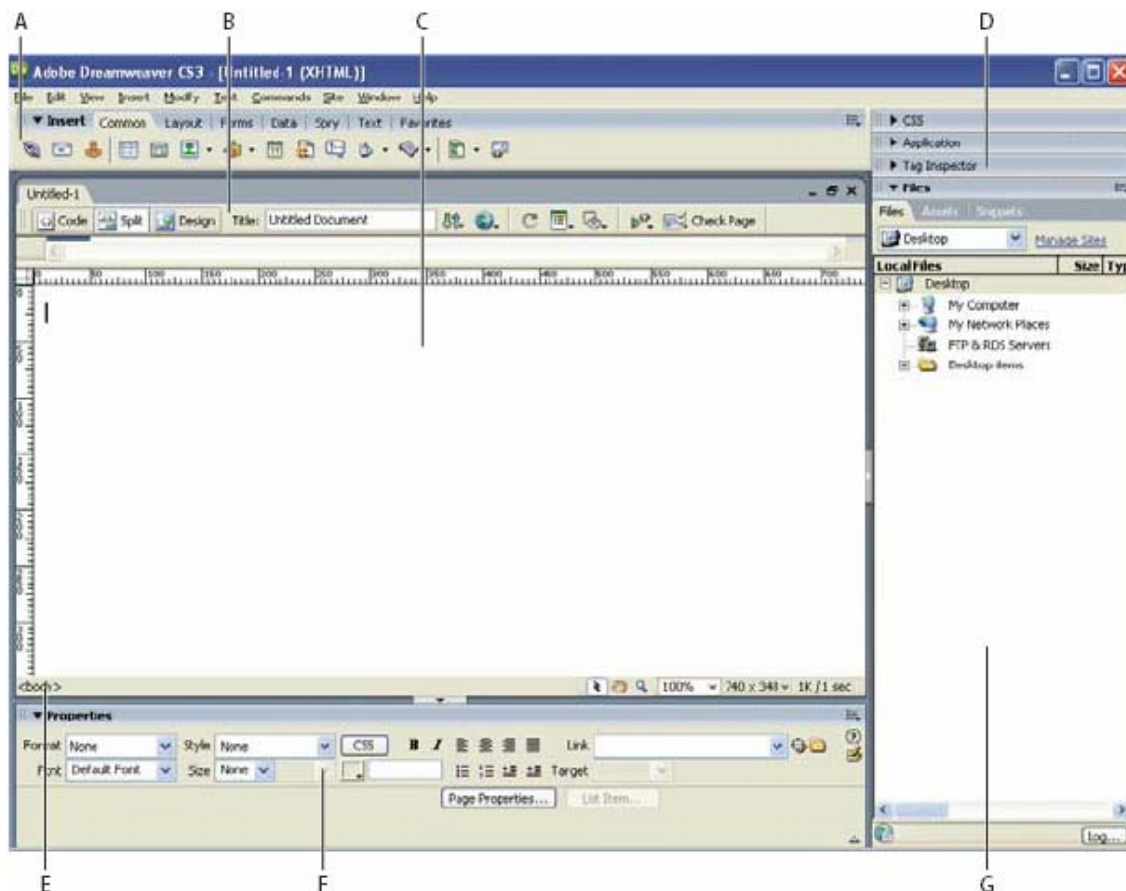


Рис. 2.2. Рабочее пространство Dreamweaver

(А. Панель объектов, В. Панель инструментов, С. Поле документа, D. Навигатор панелей, Е. Выбор HTML-тегов, F. Панель свойств объекта, G. Менеджер сайта)

На рисунке 2.1 представлен вид программы после запуска. Не удивляйтесь, если у вас она будет выглядеть немного по-другому – внешний вид основного окна и состояние дополнительных панелей может зависеть от различных настроек. На рисунке главное окно программы разделено на две части: область исходного кода документа и поле его визуального представления. Область, в которой отображается исходный код, может быть скрыта, при этом поле документа будет занимать все окно. Может быть обратная ситуация – поле визуального представления документа отключено, а включено только отображение исходного кода. Вы можете настраивать рабочее пространство программы таким образом, чтобы ваша работа с ней была максимально удобной и эффективной.

Программа Dreamweaver, являясь визуальным HTML-редактором, также позволяет переключаться в режим ручного редактирования кода, осуществляя взаимосвязь между визуальным представлением страницы и ее исходным кодом. Для переключения между режимами отображения и редактирования HTML-документа предназначены три первые кнопки    на панели инструментов.

2.1. Идеология и рабочее пространство программы

Code – исходный код;

Split – исходный код / поле документа (окно делится на две части);

Design – поле документа.

Все основные действия при создании и редактировании HTML-страницы производятся в поле документа. Как известно, кроме текстовой информации HTML-документ может содержать различные графические и мультимедийные элементы. Для удобства вставки таких элементов предназначена панель объектов **Insert** (рис. 2.1). С помощью кнопок, расположенных на этой панели, можно быстро вставить различные объекты (изображения, таблицы, Flash-анимация) в редактируемый документ.

Любой HTML-документ можно представить как совокупность таких элементов как слово, абзац, изображение, ячейка таблицы и других. Все они имеют некоторые общие свойства (например, такие как выравнивание), но в то же время, каждый объект обладает присущими только ему свойствами, параметрами и настройками. Для быстрого доступа к редактированию параметров объекта предназначена панель свойств **Properties** (рис. 2.1). В зависимости от текущего выделенного объекта она изменяет свой внешний вид – предлагается просмотреть и при необходимости изменить набор свойств именно этого объекта.

При работе со сложными страницами полезным инструментом является панель **History** (рис. 2.1). В список на этой панели заносятся все действия, выполняемые в ходе редактирования, начиная с момента открытия файла (по умолчанию сохраняется 50 последних операций). Используя панель **History**, можно отменить, повторить, сохранить выделенный диапазон операций.

Как и в большинстве программ Windows, все основные команды и операции можно выполнить различными способами – через панели инструментов, пункты меню или по комбинации клавиш. Не исключение и описываемые здесь действия, поэтому не нужно слепо ими руководствоваться. Следует выбирать такой способ выполнения действий, который является для вас наиболее удобным и сделает вашу работу быстрой и эффективной.

2.2 Подготовка редактора Dreamweaver к работе

Для того, чтобы сделать работу с редактором эффективной и комфортной, нужно определить значение некоторых параметров. Чтобы вызвать настройку параметров программы, нужно в главном меню открыть пункт **Edit**, а в нем выбрать пункт **Preferences...** (**Edit=>Preferences...**)

После этого на экране отображается окно редактирования параметров работы (рис. 2.2). В левой части окна отображаются категории возможных настроек, а в правой – параметры выбранной категории.

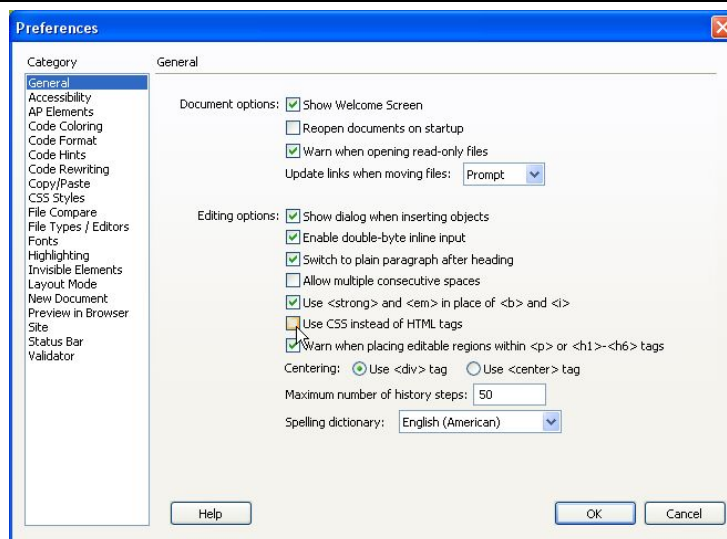


Рис. 2.2. Диалоговое окно редактирования параметров работы редактора Dreamweaver

В категории **General** собраны общие настройки программы. В категориях **Code Colors**, **Code Format**, **Code Rewriting** задаются параметры, определяющие отображение и редактирование непосредственно HTML-кода. Параметры работы с таблицами стилей задаются в категории **CSS Styles**. В категории **File Types / Editors** приводятся в соответствие типы файлов и их внешние редакторы.

Флажок **Use CSS instead of HTML tags** включает или отключает использование для форматирования текста стилей CSS вместо тегов HTML. По умолчанию он включен, и его желательно выключить.

Важным моментом, на котором следует остановиться подробнее, является подготовка редактора Dreamweaver к работе с русскоязычными документами.

Общепринятым, стандартным языком в компьютерных технологиях является английский язык. Поэтому при размещении англоязычных документов в Internet не возникает никаких затруднений. Но как только речь заходит о документах, которые используют другую кодировку – возникает целый ряд проблем.

Каждому символу присвоен определенный номер – его код, такая взаимосвязь носит название *таблица кодировки*. Первая часть всех таблиц кодировок совпадает – там содержатся коды цифр, букв латинского алфавита, знаков пунктуации. А вот дальше в таблице кодировки идут коды русских букв и некоторых специальных символов. Шрифты строятся на основе какой-либо таблицы кодировки и определяют вид отрисовываемого символа, соответствующего конкретному коду. Поэтому для корректного отображения HTML-документов необходимо указывать кодировку, в которой он сохранен.

При работе в редакторе Dreamweaver необходимо указать, в какой кодировке предполагается редактировать документы. Для того, чтобы определить кодировку по умолчанию для вновь создаваемых HTML-документов, нужно в окне **Preferences** (рис. 2.3) выбрать категорию **New Document**.

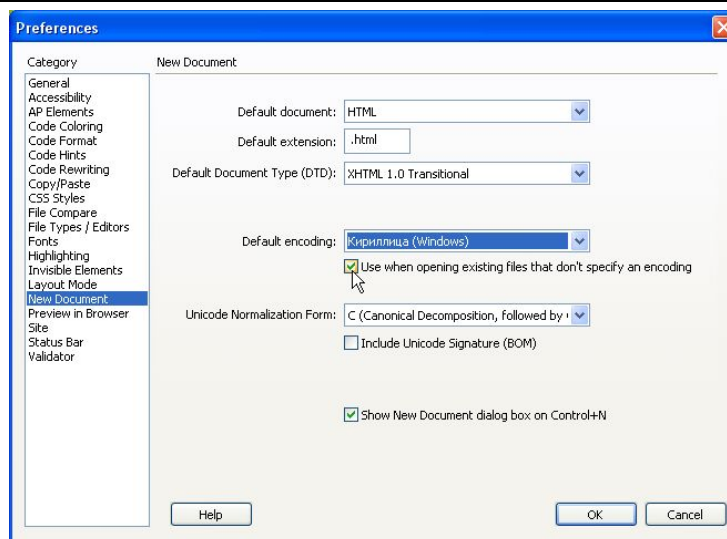


Рис. 2.3. Диалоговое окно редактирования параметров работы редактора Dreamweaver

В списке **Default encoding** указывается какая-либо из кириллических кодировок. Наиболее употребительными являются *Кириллица (KOI8-R)* и *Кириллица (Windows)*. Какую из них выбрать зависит от настроек сервера, на котором планируется размещать HTML-документы. Рекомендуется, если это возможно, использовать кодировку *Кириллица (Windows)*. Теперь нужно включить флажок **Use when opening existing files that don't specify an encoding**, если он не включен по умолчанию. После этого Dreamweaver будет автоматически применять выбранную в списке **Default encoding** кодировку ко всем открываемым Web-страницам, если в них не задана явно другая кодировка. Далее, в списке **Fonts Setting** категории **Fonts** нужно указать, какие шрифты будут использоваться для отображения текста.

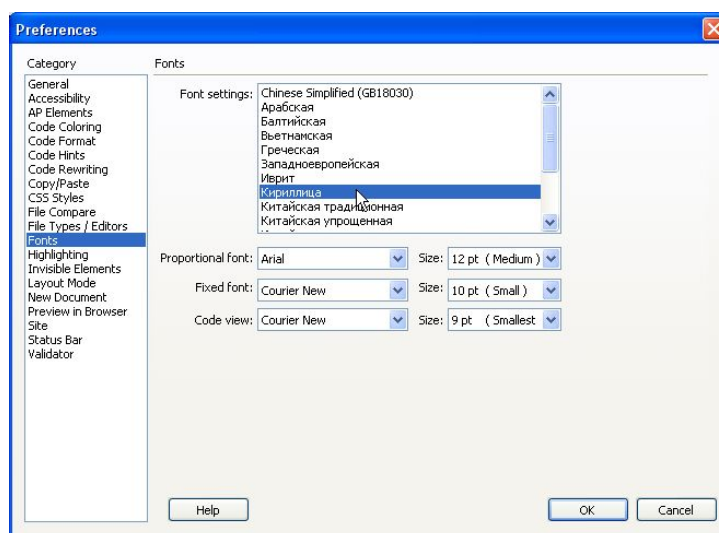


Рис. 2.3. Диалоговое окно редактирования параметров работы редактора Dreamweaver

Для русскоязычных HTML-документов нужно выбрать пункт *Кириллица*. Теперь все вновь создаваемые HTML-документы будут поддерживать русские буквы.

Для того чтобы изменить кодировку уже существующего HTML-документа необходимо выполнить команду **Modify=>Page Properties**. Появится окно параметров текущей страницы (рис. 2.4).

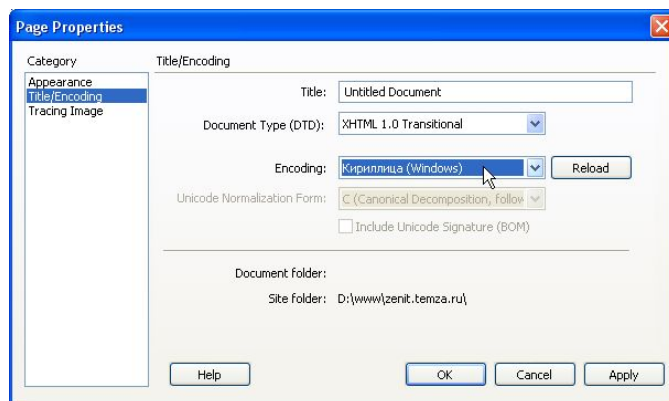


Рис. 2.4. Диалоговое окно настройки параметров страницы (категория *Title/Encoding*)

В категории **Title/Encoding** из списка **Encoding** следует выбрать новую желаемую кодировку и нажать кнопку **Reload**, расположенную справа. Если перед этим документ не был сохранен, то все изменения, сделанные после последнего сохранения, будут утеряны. Чтобы этого избежать, перед изменением кодировки необходимо сохранить документ. Преобразование кодировок возможно только между кодировками одного типа. Например, изменение кодировки *Кириллица (KOI8-R)* в *Кириллица (Windows)* будет выполнено корректно, а преобразование из *Западноевропейской* в *Кириллицу (KOI8-R)* будет некорректным.

На диалоговом окне **Preferences** (рис. 2.2) в списке **Category** осталось еще несколько групп параметров, которые требуют пояснения. В категориях **Highlighting** и **Invisible Elements** собраны параметры, отвечающие за подсвечивание цветом служебных областей и важных, но невидимых тегов на поле редактирования HTML-документов.

Параметры работы со слоями собраны в категории **AP Elements**, а параметры работы в режиме **Layout** – в категории **Layout Mode**.

Настроить режим быстрого просмотра HTML-документов в браузере можно с помощью элементов управления расположенных в категории **Preview in Browser**.

Параметры работы с инструментами управления сайтом можно настроить в категории **Site**. Изменить внешний вид строки подсказки в окне редактора можно с помощью параметров, собранных в категории **Status Bar**.

Настройку параметров работы редактора Dreamweaver достаточно выполнить один раз после установки программы. А далее при редактировании HTML-документов во всех сеансах работы с программой она будет работать согласно установленным параметрам.

3. Управление сайтом средствами редактора Dreamweaver

3.1 Подготовка редактора к работе с web-сервером

HTML-редактор Dreamweaver предоставляет все необходимые инструменты для управления сайтами любого размера. Это отличает его от большинства простых HTML-редакторов и делает полноценным средством создания и управления сайтами. Рассмотрим использование этих инструментов подробнее.

Работа со средствами управления сайтом в Dreamweaver начинается с выполнения команды **Site=>Manage Sites...** На экране появится окно со списком сайтов (рис. 3.1).



Рис. 3.1. Диалоговое окно для создания и управления сайтами

Для создания нового сайта следует нажать на кнопку **New...** и в открывшемся списке выбрать **Site**. На экране появится диалоговое окно (рис. 3.2), в котором нужно указать сведения о сайте. Для редактирования свойств уже созданного сайта нужно выбрать его в списке и нажать кнопку **Edit...** По кнопке **Duplicate** будет создана копия выбранного в списке сайта, а по кнопке **Remove** сайт будет удален.

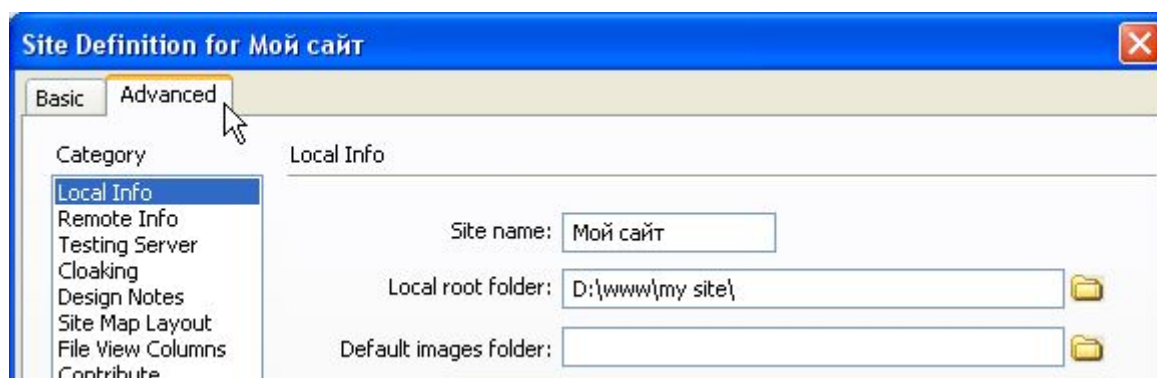


Рис. 3.2. Диалоговое окно настройки параметров сайта

Обратите внимание, что у окна настройки параметров сайта (рис. 3.2) есть две закладки **Basic** и **Advanced**. При желании можно воспользоваться мастером настройки параметров сайта – на закладке **Basic**, но лучше переключиться на закладку **Advanced** – она предоставляет больше возможностей по настройке нашего сайта.

Параметры настройки разбиты на несколько категорий, отображаемых в левой части окна (рис. 3.2). В первой категории **Local Info** задаются служебное имя сайта (**Site name**), папка, в которой хранится содержимое сайта на локальной машине (**Local root folder**), и некоторые другие свойства. Можно также указать папку, в которой по умолчанию будут храниться изображения, которые необходимы для размещения на страницах сайта (**Default images folder**). Если это поле оставить незаполненным, то изображения будут располагаться в корневой папке сайта.

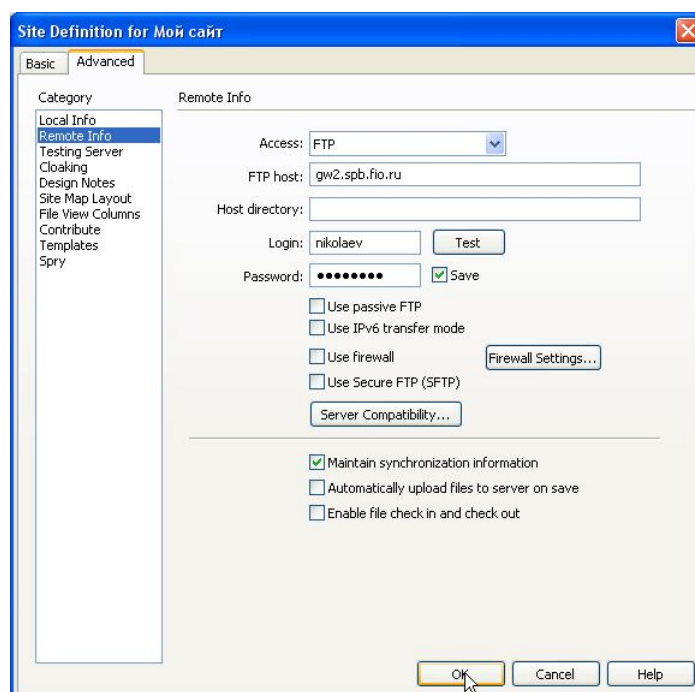


Рис. 3.3. Диалоговое окно настройки параметров соединения с сервером

Во втором разделе **Remote Info** (рис. 3.3) необходимо задать тип доступа к серверу для размещения сайта в Internet. В зависимости от выбранного типа доступа (поле **Access**) потребуется задать различные дополнительные параметры. Рассмотрим наиболее универсальный и часто используемый способ – соединение по протоколу FTP. Первое, что необходимо указать, – это адрес FTP сервера, на котором будут размещаться файлы (поле **FTP host**). В поле **Host directory** указывается путь к папке, в которой будут размещаться файлы (если она совпадает с домашним каталогом, то это поле должно остаться пустым). Далее в полях **Login** и **Password** необходимо указать имя пользователя и пароль, которые будут использоваться для соединения с FTP-сервером. Обратите внимание на флажок **Save**, справа от поля **Password**. Если этот флажок установлен, то пароль доступа к FTP-серверу автоматически

запоминается. Если флажок убрать, то пароль необходимо будет вводить при каждом подключении к серверу.

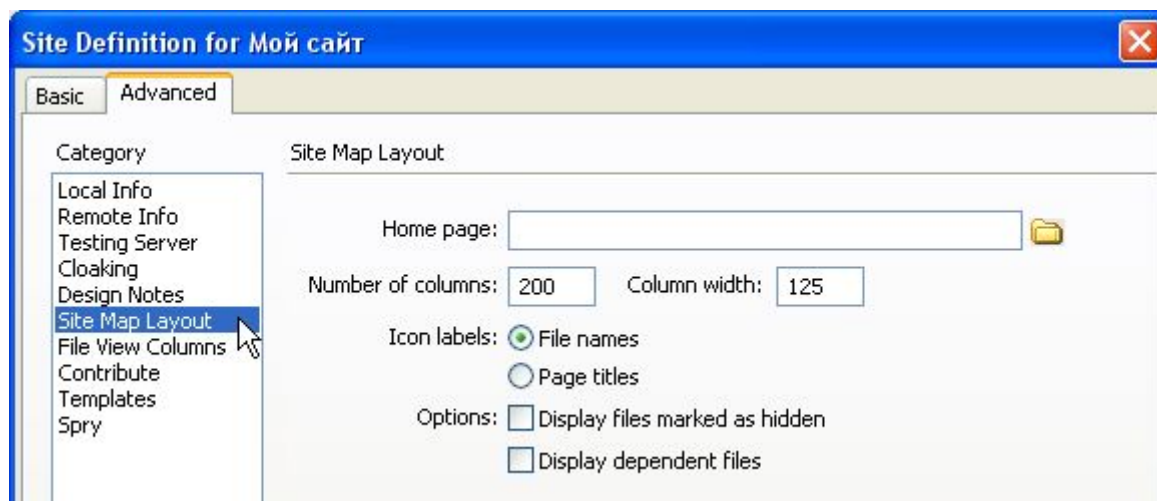


Рис. 3.4. Диалоговое окно настройки параметров карты сайта

В разделе **Site Map Layout** (рис. 3.4) можно определить первую страницу сайта (поле **Home Page**). Без этого не обойтись при построении карты сайта.

Когда все параметры заданы, можно нажать кнопку **ОК** и вернуться к окну **Define Sites**. После нажатия в нем кнопки **Done** можем приступить к работе над будущим сайтом.

3.2 Работа с локальной и удаленной версией сайта

Для работы с локальной и удаленной версией сайта в программе Dreamweaver есть диалоговое окно **Site Manager (Менеджер сайта)**. Половина этого окна находится в правой части рабочей области Dreamweaver, в панели **Files** (рис. 3.5).

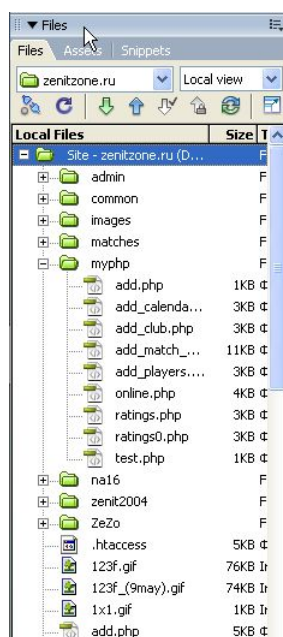


Рис. 3.5. Панель Files

Для того, чтобы полностью раскрыть окно **Site Manager**, необходимо нажать на кнопку **Expand to show local and remote sites**  в правом верхнем углу панели **Files**.

Рассмотрим основные элементы панели инструментов **Site Manager** (рис. 3.6). При нажатии на кнопку **Site Files**  включается режим просмотра файлов сайта, находящихся на сервере. Режим просмотра карты сайта включается по кнопке **Site Map** . Если работа ведется одновременно над несколькими сайтами, то переключаться между ними можно с помощью списка **Site**, расположенного на панели инструментов. Кнопка **Connects to remote host**  позволяет установить или, при необходимости, разорвать соединение с сервером. По кнопке **Refresh**  осуществляется обновление информации о файлах в локальной и удаленной версии сайта. Разместить файлы на сервере можно с использованием инструмента **Put File(s)** , а получить файлы с сервера – с использованием инструмента **Get File(s)** .

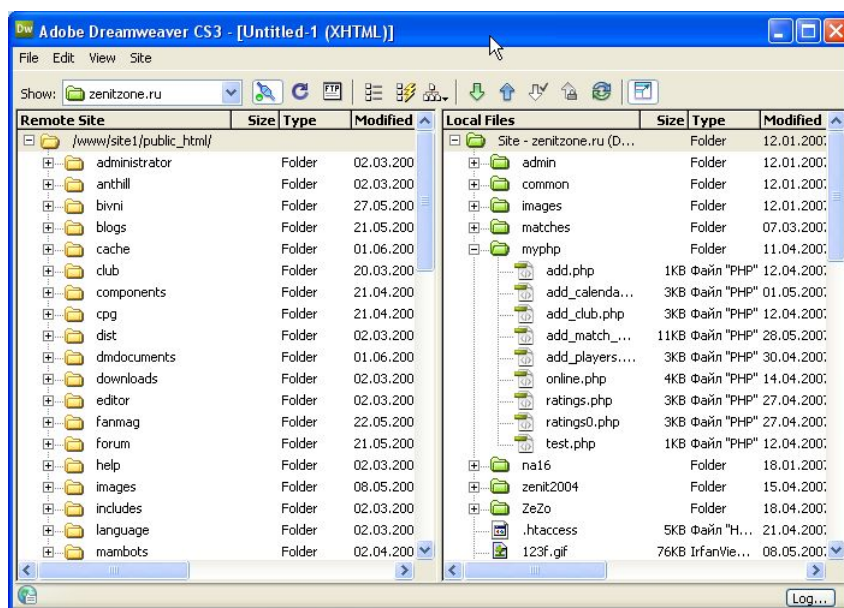


Рис. 3.6. Диалоговое окно **Site Manager**

Site Manager позволяет выполнять все основные операции по управлению файлами: создание, удаление, переименование и другие. Все операции с файлами сайта рекомендуется производить в **Site Manager**, так как при изменении имен и расположения файлов ссылки на них из других документов будут автоматически отслеживаться и обновляться. Для того, чтобы начать редактирование файла локальной копии сайта, достаточно двойного щелчка левой кнопкой мыши на нем. Редактировать можно только файлы локальной копии сайта. Вносить изменения в страницы, размещенные на сервере, можно предварительно сделав локальную копию.

В рамках сайта можно производить поиск и замену. Для этого справа необходимо вернуться к главному окну программы Dreamweaver, нажав на

кнопку **Collapse to show only local or remote sites**  и выполнить команду **Edit=>Find and Replace**. В открывшемся диалоговом окне (рис. 3.7) нужно указать параметры поиска и, если необходимо, замены. Поле **Find In** определяет, в каких файлах будет производиться поиск (Current Document – текущий выделенный файл, Entire Current Local Site – поиск по всем файлам локальной копии сайта, Selected Files in Site – поиск в выбранных файлах сайта, Folder... – поиск в какой-либо папке). В поле **Search** необходимо указать, в какой части HTML-документов следует производить поиск (Source Code – в исходном коде HTML, Text – непосредственно в тестовом наполнении документов, Text (Advanced) – позволяет производить поиск одновременно в текстовом наполнении и указанных тегах документов HTML, Specific Tag – производить поиск только в указанных тегах HTML). В поле **Find** вводится слово или словосочетание, которое нужно искать. В поле **Replace** указывается словосочетание, на которое нужно заменять найденные фрагменты.

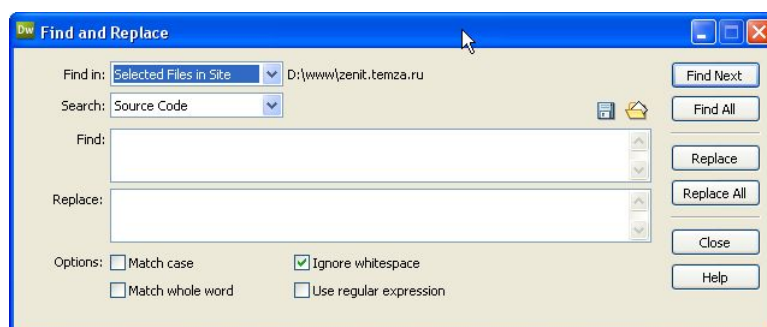


Рис. 3.7. Диалоговое окно поиска и замены

При нажатии на кнопку **Find Next** будет открыт первый файл, в котором встретилось искомое словосочетание, и первое его вхождение в этот документ будет выделено. При повторном нажатии на эту кнопку будет осуществлен дальнейший поиск ключевой фразы в открытом, а затем и в остальных заданных документах. При нажатии на кнопку **Find All** будет автоматически осуществлен поиск во всех указанных файлах, а его результаты будут выведены в список в нижней части окна поиска. Двойной щелчок по элементу списка приведет к тому, что будет открыт соответствующий файл и в нем будет выделен найденный фрагмент. Инструменты **Replace** и **Replace All** действуют аналогичным образом, за исключением того, что будут сделаны соответствующие замены найденных фрагментов.

Одним из полезных практических приемов использования инструмента поиска и замены является возможность быстрого конвертирования всех HTML-документы сайта из одной кодировки в другую (например, из windows-1251 в KOI8-R). Для этого необходимо, чтобы конвертируемые документы содержали в своем заголовке мета-тег с указанием кодировки (все документы, созданные в Dreamweaver содержат его по умолчанию). В окне параметров поиска нужно указать, что он будет выполняться в исходном коде HTML (Source Code) во всех файлах локальной копии сайта (Entire Local Site). Строка “windows-1251” должна быть заменена строкой “KOI8-R”. Поскольку изменяется тип кодировки

документов, Dreamweaver автоматически произведет конвертирование всех страниц сайта.

Локальная копия сайта фактически ничем не отличается от копии сайта на сервере. На практике локальная копия может содержать не только материал, размещаемый в Internet, но и “черновики” документов, и другие рабочие файлы. Не существует жестких требований к организации файловой структуры сайта. Все определяется требованиями целесообразности и удобства представления и редактирования материалов. При организации файлов сайта рекомендуется следовать следующим общепринятым принципам:

- в корневом каталоге сайта обязательно должен быть файл index.html, который является первой (стартовой) страницей сайта или фреймосодержащим документом;
- графические файлы размещаются в отдельных папках (обычно images);
- если имеются дополнительные файлы, используемые как компоненты страниц (например, скрипты JavaScript, стили CSS, анимации Flash), их размещают в отдельных папках в соответствии с типом;
- если сайт содержит несколько тематических разделов, то каждый раздел размещается в своей папке, куда также входят все дополнительные материалы, используемые в этом разделе;
- для документов, входящих в состав сайта, как правило, используют информативные названия, например links.html, page1.html, background.gif и т.д. (а не имена по умолчанию, такие как Untitled1.htm или Movie1.swf).

На сервере желательно размещать только те файлы, которые непосредственно составляют содержимое сайта, то есть документы HTML и все связанные с ними дополнительные файлы (изображения, Flash-анимации и другие).

Для того, чтобы разместить файл или группу файлов на сервере, следует выделить их и выполнить команду **Site=>Put**, которая также вызывается по кнопке  на панели инструментов. Если выделены HTML-документы, на экране появится окно (рис. 3.8), в котором система предложит разместить на сервере вспомогательные файлы (изображения, используемые в документе, и другие объекты).

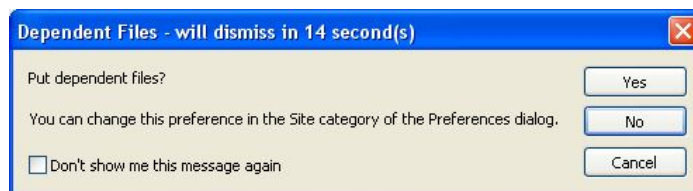


Рис. 3.8 Запрос на копирование вспомогательных файлов

Кроме ручного размещения материалов возможно выполнение синхронизации файлов локальной копии сайта и файлов, размещенных на сервере. Для этого необходимо выполнить команду **Site=>Synchronize...** После этого на экране появится окно синхронизации файлов (рис. 3.9).

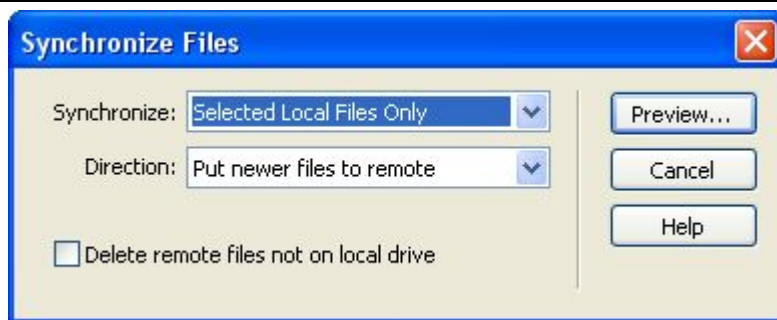


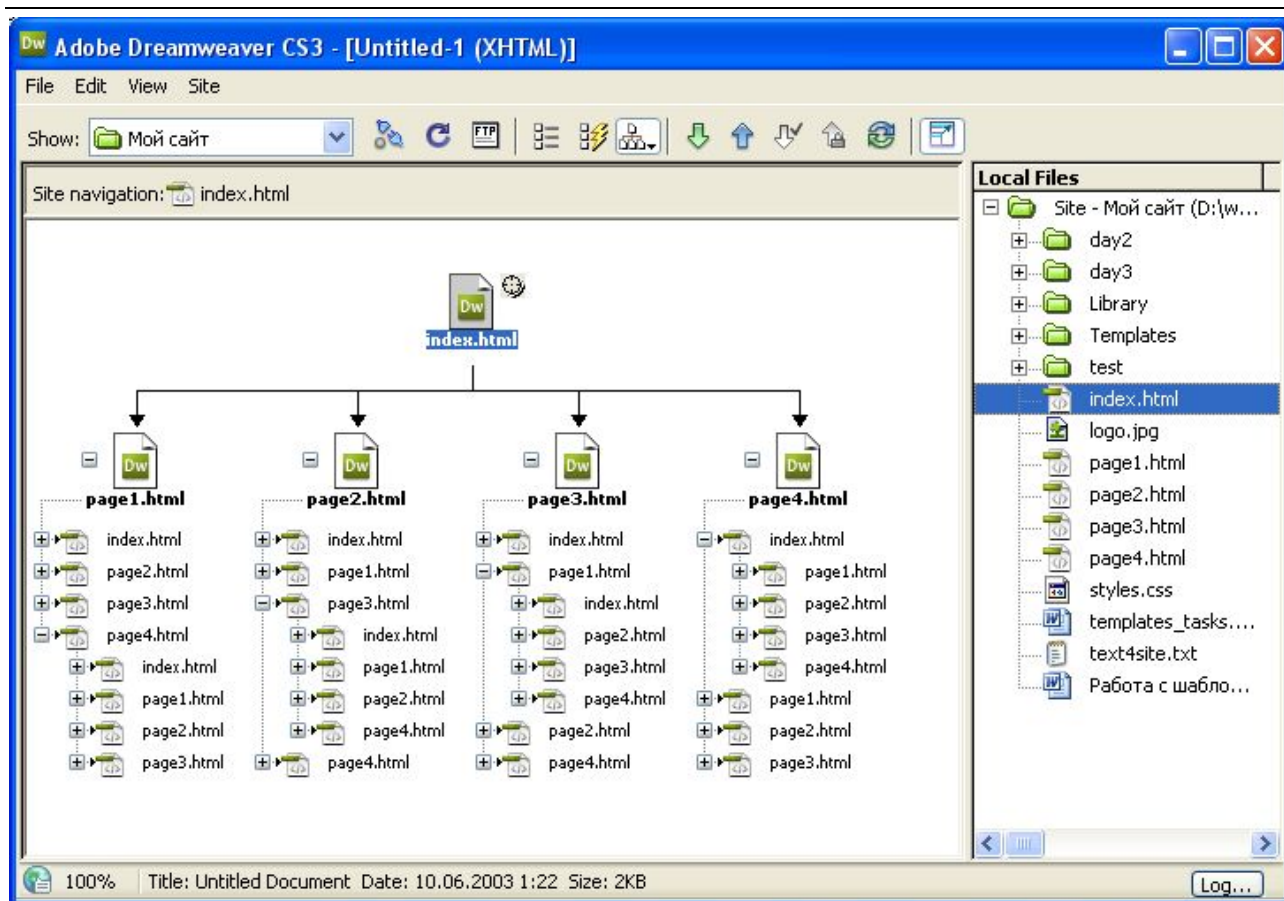
Рис. 3.9. Диалоговое окно синхронизации файлов

В списке **Synchronize** необходимо определить, какие файлы будут являться объектами синхронизации – все файлы локальной копии сайта (Entire 'имя сайта' Site) или только выделенные файлы (Selected Local Files Only). В списке **Direction** нужно определить, что требуется делать с файлами: разместить обновленные файлы на сервере (Put newer files to remote), записать в локальную копию сайта более новые версии файлов с сервера (Get newer files from remote), записать обновленные файлы локальной копии на сервер и получить в локальную копию сайта новые файлы с сервера (Get and Put newer files). Для первых двух вариантов действий с файлами можно включить опцию **Delete remote files not on local drive**, при этом с сервера будут удалены все файлы, которые отсутствуют в локальной копии сайта. В результате локальная и удаленная копии сайта должны совпадать. При выполнении операции синхронизации следует учитывать то, что время на локальной машине и на сервере может отличаться – это может привести к неверному определению местонахождения последних, обновленных версий файлов.

Site Manager позволяет также организовать ведение работы над сайтом нескольких разработчиков с разных компьютеров. Средства синхронизации помогут разрешить конфликты, которые возникают при редактировании одного и того же файла разными разработчиками.

3.3 Работа с картой сайта

Для перехода к режиму работы с картой сайта необходимо нажать кнопку **Site Map** в окне **Site Manager** (рис. 3.6). При этом в левой части этого окна будет отображена карта текущего сайта (рис. 3.10). Этот режим предназначен для представления локальной копии сайта в виде визуальной карты, на которой наглядно отображаются страницы в виде иконок, соединенных связями.



*Рис. 3.10. Диалоговое окно Site Manager
в режиме отображения карты сайта*

По умолчанию на карте отображается структура сайта на два уровня в глубину, начиная с текущей стартовой страницы. Для просмотра более глубоких уровней карты (т.е. документов, связанных с элементами второго уровня) нужно нажать на символ "+", расположенный рядом с иконками страниц (при нажатии на появляющийся при этом символ "-" соответствующая ветвь карты будет свернута). Аналогичным образом можно раскрывать и закрывать любые существующие связи.

Карта может быть использована для работы с файлами сайта (операции добавления, удаления, переименования, размещения на сервере и т.п.), но основным ее применением является контроль, создание и изменение гипертекстовых связей. Следует заметить, что в некоторых случаях невозможно построение достоверной карты сайта. Это связано с тем, что Dreamweaver не может отследить не HTML ссылки, т.е. созданные при помощи JavaScript, Flash и т.п. – подобные связи не будут отображены в карте.

4. Реализация структуры сайта

4.1 *Соглашение об именах файлов*

В процессе создания web-сайта существенным моментом является правильное назначение имен файлов. При этом разработчик должен учитывать некоторые технические ограничения и, кроме того, следовать определенным логическим правилам.

Первая проблема заключается в том, что русскоязычным авторам и пользователям давать имена своим ресурсам сложнее, так как в адресах URL невозможно использовать символы кириллицы. То же самое относится к большинству специальных и вспомогательных символов (в частности к символу пробел). Поэтому при назначении имени ресурса следует ограничиться цифрами и буквами латинского алфавита. Следующая потенциальная ошибка заключается в использовании различного регистра символов в именах файлов. Дело в том, что в ряде операционных систем (в том числе в системах семейства UNIX, под управлением которых работает большинство существующих web-серверов) в именах файлов различаются символы в верхнем и нижнем регистре. С другой стороны, разработка сайта ведется, как правило, на компьютерах под управлением системы Windows, в которой нет различия между именами файлов с разным регистром символов. Такая ситуация может привести к тому, что уже проверенные ссылки после размещения документов на web-сервере перестанут работать. Поэтому, при назначении имен файлам придерживаются какого-либо правила относительно регистра символов. Обычно это правило заключается в использовании только нижнего регистра.

При разработке Internet-ресурсов автору следует обратить внимание на адрес созданного ресурса. Исследования показывают, что пользователи полагаются на URL, когда пытаются расшифровать структуру сайта или возможный результат перехода по гиперссылке. Часто пользователи, попавшие сразу на одну из внутренних страниц сайта и еще не знакомые с его структурой, пытаются перейти к более высоким уровням, удаляя из URL последние части. Поэтому адреса ресурсов следует делать интуитивно понятными. Имена каталогов должны быть читабельными и представлять собой слово или составное слово, поясняющее смысл и содержание данного элемента сайта. Названия файлов также должны соответствовать этим требованиям и отражать их содержание. Кроме того, URL следует делать как можно более коротким, а также использовать в нем общепринятые слова естественного языка.

4.2 *Создание заглавной страницы*

После того, как структура будущего сайта продумана, и Dreamweaver настроен на работу с ним, можно приступать к разработке самих web-страниц. И начать следует с главной, или домашней, страницы сайта.

Для создания страницы необходимо открыть новый, пустой документ. Это можно сделать различными способами, например, выбрав команду **File=>New File** из главного меню инструмента **Site Manager**. Первое что необходимо

сделать после создания документа в любом случае – сохранить файл с необходимым именем (иначе при создании гипертекстовых ссылок, вставке изображений и т.п. могут быть использованы неверные адреса). Следует напомнить, что главная страница сайта должна иметь специальное имя – index.html (или index.htm). После того, как странице присвоено имя и она сохранена можно приступить к наполнению ее информационным содержанием.

Начать следует с установки параметров страницы. Для этого необходимо из главного меню выбрать команду **Modify=>Page Properties...** Основное, что необходимо задать в открывшемся диалоговом окне, это название (заголовок) страницы – поле **Title** в категории **Title/Encoding** (рис. 2.4). Считается, что каждая страница сайта должна иметь продуманное, кратко и четко сформулированное название. Визуально оно будет отображено в заголовке окна браузера и, кроме того, будет использоваться поисковыми системами и т.п.

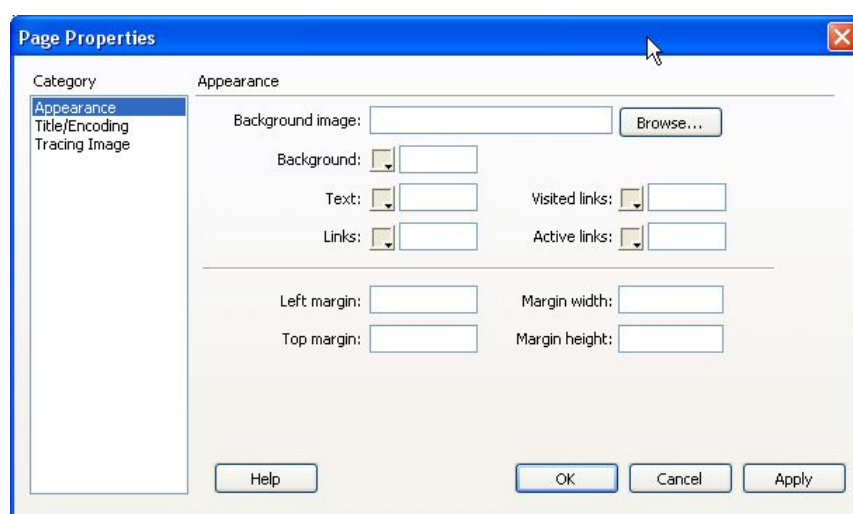


Рис. 4.1. Диалоговое окно настройки параметров страницы (категория Appearance)

Кроме заголовка, в категории **Appearance** этого диалогового окна (рис. 4.1), можно установить следующие параметры страницы:

- **Background image** – изображение, которое будет использовано в качестве фона страницы (графический файл с изображением можно выбрать, нажав кнопку **Browse...**);
- **Background / Text** – соответственно цвета фона страницы и текста, расположенного на ней (можно выбрать цвет из палитры, нажав на кнопке , расположенном рядом);
- **Links / Visited links / Active links** – соответственно цвета ссылок / посещенных ссылок / активных ссылок;
- **Left margin / Top margin / Margin width / Margin height** – задают отступы (поля) по бокам и в начале и конце страницы (Left и Top margin отображаются только в браузере Internet Explorer, а Margin width и height только в Netscape Navigator, поэтому рекомендуется использовать оба параметра одновременно для сохранения совместимости).

Основные элементы и приемы работы с HTML-документами будут рассмотрены в последующих разделах. А сейчас следует затронуть другой вопрос работы со страницей в целом – разделение ее на фреймы.

Фреймы позволяют разделить окно браузера на несколько частей, в каждую из которых будет загружен свой HTML-документ. Здесь мы не будем обсуждать недостатки и преимущества использования фреймов, а так же целесообразность их использования – более подробно с этими вопросами можно ознакомиться в пособии "Создание образовательных ресурсов на основе Internet-технологий". Если же вы решили, что для реализации структуры сайта необходимо использование фреймов, то здесь будут рассмотрены приемы работы с ними в Dreamweaver.

Прежде чем приступить к созданию фреймов, необходимо включить режим отображения их границ, выполнив команду **View=>Visual Aids=>Frame Borders**.

Начать создание фреймов проще всего выбрав один из предусмотренных вариантов, нажав на кнопку **Frames** на закладке **Layout** панели **Insert** (рис. 4.2). Кроме того, можно воспользоваться одним из вариантов, которые расположены в меню **Insert=>HTML=>Frames**.

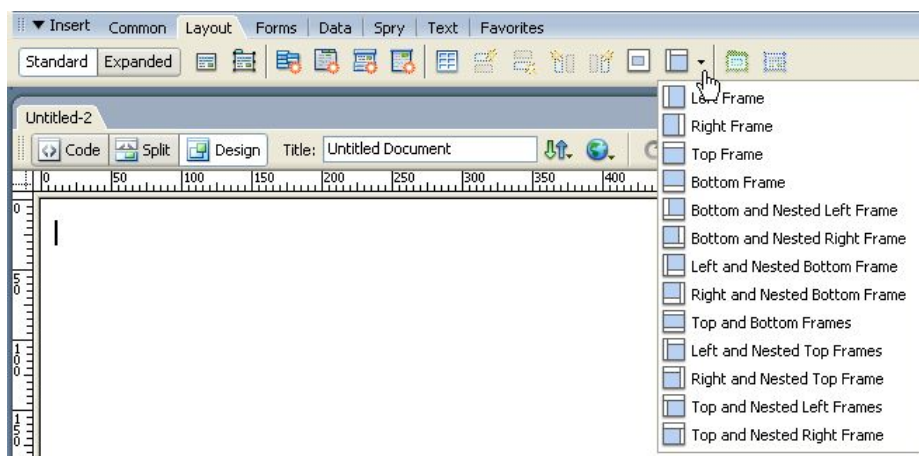


Рис. 4.2. Предустановленные наборы фреймов на закладке Frames панели Objects

После выбора фреймовой структуры откроется диалоговое окно **Frame Tag Accessibility Attributes**, в котором задаются имена фреймов. Можно оставить имена по умолчанию, нажав на кнопку **OK**.

Для добавления дополнительных фреймов можно воспользоваться командами **Modify=>Frameset=>Split Frame Left, Right, Up** или **Down** – соответственно для добавления фрейма слева, справа, сверху или снизу. Чтобы удалить фрейм нужно мышью перетащить его границу за пределы страницы.

Каждый из фреймов представляет собой отдельный HTML-документ. Поэтому, для внесения изменений в один из фреймов, или параметры набора фреймов в целом, нужно предварительно его выделить. Это можно сделать щелкнув мышью соответственно в нужном фрейме или по границе между ними, или можно воспользоваться панелью **Frames** (рис. 4.3 – чтобы ее включить нужно выбрать команду **Window=>Frames**).

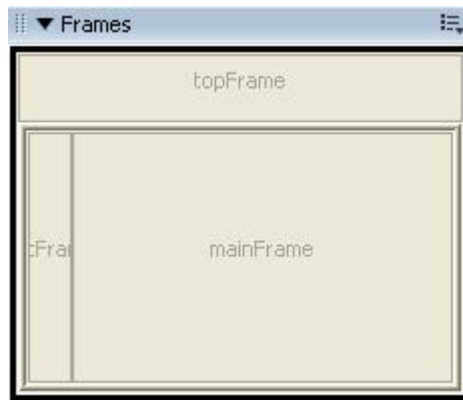


Рис. 4.3. Панель Frames

Прежде чем просматривать фреймосодержащий документ в браузере необходимо сохранить все файлы его составляющие. Для сохранения одного из фреймов нужно его выделить и выбрать команду **File=>Save Frame** или **File=>Save Frame As** для сохранения под другим именем. Чтобы сохранить сам фреймосодержащий документ необходимо выделить весь набор фреймов и выбрать команду **File=>Save Frameset** или **File=>Save Frameset As**. Команда **File=>Save All Frames** позволяет сохранить все документы входящие в набор фреймов одновременно.

Для просмотра и редактирования параметров фрейма необходимо выбрать его на панели **Frames** или щелкнуть мышью удерживая клавишу **Alt** в любом его месте в окне документа. При этом на панели **Properties** отобразятся свойства выбранного фрейма (рис. 4.4). Если нижняя часть панели свойств не видна, то ее нужно развернуть нажатием кнопки со стрелочкой ▼, расположенной в правом нижнем углу.

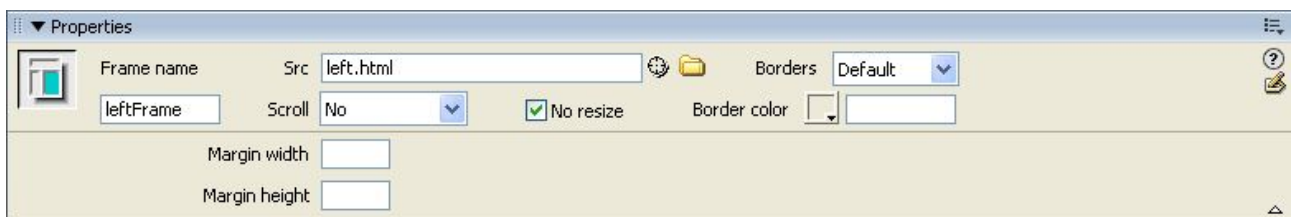


Рис. 4.4. Панель свойств в режиме отображения параметров фрейма

Параметр **Frame name** задает имя фрейма, которое может быть использовано для обращения к нему. Кроме того, можно установить следующие параметры:

- **Src** – определяет документ, который будет загружен во фрейм;
- **Scroll** – указывает будет ли выведена полоса прокрутки (возможные варианты: **Yes** – всегда, **No** – никогда, **Auto** – при необходимости, **Default** – по умолчанию, для большинства браузеров соответствует Auto);
- **No resize** – запрещает при просмотре перетаскивание границы фреймов;
- **Borders** – управляет выводом границ фреймов (**Yes** – границы отображаются, **No** – границы не отображаются, **Default** – по умолчанию, для

большинства браузеров соответствует Yes). Этот параметр перекрывает аналогичный для всего набора фреймов в целом;

- **Border color** – задает цвет границы фрейма. Этот параметр перекрывает аналогичный для всего набора фреймов в целом;
- **Margin width / Margin height** – задают величину отступа (в пикселах) между границей фрейма и его содержимым соответственно слева и справа/сверху и снизу.

Помимо изменения параметров отдельных фреймов возможна установка свойств всего набора в целом. Для их просмотра и редактирования необходимо щелкнуть по границе между фреймами в окне документа или на панели **Frames**. При этом на панели свойств (рис. 4.5) возможна установка следующих параметров:

- **Borders** – управляет выводом границ фреймов (**Yes** – границы отображаются, **No** – границы не отображаются, **Default** – по умолчанию);
- **Border width** – задает толщину границы фреймов;
- **Border color** – задает цвет границы фреймов.



Рис. 4.5. Панель свойств в режиме отображения параметров набора фреймов

Последнее, что осталось обсудить – изменение размеров фреймов. Задать размер проще всего перетаскив мышью границу фрейма в окне документа. Однако, при этом высота или ширина будут установлены "на глаз". Кроме того, следует учитывать, что размеры фреймов могут быть заданы:

- абсолютным значением (в пикселах) – задает фиксированное значение размера. Такой вариант лучше всего подходит для фреймов с постоянным размером (например, навигационной панели);
- процентным значением – задает размер фрейма в процентах относительно размера его набора (который принимается за 100%). При изменении размеров окна границы будут перемещаться на соответствующую величину;
- относительным значением – размер фрейма определяется остальными фреймами. Пространство под такие фреймы в окне браузера выделяется в последнюю очередь (первыми размещаются фреймы с абсолютными размерами, затем с процентными). Фрейм с относительным значением занимает все оставшееся свободное пространство.

Для того, чтобы указать точную величину высоты или ширины фрейма необходимо выделить набор фреймов и в правой части панели свойств (рис. 4.5) выбрать строку или столбец, размеры которого требуется изменить. В поле **Units** необходимо указать, в каких единицах будет указан размер (**Pixels** – в абсолютных, **Percent** – в процентных, **Relative** – в относительных). Само значение размера должно быть указано в поле **Value**.

4.3 Создание навигационной панели

Важнейшим, с одной стороны, а с другой стороны, являющимся вспомогательным, утилитарным компонентом web-страницы является панель навигации. Вопросы теории разработки подробно обсуждаются в пособии "Создание образовательных ресурсов на основе Internet-технологий". Здесь же мы рассмотрим инструмент **Navigation Bar**, значительно ускоряющий и упрощающий процесс разработки навигационных панелей. Этот инструмент может использоваться для создания элементов навигации сайта в целом, электронного учебника или других обучающих материалов.

Для вставки навигационной панели необходимо выполнить команду **Insert=>Image Objects=>Navigation Bar**. Диалоговое окно (рис. 4.6) позволит без программирования создать, настроить и использовать интерактивную навигационную панель.

В первую очередь следует создать элементы навигационной панели, которые будут соответствовать разделам сайта и помогать осуществлять переходы между ними. Имена навигационных элементов перечислены в списке **Nav bar elements**. Добавить новый элемент в навигационную панель можно с использованием кнопки , а удалить ненужный элемент можно при помощи кнопки . Изменить порядок следования навигационных элементов можно с использованием кнопок  и . Последовательно выбирая навигационные элементы из списка, необходимо с помощью элементов управления расположенных ниже на этом диалоговом окне, настроить их внешний вид и свойства:

- **Element name** – имя навигационного элемента;
- **Up image** – путь к файлу с изображением, которое будет использоваться в качестве навигационного элемента;
- **Over image** – путь к файлу с изображением, которое будет отображаться при наведении на навигационный элемент курсора мыши;
- **Down image** – путь к файлу с изображением, которое будет отображаться при выборе навигационного элемента левой кнопкой мыши;
- **Over while down image** – путь к файлу с изображением, которое будет отображаться при наведении курсора мыши на навигационный элемент с нажатой левой кнопкой;
- **When clicked, Go to URL** – имя файла или адрес Internet-ресурса, по которому осуществлять переход при выборе элемента навигационной панели.

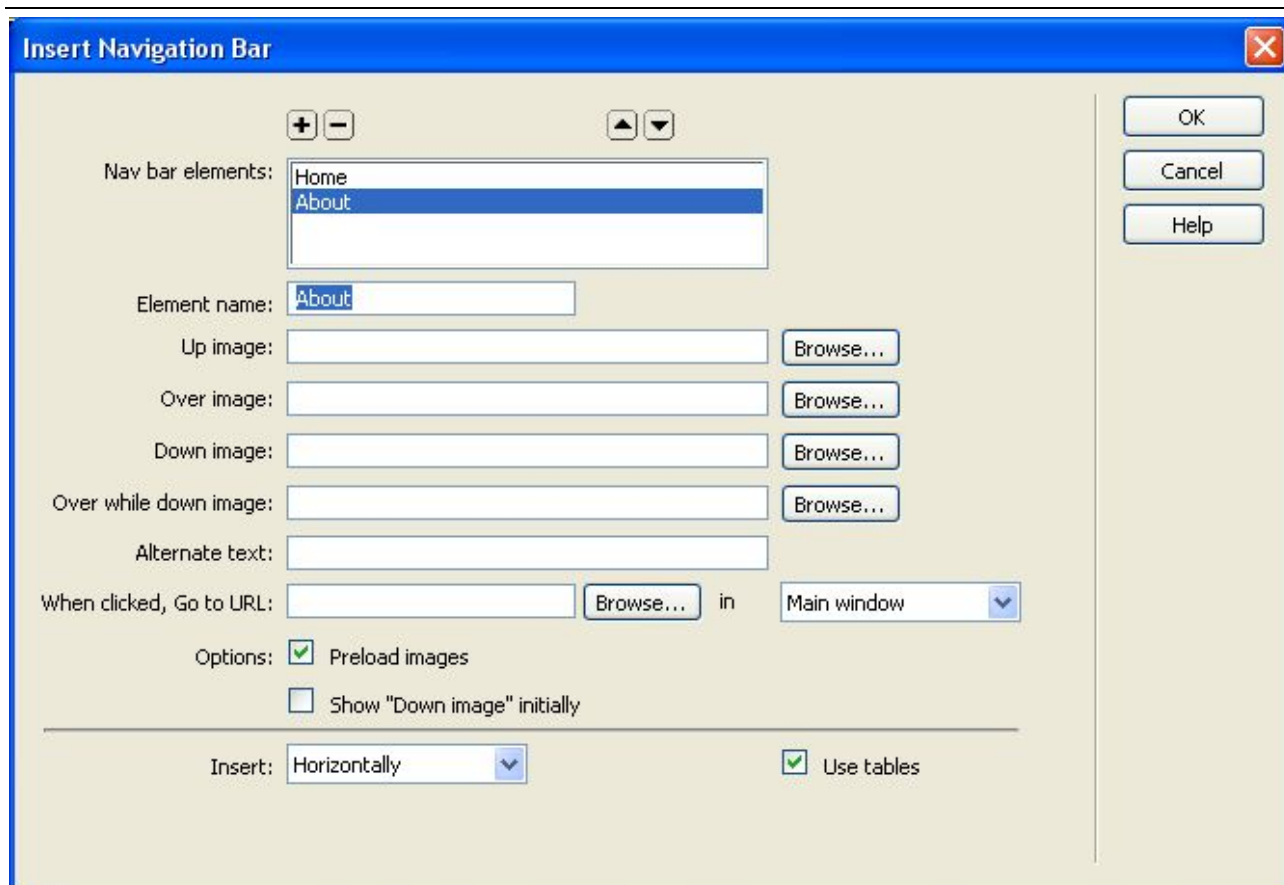


Рис. 4.6. Диалоговое окно для настройки скрипта Navigation Bar

Если включить флажок **Preload images**, то в HTML-документ будет вставлен скрипт для предварительной загрузки файлов с изображениями в кэш-память браузера.

Флажок **Show “Down image” initially** определяет внешний вид элементов навигационной панели при первоначальной загрузке HTML-документа. Если этот флажок выключен, то элементы навигационной панели отображаются в состоянии **Up**. Если этот флажок выключен, то элементы навигационной панели отображаются в состоянии **Down**.

С помощью элемента управления **Insert** можно указать, каким образом будут расположены навигационные элементы (горизонтально или вертикально). А если включить флажок **Use tables**, то элементы навигационной панели будут размещены в ячейках таблицы.

Таким образом, можно достаточно просто и быстро, без программирования, создать навигационную панель и использовать ее в качестве макета в процессе работы над сайтом или скелета для его окончательной версии.

4.4 Подготовка шаблонов

Для уменьшения рутинной работы в ходе разработки сайта полезно использовать различные средства автоматизации. Для автоматизации действий при создании и модификации сайта Dreamweaver предоставляет такие средства, как шаблоны и библиотека элементов. Кроме того, как и многие современные

программы, Dreamweaver позволяет запомнить последовательность команд и затем ее многократно повторить.

Благодаря шаблонам можно быстро и удобно создавать страницы сайта с одинаковой структурой, оформлением, а также можно действительно отделить логическое форматирование от визуального. Например, это позволяет, следуя моде, быстро изменять визуальное представление всех страниц, или, преследуя цель улучшения пользовательского интерфейса сайта, изменить его навигационные средства. Достаточно внести изменения в шаблон, и Dreamweaver автоматически внесет изменения во все страницы, созданные на его основе.

Приступать к разработке шаблонов следует после того, как продумана структура сайта, навигационные средства и дизайн, когда можно выделить основные типы документов, входящие в его состав. Для этих основных типов документов и следует разработать шаблоны.

Для создания шаблона необходимо в обычном HTML-документе создать типовую структуру, которую должны иметь все или несколько страниц разрабатываемого сайта. Создав на странице типовую структуру, разместив на ней навигационные и дизайнерские элементы, можно превратить этот документ в шаблон, выполнив команду **File=>Save as Template**. В диалоговом окне (рис. 4.7) необходимо уточнить, для какого сайта создается шаблон (поле **Site**), а также дать шаблону имя (поле **Save as**).

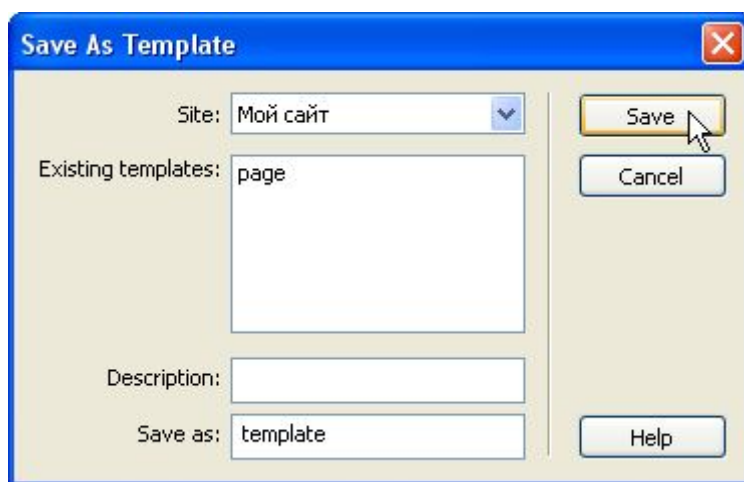


Рис. 4.7. Диалоговое окно сохранения шаблона

Подготовленный таким образом шаблон еще нельзя использовать для создания документов. Дело в том, что все содержимое документа, созданного на основе такого шаблона, будет заблокировано, “заморожено”, а редактирование документа будет невозможно. При создании шаблона необходимо указать, какие области можно редактировать, а какие следует заблокировать и оставлять неизменными. Создание новой, пустой редактируемой области (editable region) выполняется по команде **Insert=>Template Objects=>Editable Region**. В появившемся диалоговом окне (рис. 4.8) необходимо дать имя редактируемой области. Новая редактируемая область будет создана в том месте, где стоял курсор. В

редактируемую область можно превратить и любой участок уже существующего текста, если предварительно его выделить.

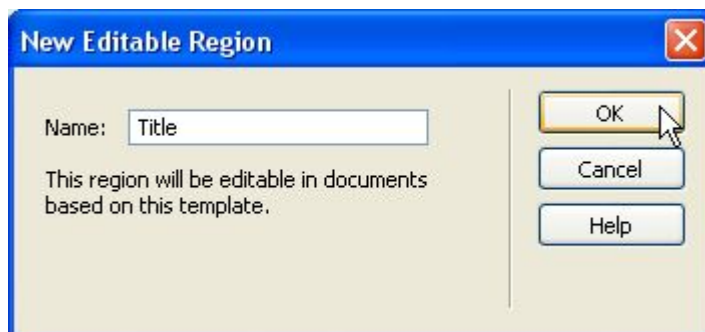


Рис. 4.8. Диалоговое окно создания редактируемой области шаблона

Если в настройках Dreamweaver включена возможность отображения невидимых элементов разметки, то редактируемая область будет выделена цветом, а в HTML-код документа будет вставлен следующий текст:

```
<!-- TemplateBeginEditable name="Title" -->
Title
<!-- TemplateEndEditable -->
```

С помощью особой структуры внутри HTML-комментариев создается служебная разметка, которая заставляет Dreamweaver воспринимать этот текст как редактируемую область шаблона. Вмешиваться и изменять эту структуру не следует. Так как служебная разметка оформлена в виде комментариев, влиять на отображение HTML-документа в браузере она не будет.

При разработке шаблона может возникнуть необходимость изменить положение редактируемых областей или удалить некоторые из них. Для этого следует выделить область, которую нужно отменить и выполнить команду **Modify=>Templates=>Remove Template Markup**.

Когда все редактируемые области шаблона определены, можно его сохранить. Теперь на основе шаблона можно создавать новые документы, которые автоматически наследуют все заданные свойства страницы, имеют такую же структуру, навигационные элементы и оформление. Для создания нового файла на основе шаблона следует выполнить команду **File=>New**.

В диалоговом окне **New Document** (рис. 4.9) следует выбрать пункт **Page from Template** и указать имя шаблона (**Template for Site**), на основе которого будет создан новый документ. Флажок **Update page when template changes** по умолчанию включен. Это приведет к тому, что документ будет связан с шаблоном и любые изменения в шаблоне приведут к изменениям и в данном документе.

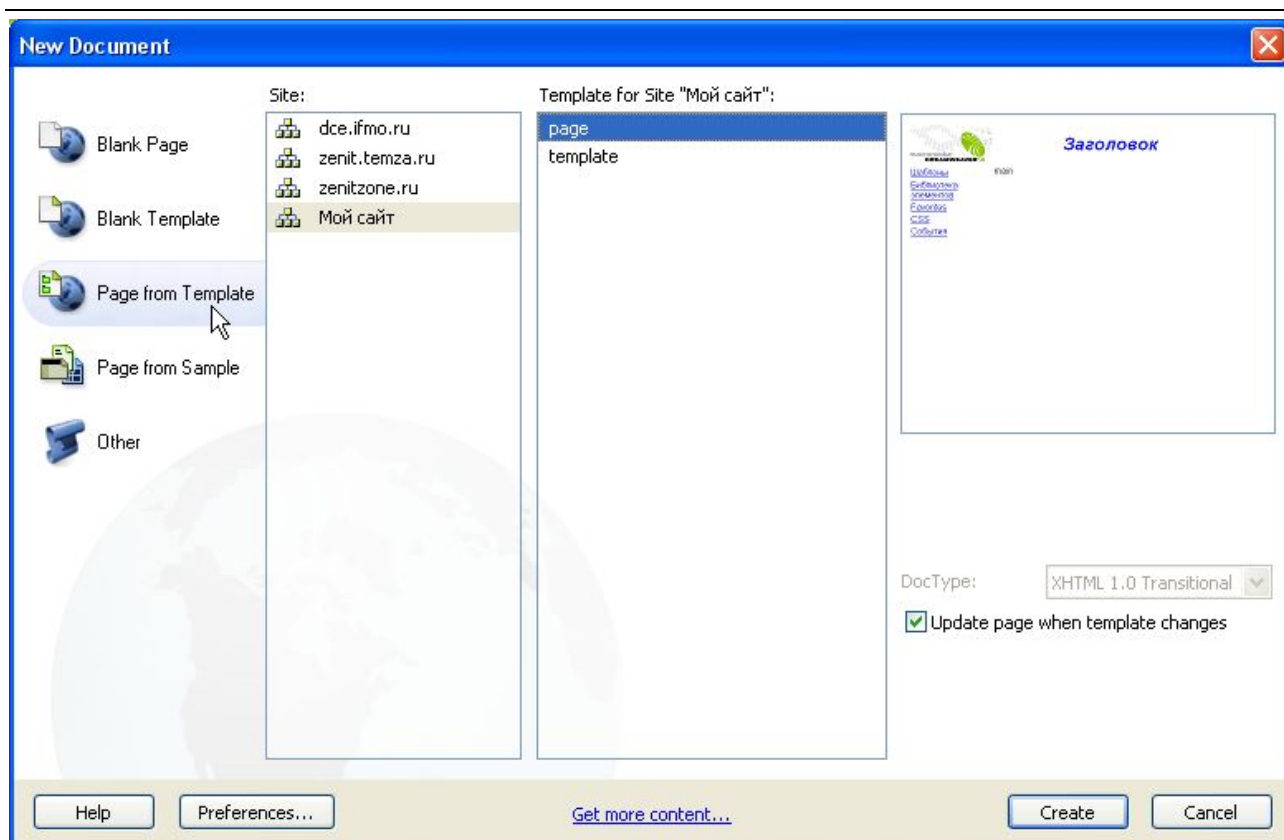


Рис. 4.9. Диалоговое окно создания нового документа на основе шаблона

Таким образом, редактируемые области текста можно будет модифицировать. Остальные части HTML-документа будут заблокированы. HTML-текст шаблона, в котором реализуется его структура, навигация и дизайн ни случайно, ни принудительно изменить нельзя. Таким образом, шаблоны не только избавляют от лишних действий, но позволяют избежать случайных ошибок. Иногда может возникнуть необходимость внести изменения в заблокированную область. Для этого необходимо разорвать связь между данным документом и шаблоном. Это можно сделать по команде **Modify=>Templates=>Detach from Template**. После выполнения этой команды Dreamweaver позволит редактировать любой участок этого документа. Но прежде чем выполнять эту операцию, следует подумать о внесении изменений в шаблон или о разработке нового шаблона для документов такого типа. Внесение изменений в шаблон может потребоваться и по другим причинам (изменение дизайна, изменение панели навигации). Благодаря использованию шаблонов, Dreamweaver может выполнить эти изменения и достаточно быстро сделать исправления во всех файлах, созданных на основе шаблона.

Для внесения изменений в шаблон, на основе которого создан редактируемый файл, следует выполнить команду **Modify=>Templates=>Open Attached Template**. По этой команде в новом окне Dreamweaver откроет шаблон, в который можно внести требуемые изменения. Шаблон можно также открыть для редактирования из папки Templates сайта, которую создал Dreamweaver специально для хранения шаблонов.

После внесения изменений в шаблон необходимо выполнить процедуру обновления файлов, созданных на его основе. Эта процедура будет вызвана

автоматически, если отредактированный шаблон сохраняется или закрывается. Ее можно вызвать принудительно, выполнив команду **Modify=>Templates=>Update Pages**.

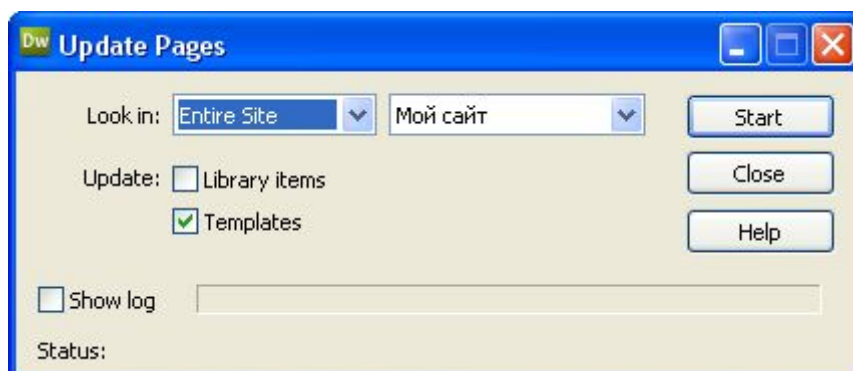


Рис. 4.11. Диалоговое окно обновления страниц, созданных на основе шаблона

В диалоговом окне **Update Pages** (рис. 4.11) в поле **Look in** следует указать, какие файлы следует обновить: либо документы в рамках данного сайта (Entire Site), либо все документы, созданные на основе этого шаблона (Files That Use...). После обновления файлов отчет о выполненных действиях выводится в нижней части окна (рис. 4.12).

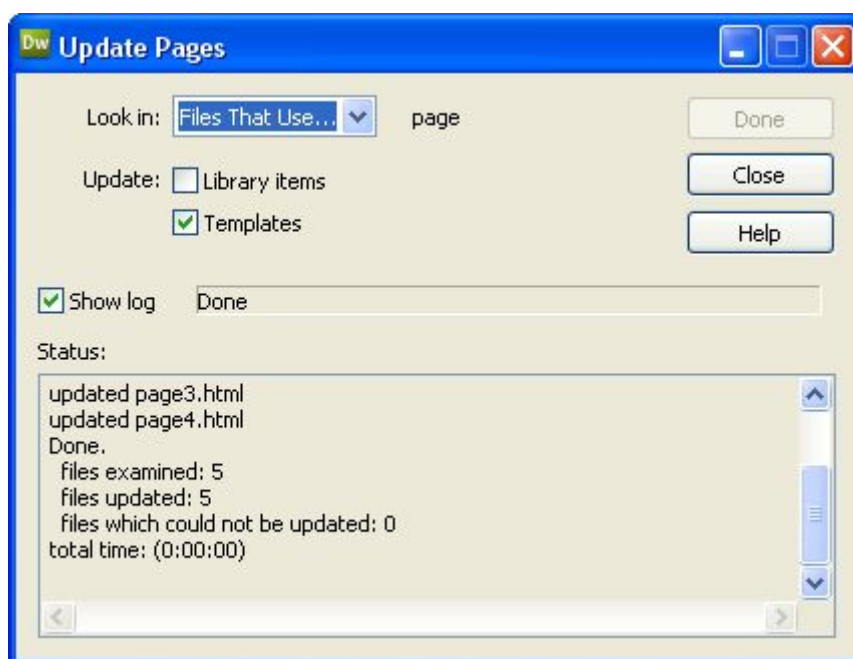


Рис. 4.12. Отчет об обновленных файлах

Можно также выполнить обновление каждой страницы отдельно. Для этого при редактировании страницы, созданной на основе шаблона, следует выполнить команду **Modify=>Templates=>Update Current Page**. По этой команде содержимое нередатируемых областей документа будет синхронизировано с шаблоном, на основе которого он был создан.

Возможно, уже было создано несколько страниц без использования шаблонов, а переделывать их достаточно долго и трудоемко, или они были

созданы на основе какого-либо другого шаблона другого сайта. Для автоматизации работы и с этими страницами можно их привязать к одному из шаблонов. Для этого достаточно выполнить команду **Modify=>Templates=>Apply Template to Page....** В диалоговом окне **Select Template** необходимо выбрать имя шаблона, который будет применен и прикреплен к этой странице. Dreamweaver добавит все необходимое содержимое из шаблона, а старый текст разместит в соответствующих редактируемых областях.

Работу с шаблонами можно осуществлять с использованием палитры **Templates** на панели **Files** (рис. 4.13).

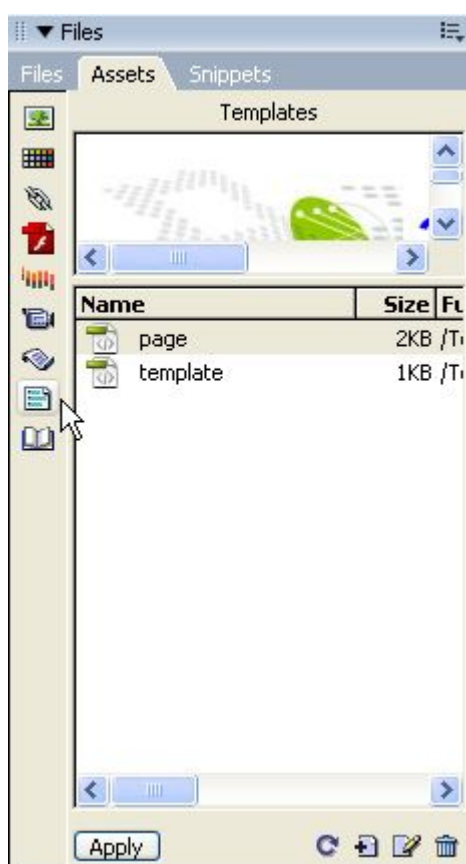


Рис. 4.13. Палитра Templates

Так Dreamweaver осуществляет работу с шаблонами, которые освобождают от рутинных операций по копированию, дублированию и настройке подобных элементов, делает автоматическими сложные операции по обновлению, изменению и синхронизации элементов сайта.

4.5 Подготовка библиотеки элементов

На автоматизацию работы с повторяющимися элементами направлено такое средство, как библиотека (Library). Библиотека реализована практически во всех продуктах фирмы Adobe и используется для хранения и управления элементами, которые часто встречаются в редактируемых документах.

При создании сайта всегда можно выделить ряд объектов (картинки, текстовые отрывки, навигационные элементы, элементы дизайна), которые

используются много раз на многих страницах. Такие элементы можно выделить и поместить в библиотеку и затем многократно использовать. Если внести изменения в типовые объекты из библиотеки, то эти изменения будут учтены на всех страницах сайта.

Работать с объектами библиотеки эффективнее всего с использованием палитры **Library** на панели **Files** (рис. 4.14). Для создания элемента библиотеки следует выделить область HTML-документа и перетащить ее на палитру **Library**. То же самое можно сделать, выполнив команду **Modify=>Library=>Add Object to Library**.

Вставить элемент библиотеки в редактируемый документ также достаточно просто. Элемент необходимо выбрать из палитры **Library** и перетащить в желаемое место.

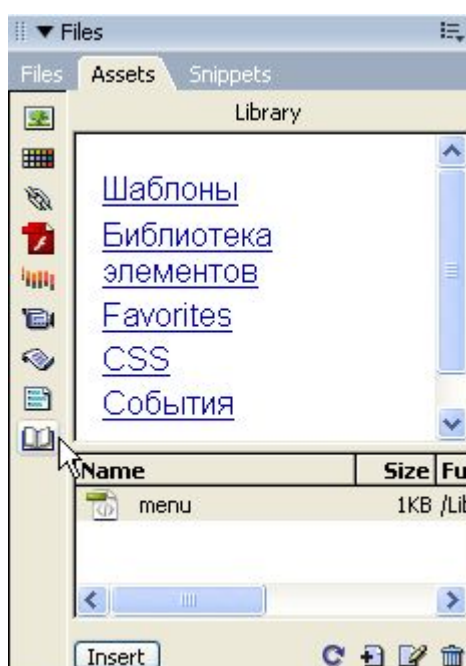


Рис. 4.14. Палитра Library (Библиотека)

Двойной щелчок мышью на имени элемента из библиотеки позволит отредактировать этот элемент. После редактирования будет вызвана процедура обновления файлов, которая работает также, как и при обновлении после редактирования шаблонов.

Элементы из библиотеки можно использовать в качестве основы для создания новых элементов. В этом случае связь с библиотекой будет только мешать. Отредактировать такой элемент будет невозможно. Поэтому необходимо выполнить команду **Detach from Original**, которую можно найти в контекстном меню, если щелкнуть правую кнопку мыши, когда ее курсор находится над элементом из библиотеки. После этого элемент станет “самостоятельным” и при обновлении файла изменения в библиотеке не повлияют на данный элемент страницы.

5. Форматирование HTML-страниц: логическое и визуальное

Идеологической основой языка HTML является концепция структурной разметки, которая заключается в разделении содержания документа и его представления. Таким образом, можно выделить два уровня форматирования документа: логический и визуальный. При форматировании HTML-страницы следует уделять больше внимания ее содержанию, абстрагируясь от представления, задавать лишь структуру документа. Для определения же визуального представления документа (интервалы между строками текста, отступы, цвета, используемые для текста и фона, размер и стиль шрифтов) следует использовать таблицы стилей (CSS – Cascading Style Sheets).

Рассмотрим основные элементы логического форматирования HTML и их практическое использование при подготовке документов с использованием HTML-редактора Dreamweaver.

5.1 Элементы структурной разметки

Каждый документ должен содержать заголовок. Причем, это должен быть не просто выделенный крупным шрифтом, цветом или каким-либо другим образом текст, а именно структурная единица – заголовок. В языке HTML существует шесть уровней заголовков: тег <H1> задает наиболее важный (заголовок первого уровня) и <H6> – наименее важный (заголовок шестого уровня). При использовании этих элементов следует соблюдать их иерархию. Общим заголовком документа должен быть заголовок первого уровня. Не следует пропускать уровни, используя для заголовков разделов сразу заголовки третьего или четвертого уровня.

Любой текст разделяется на последовательность абзацев. При этом организация информации не должна влиять на их представление: абзацы с двойным выравниванием передают тот же смысл, что и абзацы с выравниванием влево. В большинстве текстовых редакторов разбиение текста на абзацы реализуется нажатием клавиши **Enter**. В HTML для определения абзаца используется тег <P> (закрывающий тег необязателен). Перенос строк в пределах абзаца осуществляется автоматически по символам-разделителям слов (то есть пробелам). Множественные пробелы и все переводы строки в исходном тексте игнорируются.

Иногда требуется осуществить принудительный перевод строки. Для этого используется тег
 (закрывающего тега не имеет). Хотя в ряде случаев визуальный эффект использования этих двух тегов может совпадать, следует четко понимать разницу между ними. Так, например, в большинстве браузеров два тега
, использованных подряд, визуально будут соответствовать новому абзацу, однако в структуре документа оба образовавшихся блока текста составляют один абзац.

Бывают случаи, когда в HTML-документ необходимо включить блок текста, предварительно отформатированный традиционным способом (с использованием символов перевода строки, пробелов и символов табуляции).

Для этих целей используется тег-контейнер <PRE>. Текст, ограниченный этим тегом, будет отображаться моноширным шрифтом в таком виде, как он выглядит в обычном текстовом редакторе.

Рассмотрим реализацию такой структурной разметки средствами Dreamweaver. Для того, чтобы начать новый абзац, как и в большинстве обычных текстовых редакторов, достаточно нажать клавишу **Enter**. При этом в исходный текст HTML-документа автоматически вставляются открывающий и закрывающий теги <P>. Для создания принудительного перехода на новую строку используется комбинация клавиш **Shift+Enter**. В исходный код будет добавлен тег
. Также принудительный переход можно вставить в текст, выполнив команду **Insert=>HTML=>Special Characters=>Line Break**.

Для каждого абзаца текста можно также установить его логический тип, то есть сделать заголовком любого уровня или преформатированным текстом. Тогда эти абзацы займут свое место в структурной иерархии документа. Проще всего это сделать, используя панель свойств объекта (рис. 5.1). Если панели нет на экране, то можно ее открыть, выполнив команду **Window=>Properties**. Кроме того, указать тип блока текста (абзаца) можно в меню **Text=>Paragraph Format** или в разделе **Paragraph Format** контекстного меню, появляющегося при щелчке правой кнопкой мыши в поле документа.

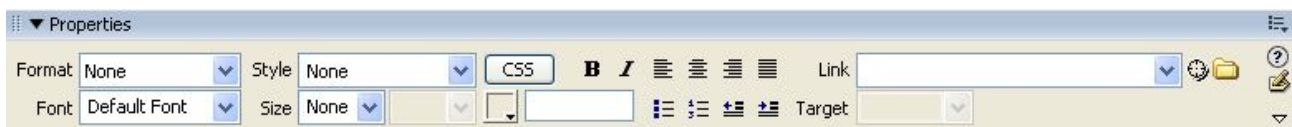


Рис. 5.1. Панель свойств объектов

Выделив блок текста или просто установив курсор в любое место абзаца, можно выбрать его тип в поле **Format**. В соответствии с элементами языка HTML возможны следующие варианты: **Paragraph** – обычный абзац (ограничен тегами <P>), **Heading 1...6** – заголовки соответствующего уровня (ограничены тегами <H1>...<H6>), **Preformatted** – предварительно форматированный текст (тег <PRE>) или **None** – блок текста перестает быть абзацем. Эти действия можно выполнить из пункта меню **Text=>Paragraph Format**.

Общий механизм добавления в документы структуры обеспечивается тегами и <DIV> (вместе с атрибутами ID и CLASS). Эти элементы определяют встраиваемую строчную информацию (тег) или информацию уровня блока (тег <DIV>) и являются контейнерами, выделяющими фрагмент документа с целью управления его параметрами. Таким образом, возможно использование этих тегов с таблицами стилей для определения визуального представления блока текста. Данные теги Dreamweaver использует автоматически для элементов разметки и форматирования, а так же при создании слоев и использовании стилей CSS (речь о которых пойдет ниже).

Для добавления в текстовые фрагменты структуры обычно используются так называемые *фразовые элементы*, которые создаются с использованием следующих тегов:

- – создает логическое выделение;
- – создает сильное выделение;
- <CITE> – содержит цитату или ссылку на другие ресурсы;
- <DFN> – указывает, что это является определением вложенного термина;
- <CODE> – содержит фрагмент программного кода;
- <SAMP> – используется для выделения примеров программ;
- <KBD> – выделяет текст, который должен ввести пользователь;
- <VAR> – создает экземпляр переменной или аргумента программы;
- <ABBR> – выделяет сокращенную форму или аббревиатуру;
- <ACRONYM> – выделяет акроним.

Для того, чтобы применить эти фразовые элементы к блоку текста в Dreamweaver, необходимо выделить его и выбрать соответствующий пункт из меню **Text=>Style**, которое также доступно по правой кнопке мыши.

5.2 Визуальное форматирование текста

Как говорилось ранее, использование тегов визуального форматирования расходится с идеологией гипертекста и потому является нежелательным. Однако в большинстве практических случаев избежать применения визуальных тегов не удастся. Это связано с тем, что использование стилей CSS, предназначенных для визуального форматирования текста, требует дополнительных знаний и навыков. Кроме того, на практике невозможно гарантировать их абсолютную совместимость с различными браузерами различных версий.

Основные функции форматирования текста собраны на панели свойств (рис. 5.1). Для выделенного блока текста можно указать гарнитуру шрифта, выбрав один из пунктов списка **Font**. Пункт **Default Font** означает, что для отображения текста браузер будет использовать шрифт по умолчанию. Следует обратить внимание, что в этом списке указаны не отдельные названия шрифтов, а комбинации шрифтов со схожими гарнитурами. Это связано с тем, что при отображении на клиентской машине, возможна ситуация, когда указанный шрифт не будет на ней установлен. В этом случае будет использован следующий из указанных шрифтов. Если на клиентской машине не окажется ни одного из перечисленных шрифтов, то будет использован шрифт, заданный в браузере по умолчанию. Изменить список шрифтов (добавить, удалить или изменить комбинации) можно при помощи команды **Edit Font List...**, расположенной в конце списка. Кроме того, гарнитуру шрифта можно выбрать в разделе **Font** контекстного меню или выполнив команду **Text=>Font**.

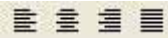
Размер шрифта можно выбрать в поле **Size** (или в меню **Text=>Size** и **Text=>Size Change** или разделе **Size** контекстного меню). Размер задается в относительных единицах, и может принимать значения от 1 до 7 (по умолчанию используется 3). Каждый из этих "виртуальных" размеров на 20% больше или меньше значения по умолчанию. Так же величина шрифта может

4.5. Визуальное форматирование текста

быть указана как относительная по отношению к основной в диапазоне от +7 до -7. При этом размер шрифта никогда не будет больше 7 или меньше 1, даже если относительный размер дает иную величину. Язык HTML не имеет возможности указания абсолютного размера шрифта в пунктах, как это принято в традиционных текстовых редакторах (это можно сделать при помощи стилей CSS).

Цвет текста можно указать в поле **Text Color** в виде шестнадцатеричного значения, которое затем и будет вписано в исходный код HTML-страницы. Цвет задается в виде #RRGGBB, где # открывает шестнадцатеричную запись, а RR, GG и BB – определяют величину соответственно красной, зеленой и синей цветовых компонент (могут принимать значения в диапазоне от 00 до FF). Гораздо удобнее выбрать цвет из палитры. По кнопке  раскрывается палитра с набором “безопасных” цветов. Так называемая *безопасная web-палитра* состоит из 216 цветов, входящих в системные палитры Macintosh и PC. Благодаря этому, цвета входящие в ее состав выводятся одинаково точно на дисплеях Mac и PC. Для выбора цвета можно использовать инструмент “пипетка”, которая появляется когда раскрыта палитра. Этой пипеткой можно “взять” цвет с любого элемента редактируемой страницы или самого окна программы. Если нажать кнопку **Default Color** на палитре, то поле значения цвета будет очищено, и текст отображаться цветом по умолчанию.

При помощи кнопок **Bold**  и **Italic**  выделенный фрагмент текста можно сделать жирным и/или наклонным соответственно. Дополнительные стили оформления текста можно выбрать в меню **Text=>Style** или в разделе **Style** контекстного меню.

Выравнивание абзаца текста можно задать при помощи кнопок **Left**, **Center**, **Right** и **Justify** . Также это можно сделать, выбрав тип выравнивания в меню **Text=>Align** или в разделе **Align** контекстного меню.

5.3 Создание гипертекстовых ссылок

Важнейшим элементом любого гипертекстового документа являются гипертекстовые ссылки. В языке HTML они реализуются с использованием тега <A>. Для создания ссылки в редакторе Dreamweaver необходимо выделить объект (блок текста или изображение), а затем на панели свойств объекта (рис. 5.1) в поле **Link** указать URL файла, на который будет осуществляться гипертекстовый переход. Если это внутренняя ссылка (то есть файл входит в состав данного сайта), то адрес перехода проще всего указать с использованием инструмента **Browse for File**, который вызывается по кнопке , расположенной рядом. В отрывшемся стандартном диалоговом окне необходимо выбрать файл, на который будет осуществляться переход. Если же это внешняя ссылка, то необходимо указать полный URL ресурса, начиная с имени протокола (например, http://spb.fio.ru/). Поле **Target** указывает, где должен быть открыт документ, на который указывает ссылка. Это может быть одно из четырех предустановленных значений:

- **_blank** – документ загружается в новом неименованном окне;
- **_parent** – документ загружается в родительском фрейме или окне;

- **_self** – документ загружается в то же окно или фрейм, что и исходный;
- **_top** – документ открывается в том же окне браузера, но все существующие фреймы при этом удаляются.

По умолчанию все ссылки открываются в том же окне или фрейме (то есть используется установка **_self**). Если необходимо сделать ссылку из одного фрейма в другой (т.е. документ, на который указывает ссылка должен быть загружен в другом фрейме), то в поле **Target** необходимо указать имя целевого фрейма. Кроме того, в поле может быть указано имя окна браузера, в котором следует открыть ссылку. Если такого окна не существует, то будет открыто новое и ему будет присвоено указанное имя.

Кроме ссылок на файлы целиком, существует возможность создания ссылки на определенное место в документе – *якорь*. Для этого сначала необходимо создать в документе сами якоря. Это можно сделать, выполнив команду **Insert=>Named Anchor** или нажав кнопку **Named Anchor** на закладке **Common** панели объектов (рис. 5.2).



Рис. 5.2. Вставка якоря

В открывшемся диалоговом окне (рис. 5.3) в поле **Anchor name** необходимо указать имя якоря, которое должно состоять из латинских символов и/или цифр и не может содержать пробелов. Для создания ссылки на якорь в поле **Link** после имени файла необходимо вписать символ # и указать после него имя якоря (например, filename.html#part1). Если якорь находится в этом же файле, то имя файла можно опустить (например, #part1).

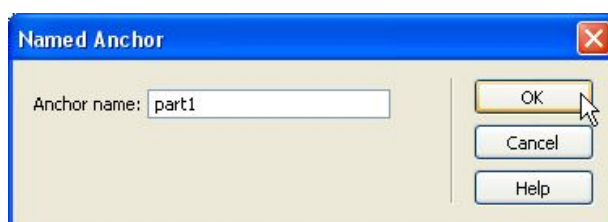


Рис. 5.3. Диалоговое окно для вставки якоря

Отдельный случай представляют собой ссылки на адреса электронной почты. Для создания таких ссылок можно воспользоваться либо кнопкой **Email Link** на закладке **Common** панели объектов (рис. 5.4) или выполнить команду **Insert=>Email Link**.



Рис. 5.4. Вставка ссылки на адрес электронной почты

В открывшемся диалоговом окне (рис. 5.5) в поле **Text** необходимо указать тот текст, с которого будет идти ссылка, а в поле **E-Mail** – адрес электронной почты, на который эта ссылка будет указывать.

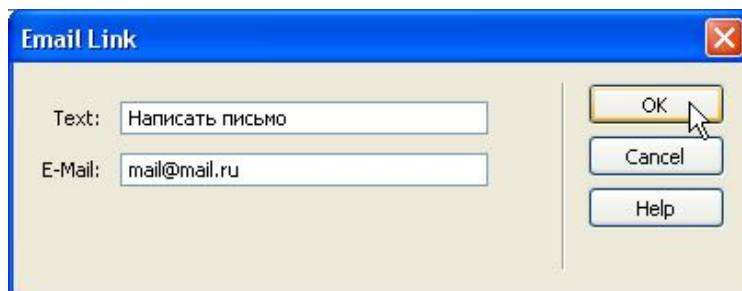


Рис. 5.5. Диалоговое окно для создания ссылки на адрес электронной почты.

5.4 Создание списков

Язык HTML предоставляет возможность организации информации в виде списка. Возможно создание списков следующих типов:

- *маркированный список* (тег) – для представления неупорядоченной информации;
- *нумерованный список* (тег) – для представления упорядоченной информации;
- *список определений* (тег <DL>) – для представления информации вида термин/определение.

Элементы маркированных и нумерованных списков создаются при помощи тега (закрывающий тег необязателен).

Списки определений отличаются от других типов тем, что их элементы состоят из двух частей: термина и определения. Термин задается с помощью тега <DT>, а его описание указывается с помощью тега <DD>.

В редакторе Dreamweaver списки создаются очень просто. Для создания нового маркированного или нумерованного списка можно воспользоваться кнопками на панели свойств (рис. 5.6). Так же можно использовать команды из пункта меню **Text=>List: Unordered** – маркированный список, **Ordered** – нумерованный, **Definition List** – список определений. Для создания следующего элемента списка необходимо нажать клавишу **Enter**, а для завершения списка – нажать **Enter** дважды.

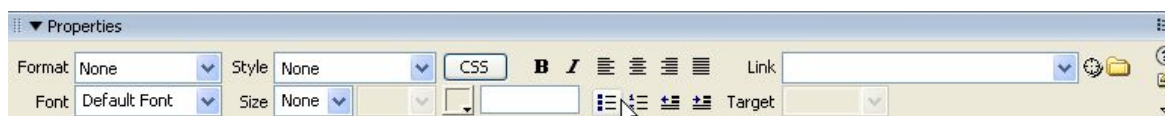


Рис. 5.6. Кнопки для создания списков

Списки можно создавать из уже существующего текста – для этого необходимо выделить несколько абзацев, каждый из которых станет отдельным элементом, а затем указать тип списка одним из описанных выше способов.

5.5 Создание и форматирование таблиц

Одним из наиболее широко используемых в HTML средств являются таблицы. Причем, очень часто они используются не только традиционно – как метод представления табличных данных, но и как средство форматирования web-страниц. Хотя следует заметить, что второй вариант использования противоречит идеологии языка HTML.

Основное назначение HTML-таблиц – упорядочить данные (текст, форматированный текст, изображения, ссылки, формы, поля форм, другие таблицы) и разместить их в строки и столбцы ячеек.

Одним из главных принципов модели таблиц HTML является то, что размеры ячеек браузер определяет автоматически в зависимости от содержимого. Размеры таблиц изменяются динамически в соответствии с текущими размерами окна. И хотя в языке HTML предусмотрена возможность явного задания размеров ячеек таблиц, автор должен прибегать к этому только в самом крайнем случае.



Рис. 5.7. Вставка таблицы

Добавить таблицу в редактируемый документ в Dreamweaver можно, как обычно, несколькими способами. Самый простой способ – это выбрать кнопку **Insert Table** на панели объектов (рис. 5.7). Также можно воспользоваться командой **Insert=>Table**.

После этого на экране появляется диалоговое окно **Insert Table** (рис. 5.8), в котором предлагается указать параметры будущей таблицы. В разделе **Table size**, в полях **Rows** и **Columns** задается количество строк и столбцов соответственно, в поле **Table width** – ширина таблицы (может быть относительной в процентах или абсолютной в пикселях). Параметр **Border thickness** задает толщину рамки вокруг всей таблицы в целом и каждой из ее ячеек. Если указано нулевое значение, то рамка не будет отображаться. Параметр **Cell padding** задает расстояние между содержимым ячейки и ее границами (по умолчанию 1 пиксел), а **Cell spacing** – расстояние между соседними ячейками таблицы (по умолчанию 2 пиксела).

В разделе **Header** можно задать "шапку" и выделенный первый столбец таблицы. В этом наборе доступны варианты **None** (нет ни "шапки", ни выделенного первого столбца), **Left** (выделенный первый столбец), **Top** ("шапка") и **Both** (и "шапка", и выделенный первый столбец).

При том ячейки, составляющие "шапку" и выделенный столбец, будут оформлены как ячейки заголовка, а текст, который мы введем в них, будет автоматически выровнен по центру и выделен полужирным шрифтом.

В разделе **Accessibility** в поле **Caption** задается название таблицы. Это название будет располагаться рядом с создаваемой таблицей. Раскрывающийся список **Align caption** позволяет задать местоположение и выравнивание названия. Здесь доступны следующие пункты:

- *default (по умолчанию)* – выравнивание выполняет сам Web-обозреватель, обычно в этом случае название находится над таблицей и выравнивается по центру;
- *top* – название находится над таблицей и выравнивается по центру;
- *bottom* – название находится под таблицей и выравнивается по центру;
- *left* – название находится над таблицей и выравнивается по левому краю;
- *right* – название находится над таблицей и выравнивается по правому краю.

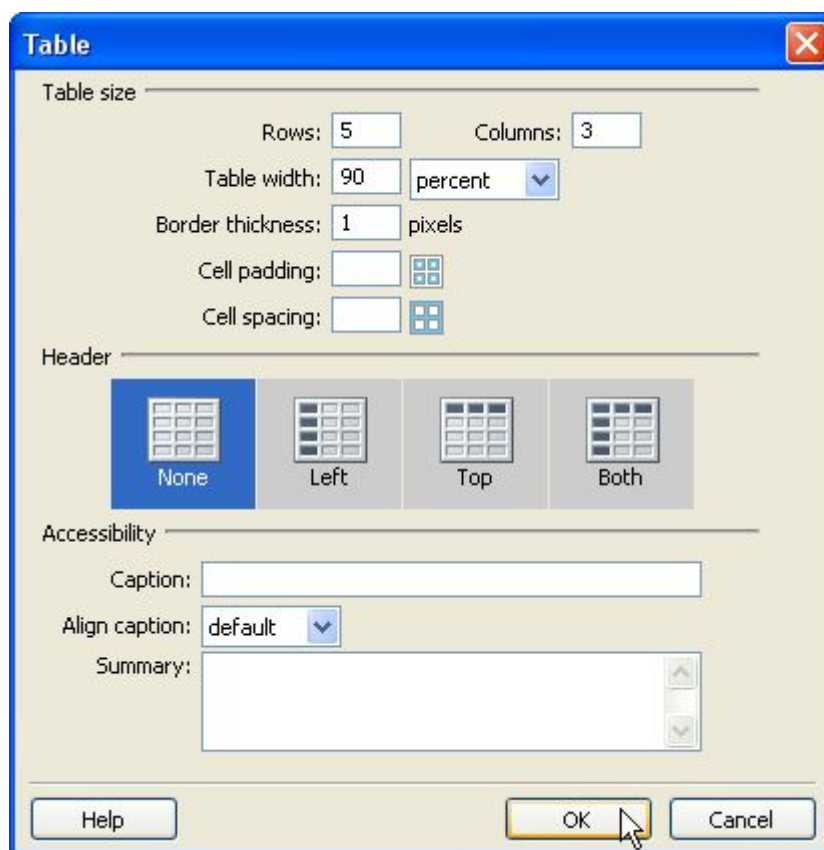


Рис. 5.8. Диалоговое окно параметров вставки таблицы

После того, как таблица создана, можно наполнять ее содержанием. Кроме того, можно изменять параметры таблицы: добавлять и удалять строки и столбцы, объединять и делить ячейки, менять размеры всей таблицы и отдельных строк и столбцов, цвет фона. Для того, чтобы изменить свойства всей таблицы в целом необходимо ее выделить, щелкнув мышкой в левом верхнем углу или по нижней или правой границе таблицы. Также можно выполнить команду **Modify=>Table=>Select Table** (в этом случае курсор должен находиться в любой ячейке таблицы). Вид панели свойств после

выделения всей таблицы показан на рисунке 5.9. Здесь можно изменить количество строк и столбцов (**Rows** и **Cols**), ширину таблицы (**W**), параметры отступов (**CellPad** и **CellSpace**) толщину рамки (**Border**). Кроме того, можно указать выравнивание таблицы относительно остальных элементов абзаца (**Align**), цвета фона и рамки таблицы (**Bg color** и **Brdr color**), а также задать фоновое изображение (**Bg Image**).

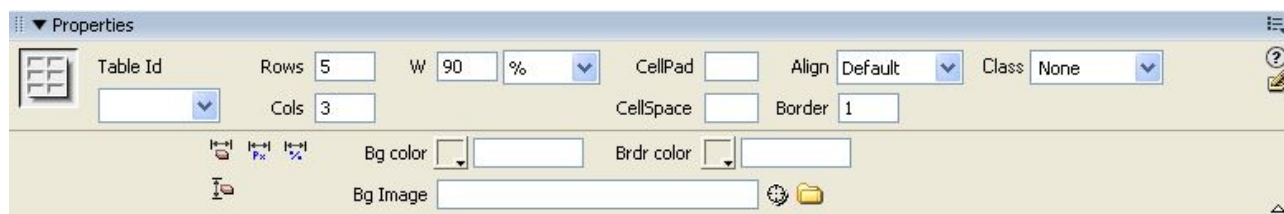


Рис. 5.9. Панель свойств таблицы

Полезными могут быть четыре инструмента, кнопки которых расположены на панели свойств слева внизу (рис. 4.9). Инструменты **Clear Row Heights** и **Clear Column Widths** предназначены для удаления значений высоты строк и ширины колонок таблицы соответственно. Инструменты **Convert Table Widths to Pixels** и **Convert Table Widths to Percent** предназначены для преобразования значений ширины в таблице из относительных единиц в абсолютные и обратно соответственно.

Для изменения свойств отдельной ячейки необходимо установить курсор в эту ячейку, а для группы ячеек – выделить их. При этом в нижней части панели свойств появятся параметры выделенной группы ячеек (рис. 5.10). Можно указать их ширину и высоту (поля **W** и **H** соответственно), цвета фона и рамки ячеек (**Bg** и **Brdr** соответственно), а также задать фоновое изображение в поле **Bg**. Кроме того, можно указать вертикальное и горизонтальное выравнивание в ячейках (поля **Vert** и **Horz** соответственно).

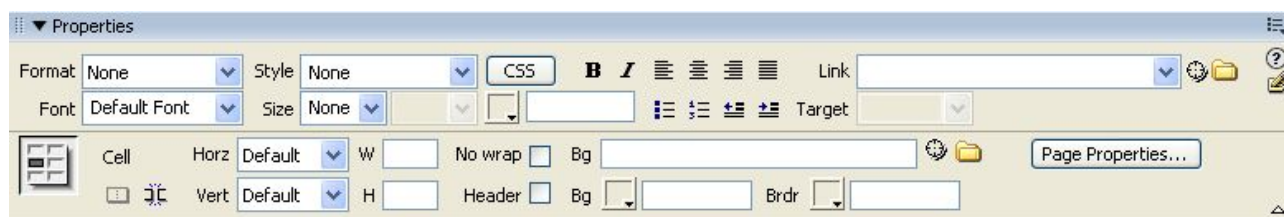


Рис. 5.10. Свойства ячейки таблицы

Следует обратить внимание на то, что указывать высоту отдельных ячеек, следует в самом крайнем случае. Предпочтительно оставлять эти поля пустыми, тогда высота будет определяться содержимым, заполняющим ячейки таблицы. При помощи кнопок **Merge** и **Split** можно соответственно объединить или разделить ячейки. Кроме того, это можно сделать, вызвав команды **Split Cell...** или **Merge Cells** из пункта меню **Modify=>Table** или из раздела **Table** контекстного меню. При делении ячеек открывается диалоговое окно **Split Cell**

(рис. 5.11), в котором предлагается указать количество строк (**Rows**) или столбцов (**Columns**) разбиения.

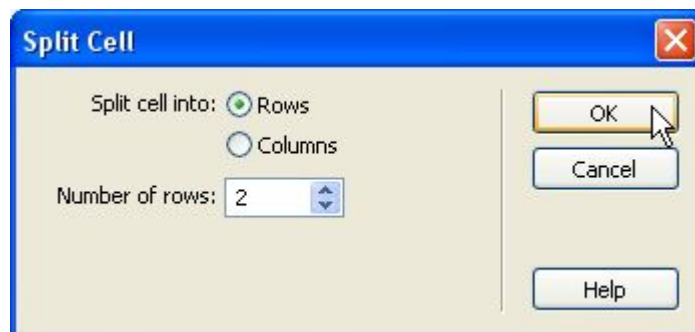


Рис. 5.11. Параметры деления ячейки таблицы

В таблице можно удалять и добавлять строки или столбцы. Команды, предназначенные для этого, собраны в пункте меню **Modify=>Table** и в разделе **Table** контекстного меню.

Для удаления строк или столбцов таблицы нужно их выделить и выполнить команды **Delete Row** или **Delete Column** соответственно. Если выделения нет, а курсор находится в ячейке таблицы, то будет удалена строка или столбец, которым принадлежит текущая ячейка. Так же для удаления можно использовать клавишу **Delete**.

Для вставки строки или столбца предназначены команды **Insert Row** и **Insert Column** (добавляется одна строка или столбец перед выделенной текущей ячейкой). Когда требуется вставить сразу несколько строк или столбцов, удобнее всего использовать команду **Insert Rows or Columns ...** В открывшемся диалоговом окне (рис. 5.12) необходимо указать, что требуется добавить – строки или столбцы (параметр **Insert**) и их количество (**Number of ...**). Можно указать куда необходимо их добавить – до (**Above the Selection**) или после (**Below the Selection**) текущей.

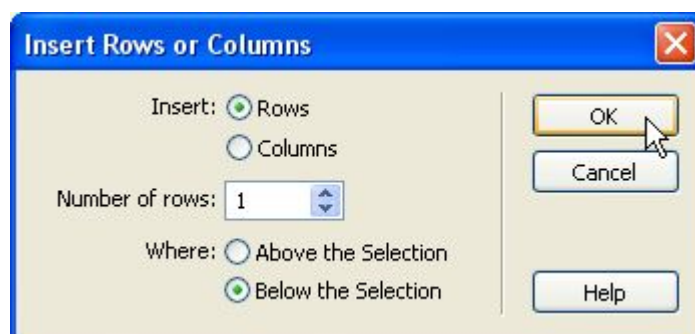


Рис. 5.12. Параметры вставки строк или столбцов таблицы

В редакторе Dreamweaver существует возможность создания таблиц на основе данных из файла. Данные, подготовленные в другом приложении (например, Microsoft Excel) должны быть предварительно сохранены в текстовый файл с разделителями (символами табуляции, запятыми, точками с запятыми и т.п.), после чего их можно будет импортировать в HTML-таблицу.



Рис. 5.13. Вставка табличных данных из файла

Сначала необходимо подготовить указанный файл. Затем, в Dreamweaver нажать кнопку **Import Tabular Data** на закладке **Data** панели объектов (рис. 5.13) или выполнить команду **Insert=>Table Objects=>Import Tabular Data**.

В открывшемся диалоговом окне (рис. 5.14) необходимо указать путь к файлу с данными (поле **Data file**). В списке **Delimiters** выбрать тип разделителя, использованного в файле с данными. Если выбрать пункт **Other**, то в дополнительном поле можно указать нестандартный разделитель (который, вообще говоря, может быть любым символом). Параметр **Table width** позволяет задать ширину таблицы и может принимать одно из следующих значений: **Fit to Data** – ширина колонок таблицы будет выбрана автоматически таким образом, чтобы уместить самую длинную строку, и **Set** – позволяет указать ширину таблицы в процентах по отношению к ширине окна браузера или в пикселах. В списке **Format Top Row** можно выбрать один из четырех вариантов оформления верхней строки таблицы: **No Formatting** – не форматировать, **Bold** – выделить жирным шрифтом, **Italic** – выделить курсивом или **Bold Italic** – выделить жирным курсивом. Параметры **Cell Padding**, **Cell Spacing** и **Border** имеют тот же смысл, что и в обычных таблицах.

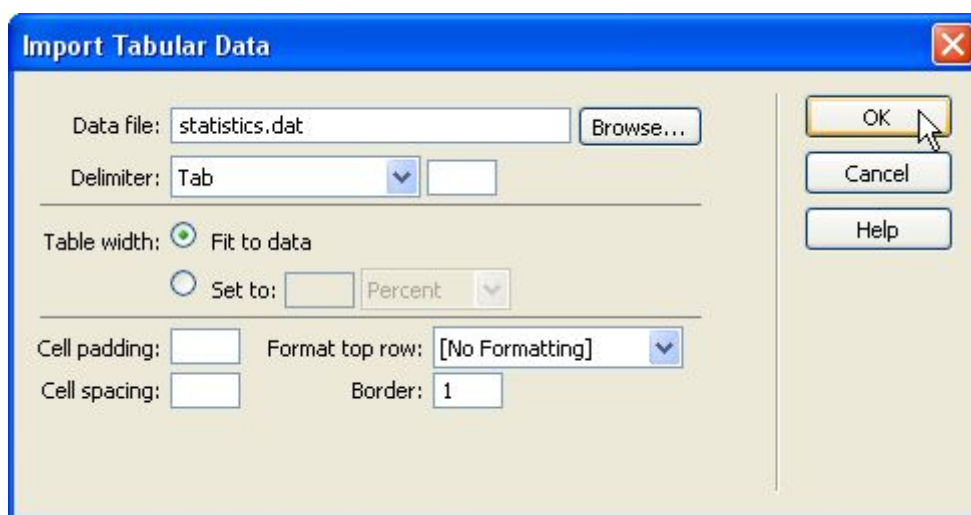


Рис. 5.14. Создание таблицы на основе данных из файла

5.6 Создание и использование стилей CSS

Каскадные таблицы стилей (CSS, Cascade Style Sheet) являются достаточно новой технологией в web-дизайне. Являясь аналогами стилей в программах верстки и мощных текстовых процессорах, они позволяют авторам

использовать типографские приемы определения представления и размещения к элементам на web-странице.

Основным препятствием для широкого использования стилей CSS является проблема поддержки со стороны браузеров. С выходом новых версий ситуация изменяется в положительную сторону, поэтому при надлежащем тестировании продуманное использование стилей CSS является оптимальным способом определения внешнего представления web-страницы.

Таблицы стилей состоят из набора правил, описывающих формат представления элементов документа. Правило представляет собой объявление способа отображения элементов страницы, и записывается в следующем формате:

Селектор {правило: значение; правило: значение; ...}

Селектор служит для идентификации элемента, на который будет оказывать воздействие правило. В фигурных скобках перечисляются команды управления стилем или отображением этого элемента. Они состоят из свойства и значения, отделенных друг от друга символом двоеточия, за которым следует пробел. Свойство – это параметр стиля, который может быть задан (например, гарнитура шрифта, цвет, размер и т.п.). Описание стиля может содержать несколько пар свойство/значение, отделенных друг от друга точкой с запятой.

Следующий пример содержит правило для всех заголовков документа 1-го уровня. Они будут выведены шрифтом размером 14 пунктов, гарнитура Verdana или sans-serif и красным цветом.

```
H1 { font-size: 14pt;  
      font-face: Verdana, sans-serif;  
      color: red;  
}
```

В Dreamweaver для работы со стилями CSS предназначена панель стилей (рис. 5.15). Для ее включения нужно выполнить команду **Window=>CSS Styles**.

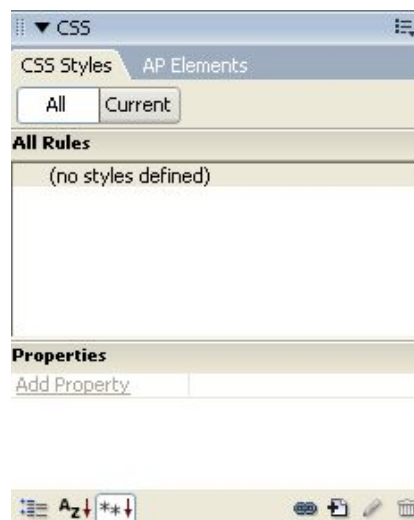


Рис. 5.15. Панель стилей CSS

Dreamweaver может работать с тремя типами каскадных таблиц стилей:

- стили, переопределяющие действие того или иного тега HTML (как в приведенном выше примере);
- индивидуальные стили на основе идентифицирующих атрибутов. Они подобны стилям, используемым в традиционных текстовых процессорах (например, Microsoft Word). В отличие от приведенного выше примера здесь в качестве селектора используется ключевое слово – имя класса. Такой стиль может быть применен к различным блокам текста или другим элементам страницы;
- стили переопределяющие воздействие определенной комбинации тегов или тегов содержащих специальный атрибут-идентификатор.

Для создания нового стиля нужно нажать кнопку **New CSS Rule** на панели стилей или выбрать команду **New...** из контекстного меню, появляющегося при щелчке правой кнопкой мыши в любом месте панели стилей.

В открывшемся диалоговом окне (рис. 5.16) необходимо в первую очередь указать тип создаваемого стиля – список **Selector Type**. **Class** создает именованный стиль, который может быть применен к различным блокам текста. При этом в поле **Name** необходимо указать имя, идентификатор (CLASS) стиля. Оно должно начинаться с символа точка, являющегося идентификатором класса. **Tag** переопределяет параметры форматирования определенного тега, который необходимо выбрать в списке **Tag**. **Advanced** позволяет определить форматирование для комбинации тегов или для тегов содержащих особый идентификатор. В данном случае в поле **Selector** необходимо указать идентификатор (ID). Идентификаторы используются аналогично именованным стилям, но предназначены для форматирования элементов, требующих индивидуальной обработки. В остальных случаях предпочтительным является использование именованных стилей (CLASS).

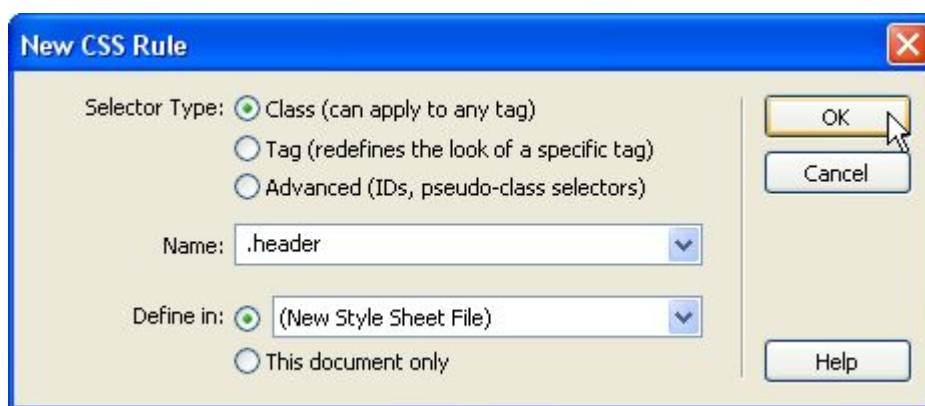


Рис. 5.16. Диалоговое окно создания нового стиля

Далее необходимо определить где будет расположено описание стилей – список **Define in**. При выборе **New Style Sheet File** таблица стилей будет сохранена в отдельном файле, ссылка на который будет вставлена особым образом в заголовок документа (если с редактируемой страницей уже связаны какие-либо файлы с описанием стилей, то их имена будут перечислены в

списке). При таком подходе одна и та же таблица стилей может использоваться в различных документах. Это особенно удобно, когда надо определить элементы форматирования общие для всех страниц сайта. Во втором случае (**This document only**) описание стилей будет сохранено непосредственно в редактируемом документе (и, естественно, может быть использовано только в его пределах). Такой вариант удобен для создания стилей, специфичных для данной страницы.

После того, как все параметры определены можно нажимать **ОК**, после чего откроется диалоговое окно определения параметров нового стиля (рис. 5.17). В левой его части в списке **Category** можно выбирать группы параметров форматирования, которые необходимо определить. Если какие-то из параметров не имеют значения для создаваемого стиля, то соответствующие им поля должны остаться незаполненными.

В первой группе **Type** собраны параметры форматирования шрифта. В поле **Font** можно задать его гарнитуру. Поле **Size** позволяет определить размер шрифта. Он может быть задан числовой величиной, тогда в соседнем поле необходимо указать единицы измерения (наибольший интерес здесь представляют пиксел – **pixels** и традиционная типографская единица измерения пункт – **points**). Кроме того, размер может быть задан при помощи одной из предопределенных относительных единиц измерения или в процентах от унаследованного размера.

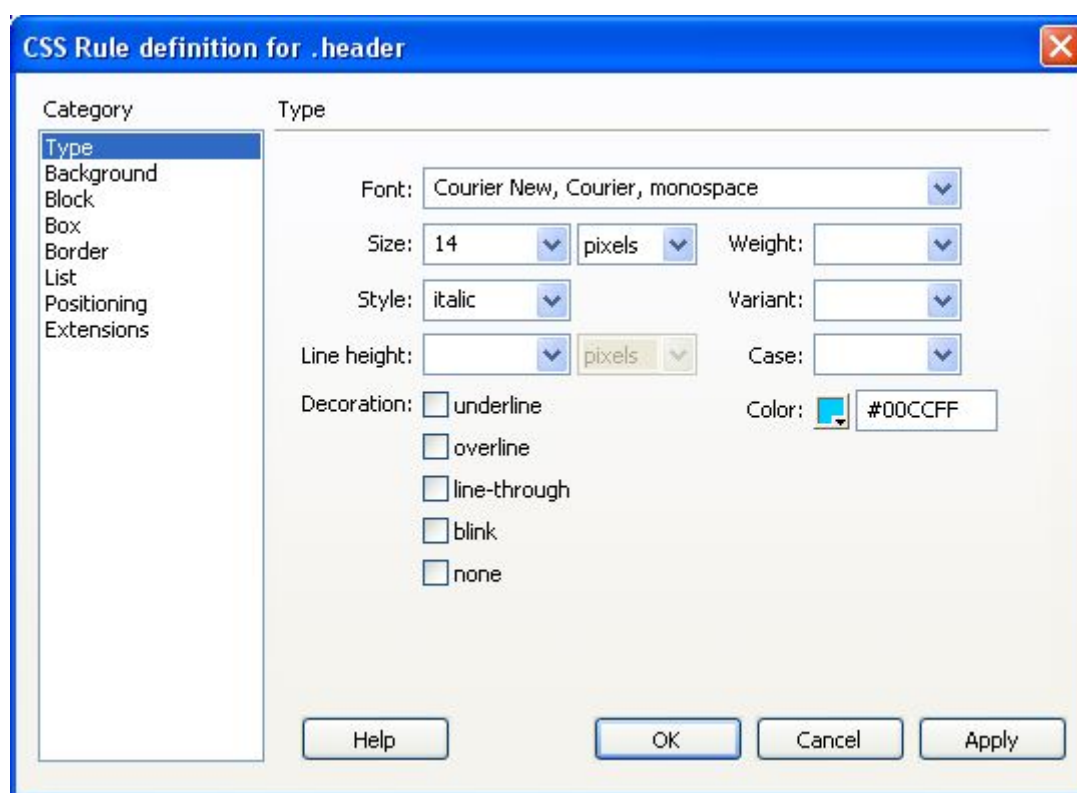


Рис. 5.17. Диалоговое окно определения параметров стиля

В поле **Style** можно выбрать один из вариантов начертания шрифта: **normal** – обычное (используется по умолчанию), **italic** – курсив, **oblique** – наклонное. Параметр **Line height** позволяет задать расстояние между базовыми

линиями смежных строк текста – интерлиньяж. Свойство **Decoration** применяет к тексту подчеркивание (**underline**), надчеркивание (линия над текстом, **overline**), зачеркивание (**line-through**), эффект мигания (**blink**). Параметр **none** используется по умолчанию для обычного текста. Если его применить к гипертекстовой ссылке (для которых по умолчанию используется **underline**), то она не будет подчеркнута.

Обратите внимание, что действие некоторых свойств не будет отображаться в поле документа Dreamweaver, а только в окне браузера.

Параметр **Weight** задает вес, или жирность, шрифта. Свойство **Varint** используется для отображения элемента маленькими прописными буквами (значение **small-caps**). Параметр **Case** оказывает влияние на использование заглавных букв в элементе (**capitalize** – отображает первую букву каждого слова в верхнем регистре, **uppercase** – отображает все символы в верхнем регистре, **lowercase** – отображает все символы в нижнем регистре). Параметр **Color** используется для задания цвета текста.

В группе параметров **Background** собраны свойства фона, применяемые к "холсту" позади элемента. В идеале фон занимает пространство позади содержимого и отступов вокруг него (хотя на практике это не всегда так). **Background color** и **Background image** определяют соответственно цвет фона и фоновое изображение элемента. Если задано фоновое изображение, то свойство **Repeat** определяет как оно будет повторяться (**repeat** – в вертикальном и горизонтальном направлениях, **repeat-x** – только по горизонтали, **repeat-y** – только по вертикали, **no-repeat** – без повторов, один раз). Параметром **Attachment** определяется будет ли прокручиваться фоновое изображение вместе с документом (по умолчанию, **scroll**) или нет (**fixed**). Свойства **Horizontal position** и **Vertical position** определяют начальное положение фонового изображения относительно прямоугольника, окружающего содержимое элемента.

В группе параметров **Block** можно задать выравнивание и взаимное расположение фрагментов текста в элементе. Свойство **Word spacing** задает интервал между словами. **Letter spacing** определяет размер интервала между символами. Параметр **Vertical alignment** влияет на вертикальное выравнивание элемента. **Text align** задает горизонтальное выравнивание текстовых элементов. Свойство **Text indent** определяет размер отступа перед первой строкой текста элемента. Параметр **Whitespace** определяет, каким образом используются пробелы, заданные в исходном документе: **normal** – трактует текст обычным образом, заменяя множественные пробелы на один; **pre** – аналог тега `<pre>` в HTML (но при этом не используется шрифт с фиксированной шириной); **nowrap** – предотвращает перенос элементов текста на следующую строку, если не задан тег `<br>`.

Свойства, собранные в разделе **Box** используются для управления размещением элементов на странице.

Параметры из раздела **Border** позволяют управлять выводом и параметрами рамок вокруг элементов.

В разделе **List** можно задать параметры форматирования списков.

Параметры раздела **Positioning** позволяют преобразовать тег или выделенный блок текста в слой. Использование ограничено тем, что эти свойства поддерживаются только Internet Explorer.

В группе **Extensions** собраны параметры, не поддерживаемые большинством браузеров.

После того, как все необходимые параметры стиля определены, он появится в списке в панели стилей. Выбранный в списке стиль применяется к текущему выделенному элементу документа. Для того, чтобы убрать с элемента воздействие всех стилей, нужно его выделить и выбрать пункт (**none**). Обратите внимание, что в этом списке отображаются только именованные стили. Стили переопределения тегов и стили с идентификаторами автоматически применяются к соответствующим элементам страницы.

Кнопка **Attach Style Sheet** панели стилей позволяет присоединить к текущему редактируемому документу заранее подготовленный файл с описанием стилей.

Для внесения изменений в описание стилей нужно выбрать необходимый стиль в панели **CSS** и нажать кнопку **Edit Style...**, при этом открывается диалоговое окно определения параметров стиля (рис. 5.17).

Удалить стиль можно при помощи команды **Delete** из контекстного меню выбранного стиля. Так же можно удалить выделенный стиль на панели стилей при помощи кнопки **Delete CSS Rule**.

5.7 Импорт текста в формат HTML

Существенной проблемой, особенно часто встающей на начальном этапе разработки web-сайтов, является переработка материалов, подготовленных ранее в различных форматах.

Если информация была сохранена в обычном текстовом файле, то для ее размещения на web-странице проще всего воспользоваться буфером обмена Windows. Для этого необходимо открыть в Dreamweaver необходимый документ (или создать новый). Одновременно с ним нужно открыть текстовый файл, содержащий информацию, предназначенную для размещения на странице (в любом доступном текстовом редакторе). Далее необходимо выделить блок текста, который требуется скопировать, и выбрать команду копирования (обычно **Copy** из меню **Edit**, или комбинация клавиш **Ctrl+C**). В результате выделенный фрагмент текста будет помещен в буфер обмена Windows. Затем нужно перейти в окно Dreamweaver, установить курсор в необходимое место в поле документа и вставить текст из буфера (**Edit=>Paste** или комбинация клавиш **Ctrl+V**). При этом к тексту будет применено соответствующее HTML-форматирование (расставлены теги абзацев, переходов строк и т.п.).

Аналогичным образом возможен перенос текстовых фрагментов из различных источников (например, из окна браузера, редактора и т.п.). Но применение такого подхода становится невозможным, когда требуется перевести в формат HTML документы, содержащие сложное форматирование (параметры шрифта, расположение элементов на странице, встроенные

изображения, таблицы и т.п.). Чаще всего документы такого рода готовятся в формате Microsoft Word.

В Dreamweaver существует возможность импорта документов, сохраненных в Microsoft Word в виде HTML-файлов. Для этого необходимо выбрать команду **File=>Import=>Import Word HTML** и указать файл, который необходимо открыть. Редактор Word имеет встроенные средства создания и редактирования формата HTML. Кроме того, имеется возможность сохранения уже готовых документов в виде web-страниц. Для этого необходимо выполнить команду **Файл=>Сохранить как Web-страницу**. Однако, HTML-код, сгенерированный при таком экспорте, обычно имеет неприемлемо низкое качество. В первую очередь это связано с тем, что в документ добавляется большое количество тегов и стилей, специфичных для программных продуктов фирмы Microsoft, и не несущих необходимой для HTML-представления информации. Особенно большое количество подобных элементов добавляет Word 2000 (в получившемся HTML-файле объем вспомогательной информации может в 3-5 раз превышать объем самой информативной части). И даже при таком подходе невозможно сохранить в HTML-файле некоторые параметры форматирования, присущие Word. Для "чистки" такого кода необходимо выбрать команду **Commands=>Clean Up Word HTML....**

Открывается диалоговое окно параметров "очистки" HTML, сгенерированного Word. При этом по возможности удаляется вся информация о форматировании, свойственная Word.

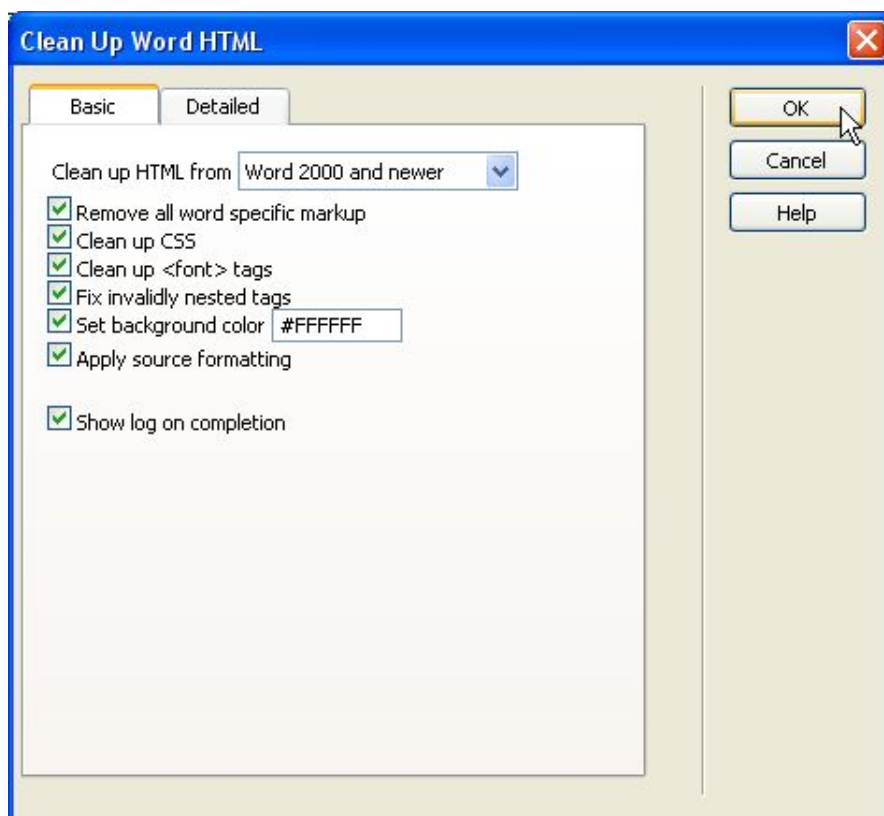


Рис. 5.20. Диалоговое окно параметров "очистки" файла, сохраненного в Microsoft Word в формате HTML

В первую очередь делается попытка определения версии Word, сгенерировавшей файл (это важно, так как каждая из них имеет определенные особенности). Если версия была определена неверно, то ее можно изменить в поле **Clean up HTML from**. Параметр **Remove all word specific markup** позволяет удалить всю разметку HTML, специфичную для Word. На закладке **Detailed** можно настроить некоторые параметры удаления этих элементов. Параметр **Clean up CSS** позволяет удалить все каскадные стили CSS, свойственные Word, и не-CSS стили. **Clean up tags** удаляет лишние теги размера шрифта, преобразуя их к соответствующим стилям HTML. Параметр **Set background color** позволяет задать цвет фона документа. Так как Word генерирует документы без фоновой заливки, то если не задать этот параметр страница при просмотре будет иметь серый фон. Параметр **Apply source formatting** применяет к документу параметры форматирования, заданные в настройках редактора. Если установлен параметр **Show log on completion**, то по окончании преобразования файла будет выдано диалоговое окно, с сообщением о внесенных в документ изменениях.

В зависимости от размера документа и сложности его форматирования преобразование может занимать достаточно продолжительное время. После окончания такой очистки документа в большинстве случаев потребуется дополнительная ручная доводка страницы.

6. Вставка изображений и других элементов

6.1 Использование графических материалов

Важнейшей возможностью, которой обладает язык HTML, является возможность вставки иллюстраций в текстовый документ. Для того, чтобы вставить изображение в редактируемый документ, необходимо нажать кнопку **Images** на панели объектов и в открывшемся списке выбрать **Image** (рис. 6.1) или выполнить команду **Insert=>Image**. В открывшемся диалоговом окне необходимо выбрать графический файл (в формате GIF, JPEG или PNG). Следует обратить внимание на то, что документ перед вставкой графических и большинства других элементов необходимо сохранить, иначе путь к подключаемому файлу может быть абсолютным (начиная от корня диска).

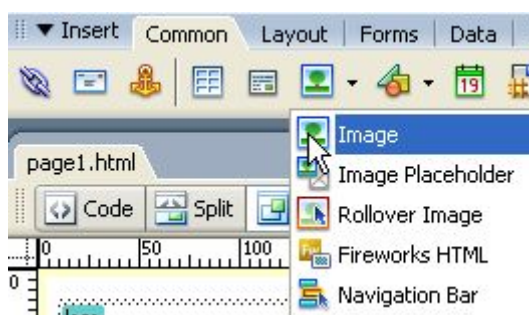


Рис. 6.1. Вставка изображения

Если выделить изображение (для этого достаточно щелкнуть по нему мышью), то на панели свойств будут отображены его свойства (рис. 6.2).



Рис. 6.2. Свойства графического объекта

Параметры **W** и **H** определяют соответственно ширину и высоту изображения. По возможности следует изменять размеры самого графического файла (используя графические редакторы), оставляя параметры **W** и **H** равными исходным размерам. Если одна из этих величин не соответствует истинному значению размеров изображения, она будет выделена жирным шрифтом. Для того, чтобы вернуть параметрам их действительные значения, достаточно щелкнуть мышкой по подписи к полю (буквам **W** или **H**) или воспользоваться кнопкой **Reset Size**.

В поле **Src** указан путь к графическому файлу: при необходимости его можно отредактировать вручную или выбрать файл заново, нажав расположенную рядом кнопку **Browse for File** . Параметр **Align** задает

выравнивание графического объекта относительно других элементов абзаца или строки. Возможны следующие варианты выравнивания:

- **Browser Default** – выравнивание по умолчанию (обычно **Baseline**, но может отличаться в зависимости от настроек браузера клиента);
- **Baseline** и **Bottom** – базовая линия текста (или другого объекта) будет выровнена по нижнему краю изображения;
- **Top** – верхняя граница изображения будет выровнена по верхнему краю самого высокого элемента в текущей строке;
- **Middle** – выравнивание базовой линии текста по середине изображения;
- **Text Top** – верхняя граница изображения будет выровнена по самому высокому текстовому элементу текущей строки;
- **Absolute Middle** – середина изображения будет выровнена по середине строки;
- **Absolute Bottom** – выравнивание нижнего края самого “низкого” символа по нижнему краю изображения;
- **Left** – располагает изображение у левого края, при этом текст обтекает его с правой стороны. Если предшествующий текст также имел выравнивание по левому краю, объект будет перенесен на следующую строку;
- **Right** – располагает изображение у правого края, при этом текст обтекает его с левой стороны. Если предшествующий текст также имел выравнивание по правому краю, объект будет перенесен на следующую строку.

Параметр **V Space** позволяет задать области пустого пространства над и под изображением (размер указывается в пикселах). Аналогично, **H Space** задает размер свободного пространства слева и справа от изображения.

Параметр **Link** позволяет создать гипертекстовую ссылку с данного графического элемента. При этом вокруг изображения появляется рамка, которая отображается цветом ссылок. Толщину этой рамки определяет параметр **Border** (если указать 0, то рамки не будет). В поле **Alt** следует *альтернативный текст*, который будет отображаться вместо картинок в текстовых браузерах и при отключенной автоматической загрузке изображений. Кроме того, этот текст будет выведен в качестве всплывающей подсказки при наведении указателя мыши на объект.

При нажатии кнопки **Edit** для редактирования изображения будет открыт графический редактор, выбранный в настройках Dreamweaver.

6.2 Создание карты ссылок на изображении

Для организации ссылок с изображения возможно использование так называемых *карт изображений*. При этом ссылка создается не со всего изображения в целом, а с отдельных областей (одной или нескольких).

Для создания карты ссылок необходимо выделить на изображении *активные области (hotspot)*. В Dreamweaver можно создавать области трех видов – прямоугольные , круглые  и произвольной конфигурации . Для этого нужно нажать одну из кнопок и при помощи мыши выделить необходимую область на изображении. Параметры активной области будут отображены на панели свойств (Рис 6.3).

Поле **Map** указывает имя карты, к которой относится данная область. Поля **Link** и **Target** задают, как обычно, параметры ссылки, а поле **Alt** – альтернативный текст. Для добавления дополнительных областей нужно выбрать необходимый инструмент выделения и повторить описанные действия. Созданные области можно перемещать по изображению, удерживая левую кнопку мыши.

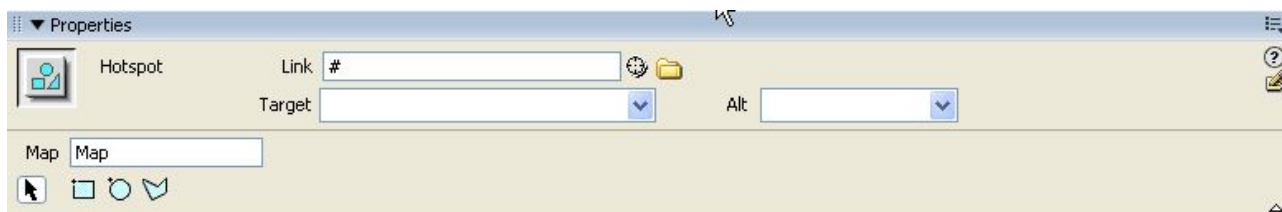


Рис. 6.3. Параметры активной области на карте изображения

Созданные ранее активные области можно редактировать: передвигать и изменять их размеры. Так же можно копировать изображения с активными областями из одного документа в другой – их параметры при этом сохраняются. Можно копировать активные области с одного изображения на другое. Для выполнения какой-либо операции над областью необходимо выделить ее. Для этого предназначен инструмент **Pointer Hotspot Tool**.

6.3 Вставка изменяющегося изображения

Для создания изображений, изменяющих свой вид при прохождении курсора мыши, используется скрипт **Rollover Image**.

Для вставки изменяющегося изображения необходимо кнопку **Images** на панели объектов и в открывшемся списке выбрать **Rollover Image** (рис. 6.1) или выполнить команду **Insert=>Image Objects=>Rollover Image**.

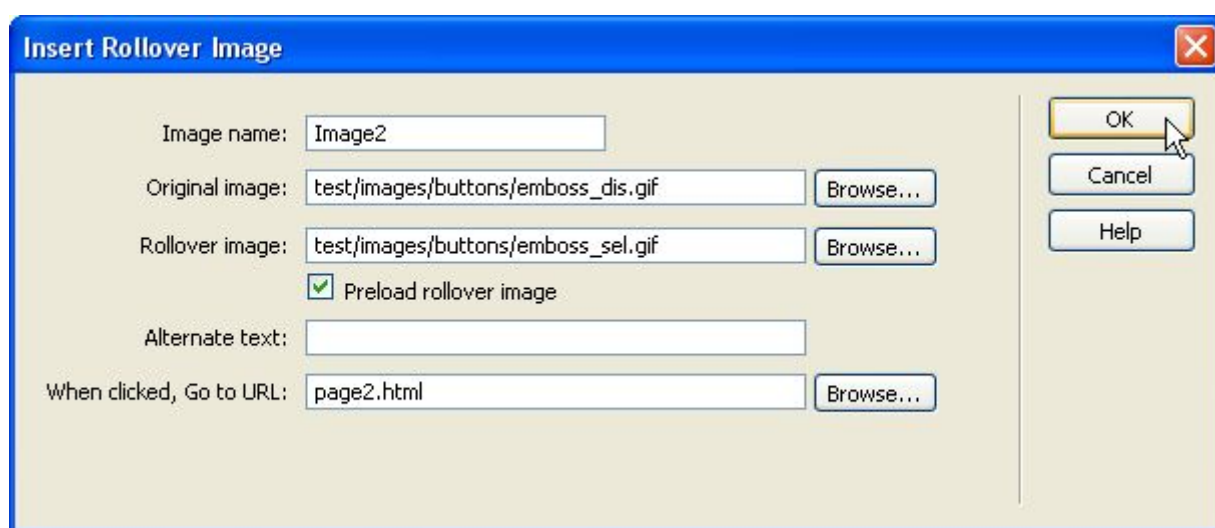


Рис. 6.4. Диалоговое окно для настройки скрипта Rollover Image

В диалоговом окне (рис. 6.4) необходимо указать имена файлов, которые будут использоваться при работе изменяющегося изображения: **Original image** – путь к начальному файлу с изображением, **Rollover image** – путь к

файлу с изображением, которое будет появляться, когда курсор мыши окажется над областью изображения.

Имя интерактивного элемента (**Image name**) используется при работе скриптов. Если не предусматривается взаимодействие с этим объектом или изменение алгоритмов работы скриптов, то имя можно оставить заданным по умолчанию.

Если включить флажок **Preload rollover image**, то в HTML-документ будет вставлен скрипт для предварительной загрузки файлов с изображениями в кэш-память браузера.

Изменяющееся изображение можно сделать ссылкой, если в поле **When clicked, Go to URL** указать адрес Internet-ресурса.

6.4 Определение интерактивных свойств объектов

Практически любой объект HTML-документа можно сделать интерактивным. Любой участок текста, любое изображение, ячейки таблиц и слои могут изменять свои свойства при выполнении учеником каких-либо действий на клавиатуре или мышью. Изменение свойств объектов осуществляется с помощью программ на языке JavaScript. Благодаря тесной интеграции с форматом HTML, программы на языке JavaScript могут получить доступ к свойствам всех объектов HTML-страницы. А использование технологии событийного программирования позволяет отслеживать события, происходящие на HTML-странице, и выполнять те или иные действия.

HTML-редактор Dreamweaver с помощью специального инструмента **Behaviors** позволяет управлять поведением объектов в зависимости от происходящих на странице событий, а также предоставляет набор готовых, часто используемых скриптов.

Вызов окна инструмента **Behaviors** выполняется по команде **Window=>Behaviors**.

Выделив объекты, поведение которых будет изменяться в зависимости от происходящих событий, в окне **Behaviors** (рис. 6.5) необходимо указать, какие действия (Actions) будут выполняться после возникновения того или иного события (Events). Параметры управления поведением элемента страницы будут, в конечном итоге, установлены для соответствующего тега HTML. Имя текущего выделенного в поле документа тега отображается в заголовке окна **Behaviors** (рис. 6.5). Если для одного и того же события было назначено несколько реакций, то они перечисляются в порядке своего выполнения.

Кнопку  следует использовать для назначения действия, применимого к выделенному объекту, а кнопку  – для отмены назначенного ранее действия. Порядок выполнения действий можно изменить с помощью кнопок  и .

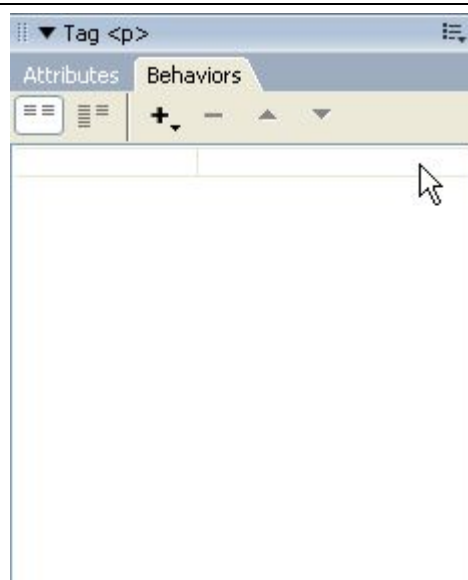


Рис. 6.5. Окно для определения поведения объектов и назначения реакции на события

Список основных заранее предусмотренных действий перечислен в таблице 6.1. Список основных событий, после которых можно назначить выполнение определенных действий, перечислен в таблице 6.2. Рассмотрим назначение некоторых из них.

Таблица 6.1 Список основных действий

Call JavaScript	вызов любой программы на JavaScript
Change Property	изменение свойств любого объекта
Check Browser	проверка версии браузера
Check Plugin	проверка наличия Plugin на Web-клиенте
Drag Layer	перемещение слоя
Go to URL	переход к другому файлу или Internet-ресурсу
Open Browser Window	создание нового окна с заданными параметрами
Play Sound	вставка в документ звуковых фрагментов
Popup Message	вывод окна с сообщением
Preload Images	предварительная загрузка изображений в кэш-память браузера
Set Nav Bar Image	изменение свойств навигационной панели
Set Text of Frame	динамический вывод HTML-текста во фрейм
Set Text of Layer	динамический вывод HTML-текста в слой
Set Text of Status Bar	динамический вывод текста в строку состояния
Set Text of Text Field	динамический вывод текста в текстовое поле
Show-Hide Layers	показать/скрыть слои
Swap Image	изменить изображение
Swap Image Restore	восстановить измененное изображение
Validate Form	проверка соответствия типа данных, введенных в поля формы

Таблица 6.2 Список основных обрабатываемых событий

События HTML-документа

onLoad	Загрузка документа завершилась
onAbort	Пользователь остановил загрузку документа
onError	Ошибка при загрузке документа
onUnload	Пользователь завершил просмотр и переходит к другому документу

События окна броузера

onResize	Размеры окна или фрейма изменились
onMove	окно или фрейм перемещается по экрану
onScroll	Пользователь прокручивает документ
OnHelp	Пользователь выбрал пункт меню Help в окне броузера

События на клавиатуре

OnKeyDown	Пользователь нажал клавишу на клавиатуре
OnKeyUp	Пользователь отпустил клавишу на клавиатуре
OnKeyPress	Пользователь нажал и отпустил клавишу на клавиатуре

События на мыши

OnClick	над объектом сделан щелчок кнопкой мыши
OnDblClick	над объектом сделан двойной щелчок кнопкой мыши
OnMouseDown	над объектом нажата кнопка мыши
OnMouseUp	Кнопка мыши отпущена над объектом
OnMouseMove	курсор мыши движется над объектом
OnMouseOut	курсор мыши вышел за границы объекта
OnMouseOver	курсор мыши вошел в пределы объекта

События на элементах управления

OnReset	Очистка значений элементов формы по кнопке Reset
OnSubmit	Подтверждение значений полей элементов формы по кнопке Submit
OnBlur	Элемент управления теряет фокус
OnFocus	Элемент управления получает фокус
OnChange	Изменилось содержимое элемента управления
OnSelect	Пользователь выделил участок текста

Действие Go To URL

Для выполнения переходов не по гипертекстовым ссылкам, а по определенным событиям можно использовать действие **Go To URL**. При назначении этого действия необходимо заполнить диалоговое окно, представленное на рисунке 6.6.

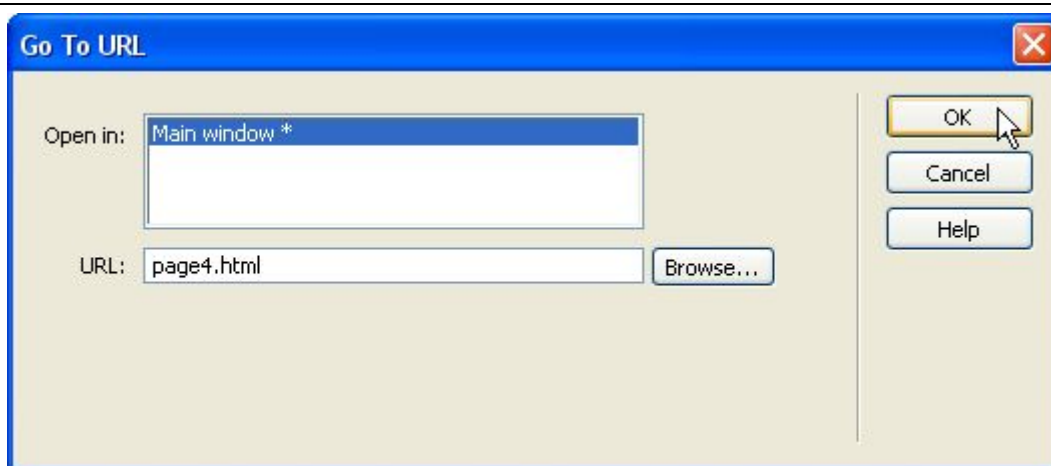


Рис. 6.6. Диалоговое окно для настройки скрипта Go To URL

В списке **Open in** необходимо выбрать имя окна или фрейма (по умолчанию в этом же окне **Main window**), в котором открывать Internet-ресурс, расположенный по адресу, указанному в поле **URL**.

Действие Open Browser Window

Скрипт **Open Browser Window** предназначен для создания нового окна, в которое будет загружен указанный Internet-ресурс. Диалоговое окно для настройки работы этого скрипта представлено на рисунке 6.7.

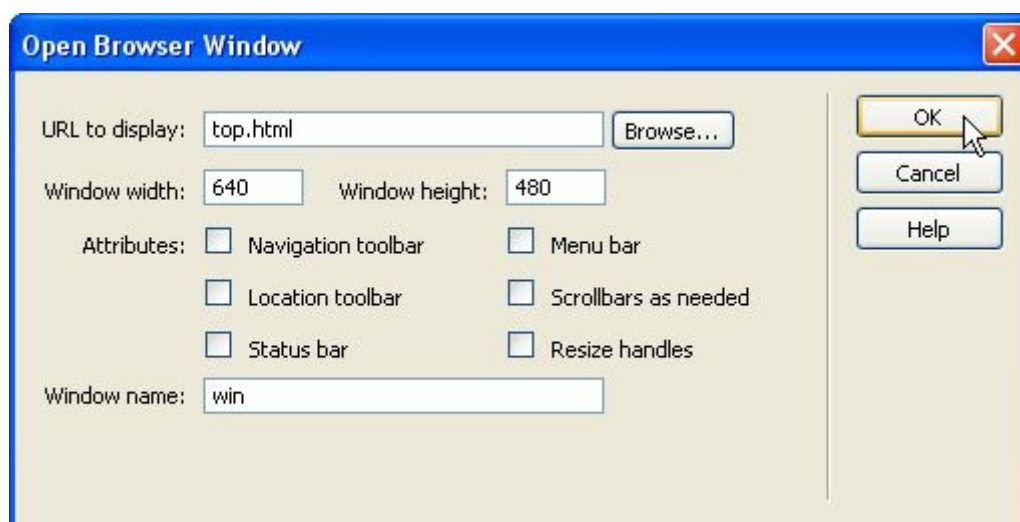


Рис. 6.7. Диалоговое окно для настройки скрипта Open Browser Window

В поле **URL to display** задается адрес Internet-ресурса или имя файла, который должен быть открыт в новом окне броузера (путь к файлу можно также указать по кнопке **Browse...**). В полях **Window width** и **Window height** задаются ширина и высота нового окна. Группа элементов управления **Attributes** определяет наличие у нового окна навигационной панели инструментов (**Navigation toolbar**), меню (**Menu bar**), строка адреса (**Location toolbar**), полос прокрутки (**Scrollbar as needed**), строки состояния (**Status bar**). Флажок **Resize handles** определяет возможность изменения размеров нового

окна браузера. В поле **Windows name** можно указать имя окна для использования в других скриптах.

Действие Play Sound

Скрипт **Play Sound** позволяет вставить в документ звуковые фрагменты. На диалоговом окне для настройки скрипта (рис. 6.8) в поле **Play sound** необходимо указать имя звукового файла. Путь звуковому файлу можно указать с использованием кнопки **Browse...**

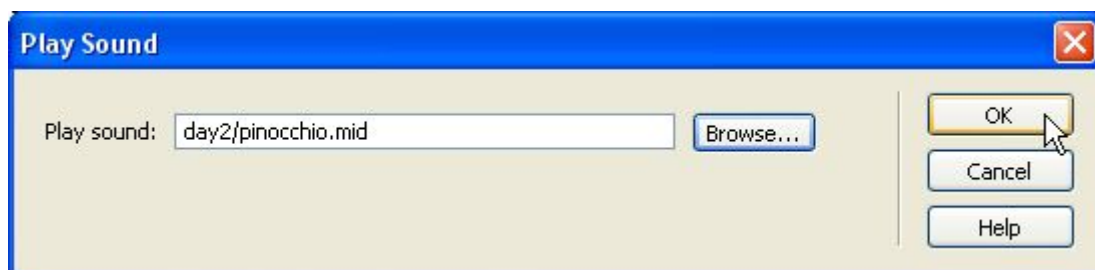


Рис. 6.8. Диалоговое окно для настройки скрипта Play Sound

Действие Popup Message

Для вывода небольшого диалогового окна с сообщением можно воспользоваться готовым скриптом **Popup Message**.

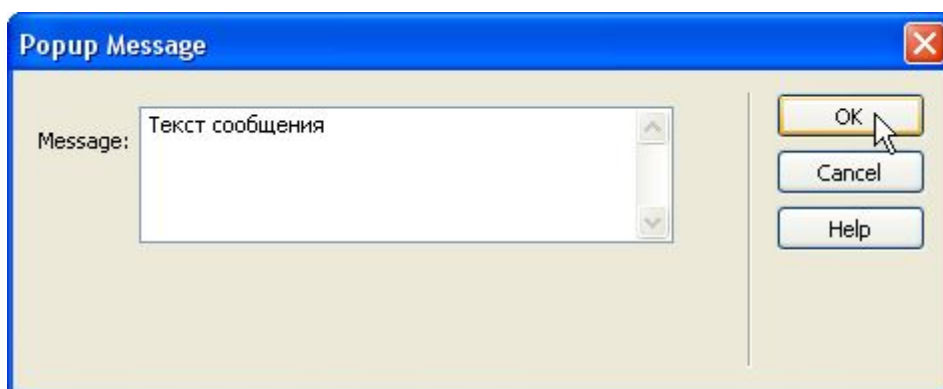


Рис. 6.9. Диалоговое окно для настройки скрипта Popup Message

Для настройки этого скрипта на диалоговом окне (рис. 6.9) в поле **Message** необходимо ввести текст сообщения, который будет отображаться при выводе окна.

6.5 Вставка Flash-кнопок

Для быстрой вставки в HTML-документы заранее заготовленных Flash-кнопок необходимо выполнить команду **Insert=>Media=>Flash Button**. При этом откроется диалоговое окно настройки параметров Flash-кнопки (рис. 6.10).

Из списка **Style** можно выбрать любую из подготовленных разработчиками Dreamweaver кнопок, внешний вид которых отображается в верхней части диалогового окна. Текст, написанный в поле **Button text**, будет размещен на кнопке (в существующих версиях Dreamweaver возможно использование

только латинских букв). Гарнитуру и размер шрифта можно изменить используя элементы управления **Font** и **Size** соответственно. Кнопку можно превратить в ссылку, если в поле **Link** указать имя файла или URL для перехода. Цвет фона можно изменить с использованием элемента управления **Bg color**.

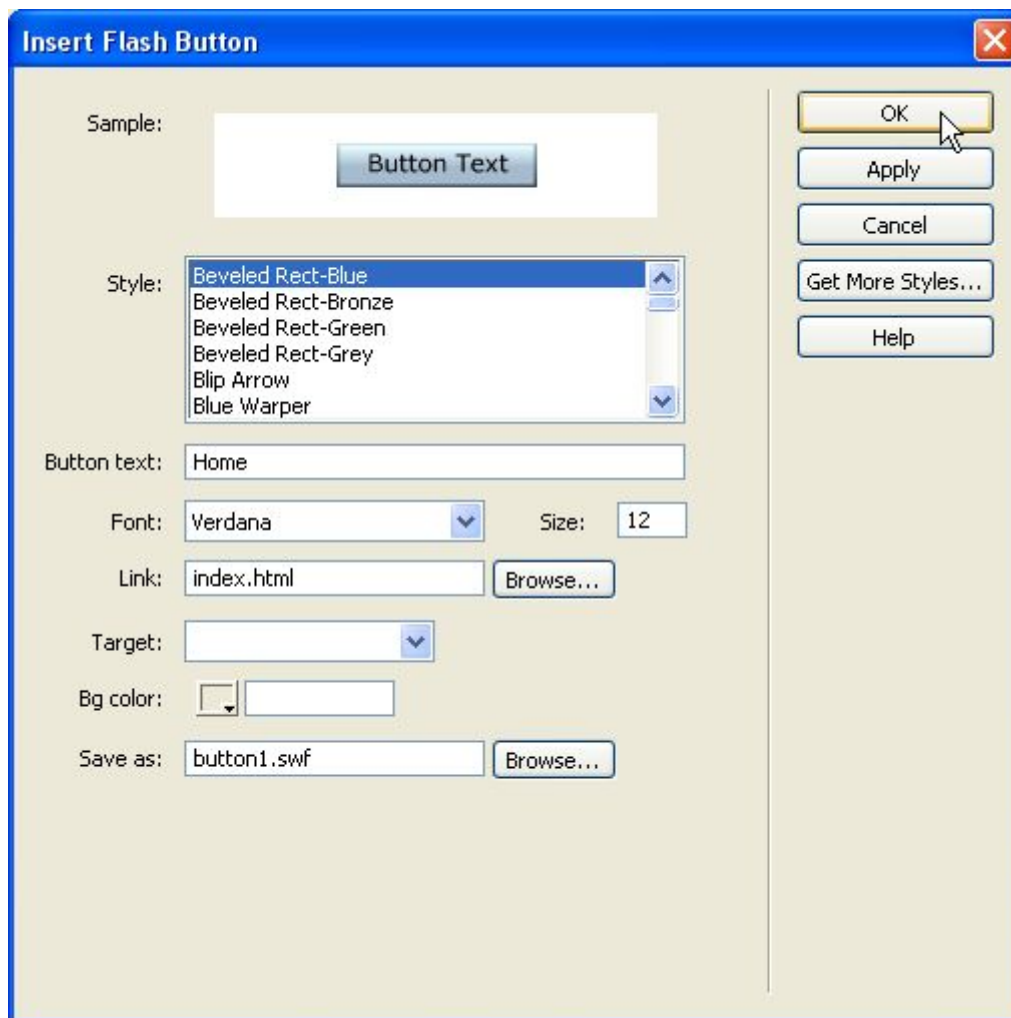


Рис. 6.10. Диалоговое окно для настройки Flash-кнопки

Кнопку, настроенную таким образом, следует сохранить в файле, имя которого можно указать в поле **Save as**.



Рис. 6.11. Панель свойств Flash-кнопки

Параметры настройки кнопки можно изменить в любой момент. Для этого в редактируемом HTML-документе следует выбрать Flash-кнопку, свойства

которой необходимо изменить. Окно настройки параметров Flash-кнопки можно вызвать по кнопке **Edit** на панели свойств (рис. 6.11).

Аналогично, используя инструмент **Insert=>Media=>Flash Text**, в HTML-документ можно добавить и настроить Flash-текст.

На панели объектов расположен ряд кнопок, при помощи которых в web-страницы можно добавлять различные интерактивные объекты, подготовленные с использованием других программных продуктов фирмы Adobe, например:

-  – Flash (анимационные и интерактивные векторные изображения);
-  – Shockwave (интерактивные мультимедийные элементы).

6.6 Использование подключаемых модулей

Одним из важнейших свойств Internet-технологий, обеспечивающих их использование в обучении, является возможность организации интерактивного взаимодействия ученика с обучающими материалами. Виртуальные лабораторные работы, обучающие программы, тестовые системы, системы навигации, поиска и создания персональных закладок в электронных учебниках – все это может быть реализовано только с использованием интерактивных возможностей компьютера. Процесс обучения превращается из пассивного в активный, а кроме теоретических знаний ученики смогут получить практические навыки. Безусловно, это существенно повышает эффективность использования обучающих ресурсов.

Основные возможности языка HTML ограничены логической разметкой текстовых документов. Основные возможности браузера ограничены интерпретацией HTML-документа и его отображением на экране компьютера. Таким образом, интерактивное взаимодействие ученика с обучающими материалами невозможно реализовать без использования программных модулей.

Одним из наиболее простых способов, позволяющих сделать обучающие материалы динамическими и интерактивными, является использование *подключаемых модулей* (plug-ins). Эта технология позволяет включать в HTML-документ не только изображения в формате GIF и JPEG, но и информацию в других форматах. Технология подключаемых модулей подразумевает наличие двух компонентов: объекта (файла в определенном формате) и плеера (программы, которая интерпретирует информацию в том или ином формате). Плеер необходимо скачать и установить на клиентской машине. Теперь, если в HTML-документе встретится встроенный объект, браузер использует плеер для его интерпретации и отображения.

С помощью подключаемых модулей можно дополнить информационное содержание страницы звуковыми и видео иллюстрациями, информацией в других форматах, а также придать материалам интерактивность, используя вставки Flash-объектов или трехмерные интерактивные изображения в формате VRML. Наиболее популярные форматы представления информации и плееры, необходимые для их отображения в окне браузера, перечислены в таблице 6.3.

Таблица 6.3 Форматы, отображаемые с использованием подключаемых модулей (plug-ins)

Анимационные и интерактивные изображения

*.swf	Flash Player (Adobe)
-------	----------------------

Форматы документов

*.pdf	Acrobat Reader (Adobe)
*.doc, *.rtf	Word Viewer (Microsoft)

Аудио - и видеоформаты

*.rpm	RealPlayer (RealNetworks)
*.aam, *.dir	Shockwave (Adobe)
*.mp3, *.wav, *.mid	Winamp MP3 Player (Nullsoft)
*.mpeg, *.wav, *.mid, *.png, *.tiff, *.bmp, *.avi	QuickTime (Apple)

Презентации

*.pqf	Corel Presentations Show It! (Corel)
*.ppz, *.ppt, *.pps, *.pot	PowerPoint (Microsoft)

Трехмерные интерактивные изображения

*.vrmf	Cosmo Player (Silicon Graphics) Cortona (ParallelGraphics)
*.tar, *.gz, *.arj, *.zip, *.lha, *.taz	Netzip Classic (RealNetworks)

HTML-редактор Dreamweaver позволяет вставить в HTML-документ файл, который должен обрабатываться подключаемым модулем. Для этого можно выполнить команду **Insert=>Media=>Plugin**.

7. Создание тестовых заданий с использованием дополнения CourseBuilder

Пакет JavaScript-программ CourseBuilder является дополнением к HTML-редактору Dreamweaver и предназначен для создания на HTML-страницах тестовых заданий и других интерактивных элементов. Теоретические вопросы разработки систем тестирования рассматриваются в пособии “Создание образовательных ресурсов на основе Internet-технологий”.

7.1 Вставка элементов CourseBuilder

Для того, чтобы разместить интерактивный элемент CourseBuilder на HTML-странице, необходимо выполнить команду **Insert=>CourseBuilder Interaction** или воспользоваться кнопкой на закладке **CourseBuilder Interaction** панели объектов (рис. 7.1).

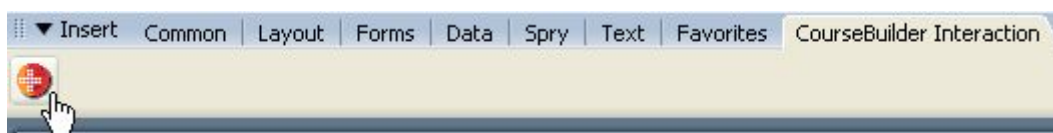


Рис. 7.1. Вставка интерактивного элемента CourseBuilder

Интерактивные элементы CourseBuilder можно размещать в любом месте HTML-документа. Если HTML-документ еще ни разу не был сохранен, то CourseBuilder предложит присвоить ему имя и сохранить. Чтобы избежать путаницы между уже существующими файлами и файлами, созданными CourseBuilder, документы, которые будут содержать тестовые задания, желательно сохранять в отдельной папке (например, папке **test**).

В папку с тестовыми заданиями CourseBuilder должен скопировать необходимые ему изображения и JavaScript-программы. CourseBuilder об этом обязательно предупредит с помощью диалогового окна (рис. 7.2).

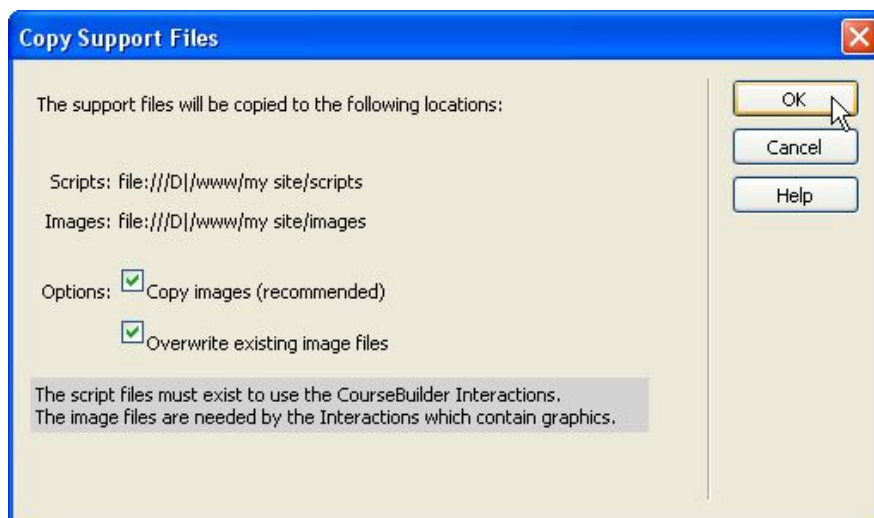


Рис. 7.2. Диалоговое окно с сообщением о копировании файлов

По нажатию кнопки **ОК** необходимые файлы будут скопированы в папку с тестовыми заданиями, и CourseBuilder будет готов к работе: появится основное окно CourseBuilder (рис. 7.3), с помощью которого можно вставить интерактивный элемент и управлять его параметрами.

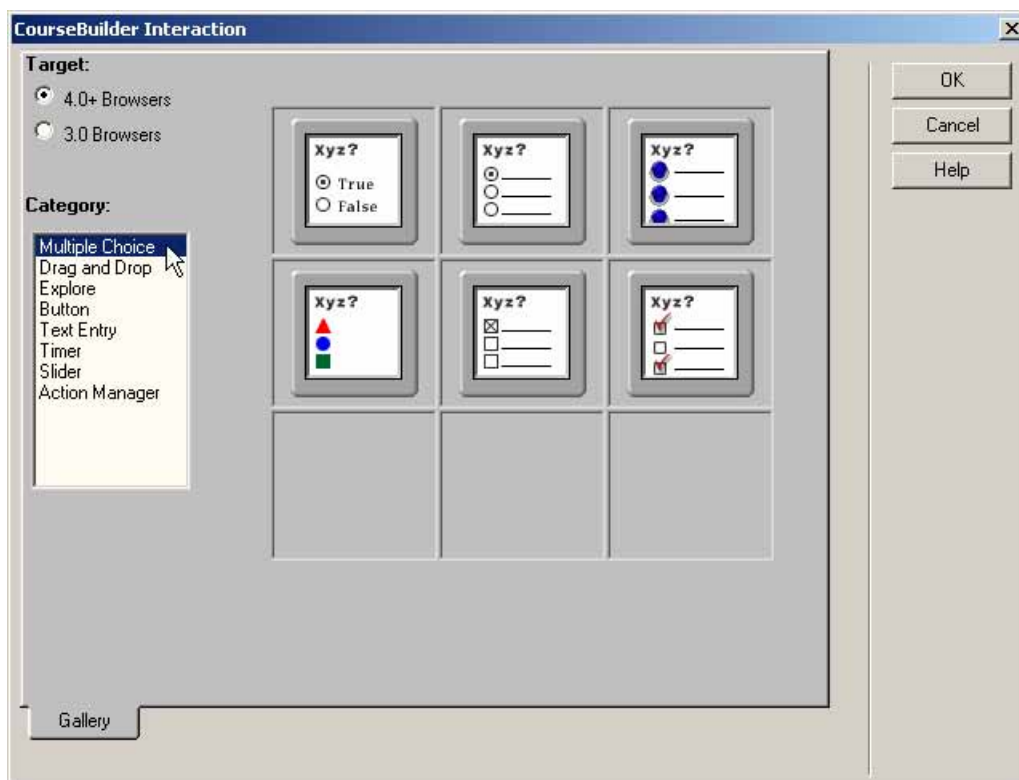


Рис. 7.3. Диалоговое окно CourseBuilder

С помощью этого диалогового окна создается “скелет” тестовых заданий и других интерактивных элементов, которые затем, впрочем, могут быть отредактированы в обычном режиме. При редактировании интерактивных элементов в обычном режиме следует быть осторожным, чтобы не разрушить созданную структуру из текста, элементов управления и невидимых вызовов JavaScript-программ.

Первая закладка диалогового окна CourseBuilder называется **Gallery**. На этой закладке необходимо из списка **Category** выбрать категорию интерактивных элементов и символическое изображение интерактивного элемента, который должен быть вставлен в HTML-документ.

Интерактивные элементы разделены на следующие категории:

- **Multiple-choice Interactions** – элементы с множественным выбором, на основе которых создаются тестовые задания в закрытой форме;
- **Drag-and-drop Interactions** – элементы с перетаскиванием, на основе которых создаются тестовые задания на соответствие и установление правильной последовательности;
- **Explore Interactions** – элементы с интерактивными переходами;
- **Button Interactions** – элемент-кнопка;
- **Text entry Interactions** – элементы с текстовым вводом, на основе которых создаются тестовые задания вопросы в открытой форме;

- **Timer Interactions** – элемент-таймер;
- **Slider Interactions** – элемент-ползунок;
- **Action Manager Interactions** – элемент управления событиями.

При выборе одной из категорий на основном центральном поле диалогового окна CourseBuilder отображаются символические изображения интерактивных элементов. Если выбрать одно из них, то в HTML-документ будет вставлена заготовка интерактивного элемента, которую можно настроить с помощью элементов управления на появившихся дополнительных закладках.

Остановимся подробнее на создании тестовых заданий с использованием интерактивных элементов CourseBuilder.

7.2 Создание тестового задания в закрытой форме

Для создания тестового задания на диалоговом окне CourseBuilder необходимо выбрать категорию **Multiple Choice**. Например, создадим тестовое задание с несколькими правильными ответами на основе интерактивного элемента, который имеет условное имя MultCh_ImageChkboxes (рис. 7.4). Элементы управления для настройки тестового задания расположены на появившихся закладках **General**, **Choices**, **Action Mgr**.

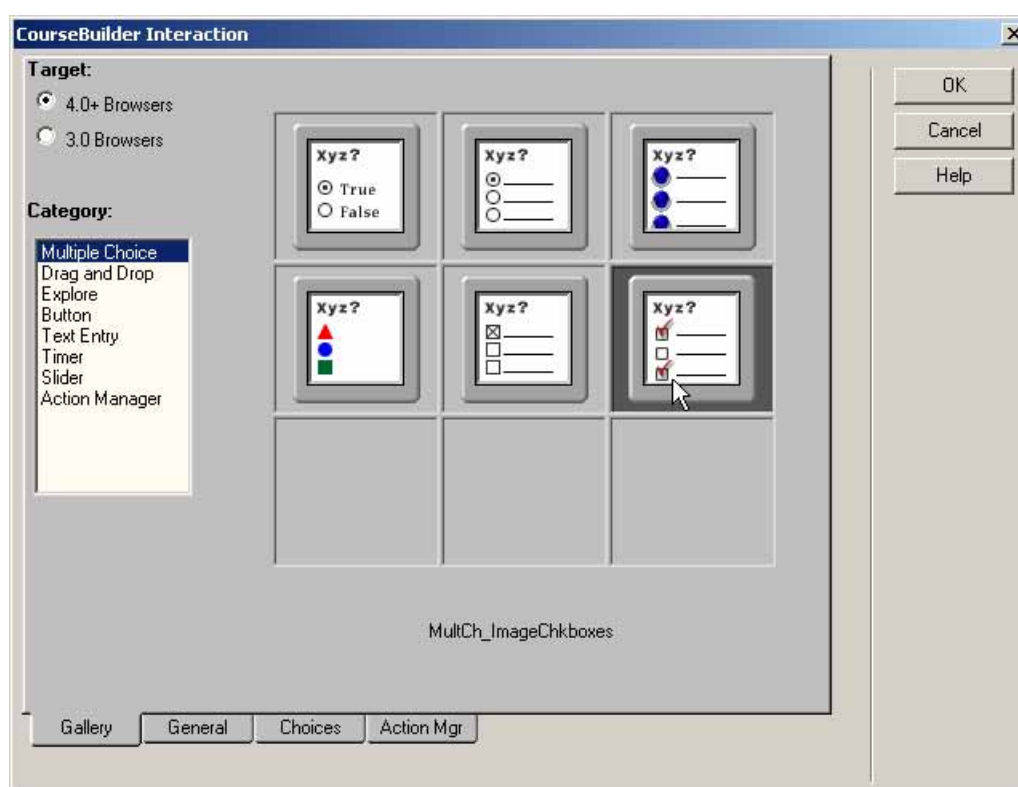


Рис. 7.4. Диалоговое окно CourseBuilder (закладка Gallery)

Параметры, которые определяют правила проверки тестового задания, расположены на закладке **General** (рис. 7.5). Имя интерактивного элемента (**Interaction Name**) CourseBuilder назначает автоматически, и без особой необходимости изменять его не следует. Это имя используется в процессе работы скриптов, поэтому должно состоять из латинских символов.

Формулировка тестового задания записывается в поле **Question Text**, может быть произвольной длины и содержать любые символы.

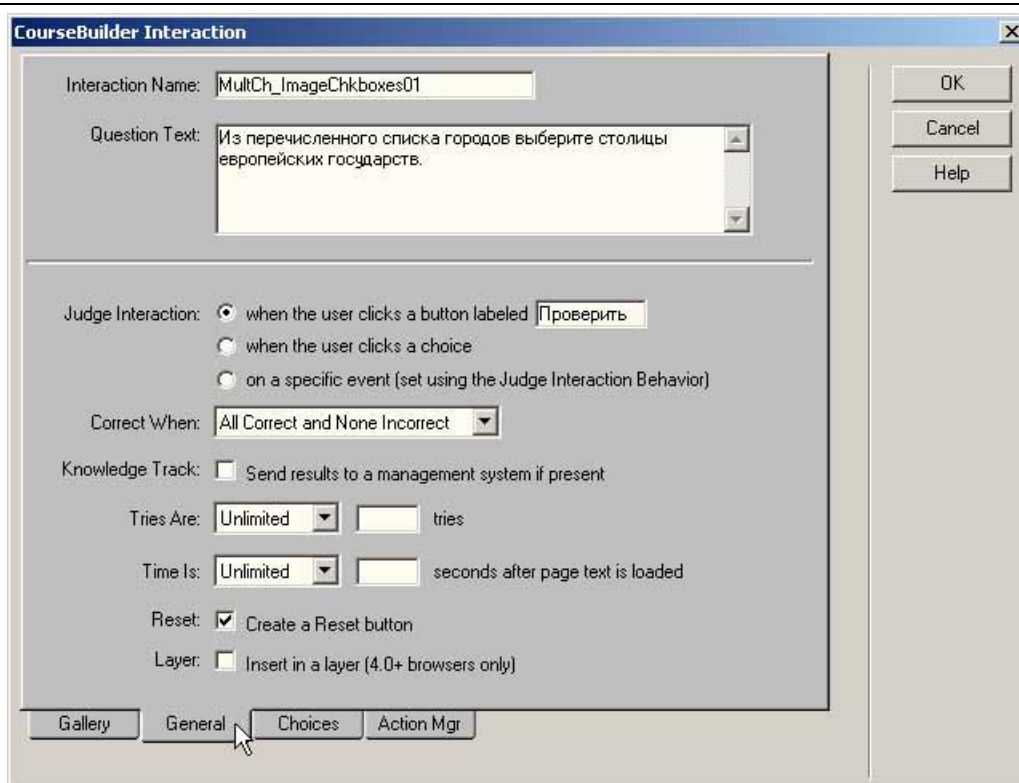


Рис. 7.5. Диалоговое окно CourseBuilder (закладка General)

Группа переключателей **Judge Interaction** определяет событие, по которому будет осуществляться проверка выполнения тестового задания. Если выбран первый переключатель, то проверка будет выполняться по нажатию на специальную кнопку. Надпись на кнопке (по умолчанию “Submit”) можно задать в текстовом поле справа (например, “Ответ готов”). Если выбран второй переключатель, то проверка будет выполняться сразу, как только ученик выберет один из ответов. Если выбран третий переключатель, то проверка будет выполняться по специальному событию, которое необходимо определить дополнительно.

Элемент управления **Correct When** позволяет задать критерий правильно выполненного тестового задания. Если в этом списке выбран пункт All Correct and None Incorrect, то тестовое задание будет считаться выполненным правильно, когда ученик выбрал все альтернативные варианты ответов, помеченные автором как правильные. Если в этом списке выбран пункт Any Correct and None Incorrect, то тестовое задание будет считаться выполненным правильно, когда ученик выберет хотя бы один из альтернативных вариантов ответов, помеченный автором как правильный. Если при выполнении тестового задания ученик выбрал неправильный ответ, то и в том, и в другом случае тестовое задание будет считаться выполненным неправильно.

Флажок **Knowledge Track** можно включить только в том случае, если результаты выполнения тестовых заданий предполагается обрабатывать с помощью специального программного обеспечения, установленного на Web-сервере (например, Lotus Pathware).

Автор тестового задания может ограничить количество попыток и время, отводимое на выполнение тестового задания. По умолчанию количество

попыток и время на выполнение задания неограничено (Unlimited). Для того, чтобы ограничить количество попыток, необходимо в списке **Tries Are** выбрать значение Limited To, а в текстовом поле справа указать количество допустимых попыток (например, 3). Аналогично можно ограничить время выполнения тестового задания. Для этого в списке **Time Is** устанавливается значение Limited To, а в текстовом поле справа указывается предельное время в секундах (например, 180). Если в процессе работы ученика с тестовым заданием закончится количество попыток или отведенное время, то по умолчанию будет выдано предупреждающее сообщение, а затем все элементы данного тестового задания будут заблокированы.

Флажок **Reset** включает в “скелет” тестового задания кнопку для приведения его в исходный вид. Надпись на кнопке можно изменить в обычном режиме редактирования.

Если включить флажок **Layer**, то “скелет” тестового задания будет размещен на отдельном слое.

Элементы управления, расположенные на закладке **Choices** (рис. 7.6) предназначены для создания и изменения параметров вариантов ответов на тестовое задание. Каждый из них должен иметь уникальное имя (**Name**), которое используется JavaScript-программами, и поэтому должно состоять из латинских символов. В списке **Choices** отображаются имена альтернативных вариантов ответов, а в нижней части диалогового окна – их параметры. Изменить количество вариантов альтернативных ответов можно с помощью кнопок **Add** и **Delete**, а поменять порядок их следования – с помощью кнопок **Up** и **Down**.

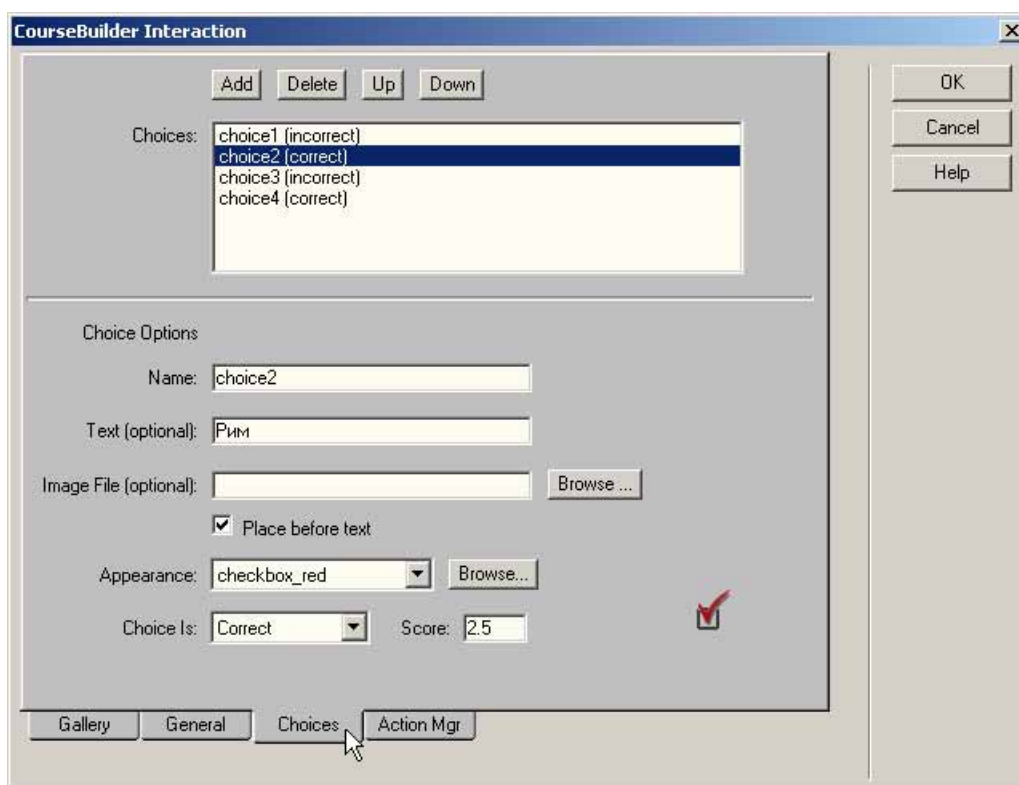


Рис. 7.6. Диалоговое окно CourseBuilder (закладка Choices)

Последовательно выбирая имена альтернативных вариантов ответов из списка **Choices**, необходимо ввести формулировку каждого варианта ответа (в поле **Text**) или его графическое представление (имя файла с изображением в поле **Image File**). Если заполнено и поле **Text**, и поле **Image File**, то формулировка альтернативного варианта ответа будет содержать и текст, и изображение. В этом случае флажок **Place before text** определяет порядок следования изображения. Если флажок включен, то изображение будет размещаться до текста, а если выключен, то после. Элемент управления **Appearance** определяет изображение, которое будет использоваться в качестве элемента для выбора ответа.

С помощью элемента управления **Choice Is** каждый из альтернативных вариантов ответов должен быть помечен как правильный (Correct) или как неправильный (Incorrect). Если альтернативный вариант ответа помечен значением Not Judged, то неважно выберет ученик этот ответ или нет, так как при проверке он не будет приниматься во внимание.

Каждому альтернативному варианту ответа можно назначить свой балл (оценку), который затем использовать для интегральной оценки по результатам выполнения нескольких тестовых заданий. Для этого в поле **Score** необходимо указать численное значение оценки. Например, если в тестовом задании с четырьмя альтернативными ответами два правильных, то можно назначить каждому из них балл 2.5, чтобы общая оценка за выполнение тестового задания оказалась равной 5.

После того, как параметры всех альтернативных вариантов ответов указаны, можно перейти к последней закладке, которая называется **Action Mgr** (рис. 7.7). Работа с закладкой **Action Mgr** при конструировании тестовых заданий в любой другой форме ничем не отличается.

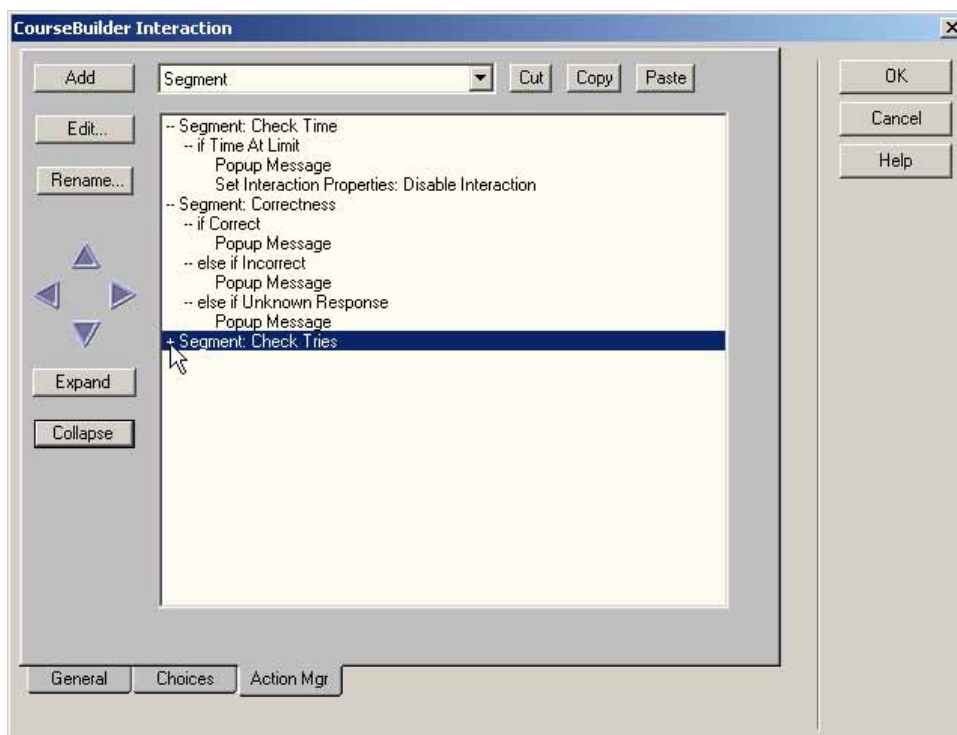


Рис. 7.7. Диалоговое окно CourseBuilder (закладка Action Mgr)

На этой закладке на псевдоязыке программирования отображается программа предъявления реакции на выполнение тестового задания (Segment: Correctness), на окончание количества попыток (Segment: Check Tries) и отведенного времени (Segment: Check Time). Эти действия собраны в сегменты, которые могут быть развернуты или свернуты с помощью кнопок **Expand** и **Collapse** соответственно. С помощью символических стрелок действия можно перемещать по тексту псевдопрограммы. Команды псевдопрограммы можно вырезать, копировать и вставлять с помощью кнопок **Cut**, **Copy** и **Paste** соответственно. С помощью кнопки **Add** в псевдопрограмму можно добавлять новые сегменты и действия, которые перечислены в списке справа от этой кнопки. А с помощью кнопок **Edit...** и **Rename...** можно отредактировать или переименовать существующие сегменты и действия. Таким образом, Action Manager является мощным средством управления событиями интерактивных элементов CourseBuilder.

При создании тестовых заданий CourseBuilder назначает приемлемые алгоритмы обработки событий. В качестве реакции на ответ, на окончание количества попыток или отведенного времени используется небольшое диалоговое окно с сообщением (Popup Message). Единственным недостатком является то, что сообщения достаточно скудные и заданы на английском языке. Из списка действий необходимо выбрать пункт **Popup Message** и в зависимости от контекста изменить текст сообщения. Например, в случае правильного выполнения тестового задания из сегмента **Correctness** будет выполнено условие **if Correct**, что приведет к вызову диалогового окна **Popup Message** с сообщением на английском языке: "Correct". Для изменения текста сообщения необходимо выбрать этот пункт списка **Popup Message** и нажать кнопку **Edit...** На появившемся диалоговом окне в текстовом поле **Message** необходимо заменить английское слово "Correct" на русское "Правильно", "Ответ правильный" или любое другое. Аналогично следует поступить и с реакцией на неправильный ответ, которая вызывается, если выполняется условие **else if Incorrect**. Если ученик еще не дал ответ, а уже нажал кнопку "Проверить", то выполнится условие **else if Unknown Response**. Сообщение "Unknown Response" в действии **Popup Message** необходимо заменить на сообщение "Задание не выполнено" или на повелительное сообщение "Выполните тестовое задание!"

После того, как все необходимые действия по настройке "скелета" тестового задания выполнены, на диалоговом окне CourseBuilder можно нажать кнопку **ОК**. Внешний вид тестового задания в окне редактора представлен на рисунке 7.8, а в окне браузера – на рисунке 7.9.

При необходимости можно вернуться в режим редактирования тестового задания. Для этого необходимо выделить иконку , которая расположена после тестового задания (рис. 7.8). Эту иконку следует также использовать для копирования и переноса соответствующей ей интерактивного элемента CourseBuilder. При ее выделении на панели свойств Dreamweaver появляются свойства элемента CourseBuilder, а в окне редактора должно выделиться все

тестовое задание полностью. Необходимое для изменения параметров диалоговое окно вызывается по кнопке **Edit...** на панели свойств (рис. 7.10).

Из перечисленного списка городов выберите столицы европейских государств. |

☐ Вашингтон 
☐ Рим 
☐ Цюрих 
☐ Варшава 



Рис. 7.8. Тестовое задание в окне редактора

Из перечисленного списка городов выберите столицы европейских государств.

☐ Вашингтон
☐ Рим
☐ Цюрих
☐ Варшава

Рис. 7.9. Тестовое задание в окне браузера

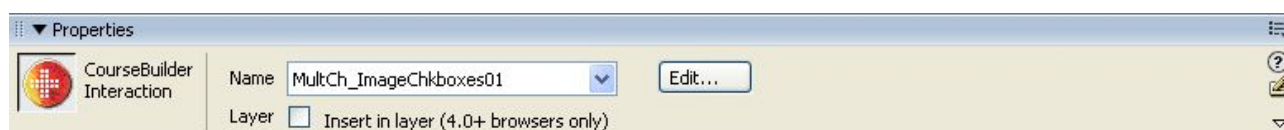


Рис. 7.10. Внешний вид окна Properties для элемента CourseBuilder

7.3 Создание тестового задания в открытой форме

Для создания тестового задания в открытой форме в диалоговом окне CourseBuilder необходимо выбрать категорию **Text Entry** (рис. 7.11). Типы тестовых заданий открытого типа, которые предлагает создать CourseBuilder, отличаются лишь размерами поля для ввода ответов (однострочное или многострочное). Элементы управления для настройки тестового задания расположены на появившихся закладках **General**, **Responses**, **Action Mgr**.

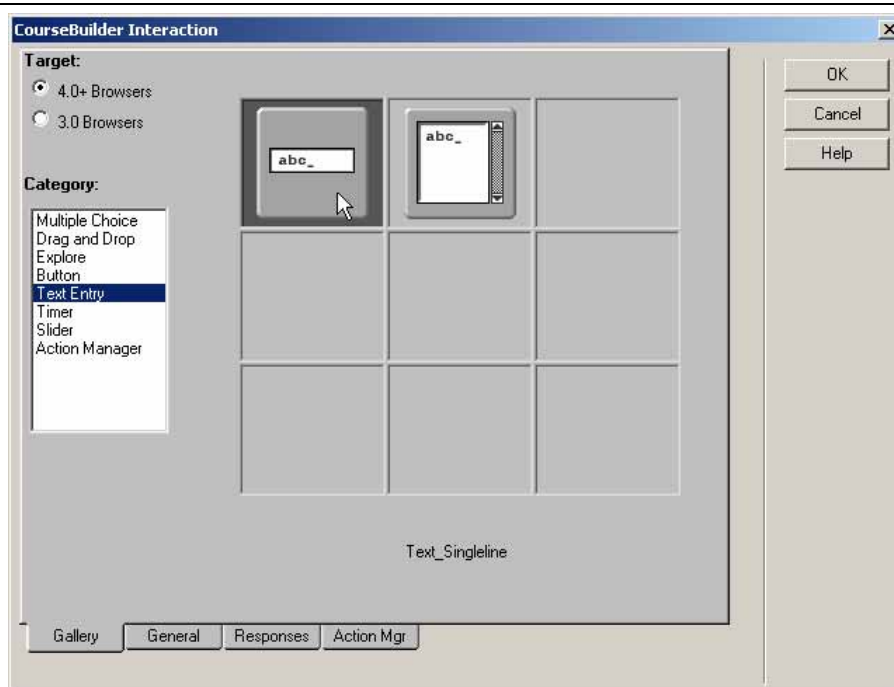


Рис. 7.11. Диалоговое окно CourseBuilder (закладка Gallery)

Закладка **General** (рис. 7.12) практически не изменилась по сравнению с тестовыми заданиями закрытого типа. Только вместо поля **Question Text** размещен элемент управления **Initial Text**. В этом поле можно записать начальный текст, который будет выведен в поле для ответа при предъявлении тестового задания. Начальный текст может сообщать о формате, в котором должен быть введен ответ. Например, если ответом на тестовое задание должна быть какая-либо дата, то в поле **Initial Text** можно записать: “Ответ введите в форме ДД.ММ.ГГГГ”. Формулировку же тестового задания придется записать в обычном режиме редактирования.

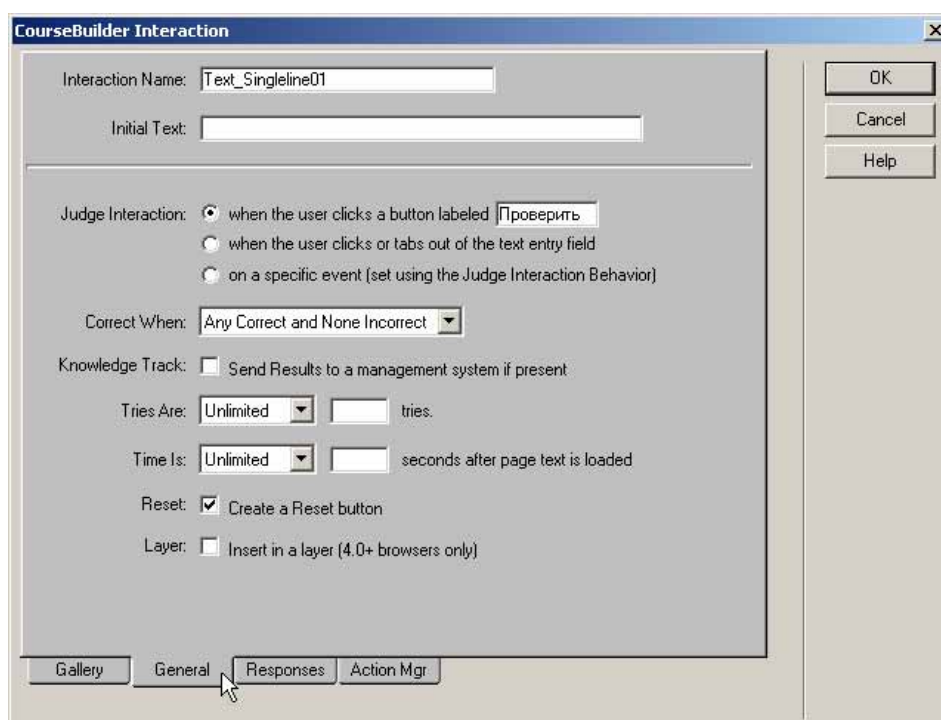


Рис. 7.12. Диалоговое окно CourseBuilder (закладка General)

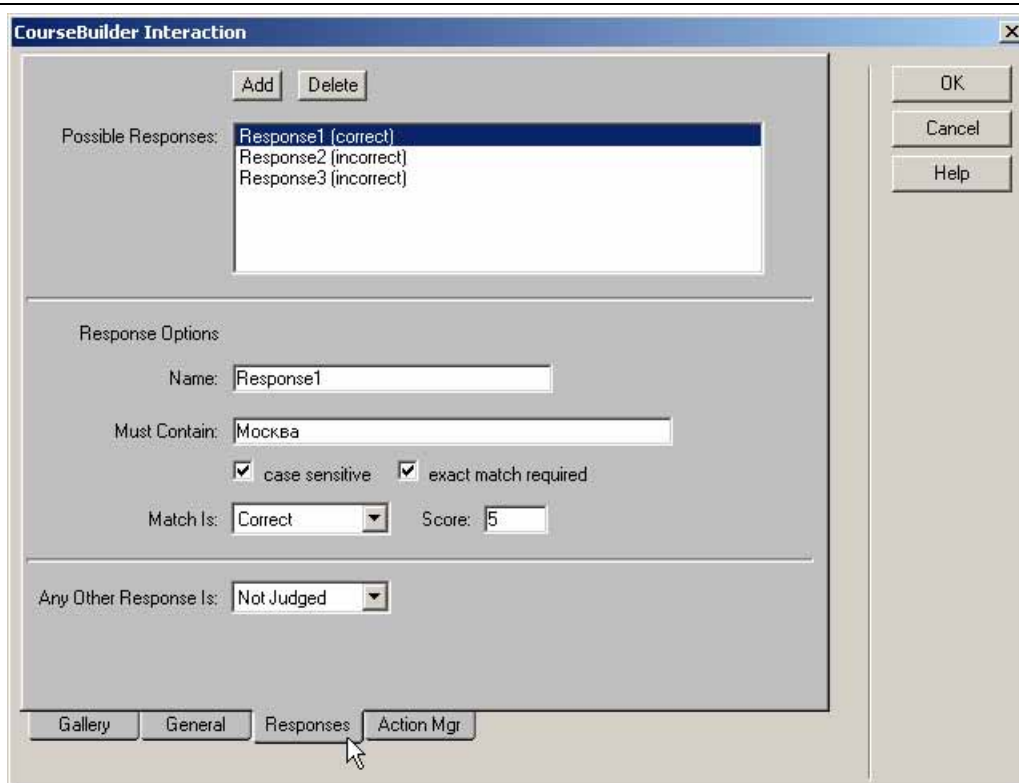


Рис. 7.13. Диалоговое окно CourseBuilder (закладка Responses)

На закладке **Responses** (рис. 7.13) задаются возможные варианты ответов. Ответы на тестовое задание открытого типа должны представлять собой ключевое слово или словосочетание. Каждый из них должен иметь уникальное имя (**Name**), которое используется JavaScript-программами, и поэтому должно состоять из латинских символов. В списке **Possible Responses** отображаются имена ответов, а в нижней части диалогового окна – их параметры. Изменить количество ключевых слов, которых автор ожидает в качестве ответа, можно с помощью кнопок **Add** и **Delete**.

Последовательно выбирая имена ответов из списка **Possible Responses**, необходимо ввести формулировку ожидаемого варианта ответа (в поле **Must Contain**). Флажок **case sensitive** включает при проверке тестового задания режим учета регистра символов, а флажок **exact match required** включает режим проверки, при котором ответ, данный учеником, должен полностью соответствовать ответу, предусмотренному преподавателем, а наличие других ключевых слов в ответе запрещается.

С помощью элемента управления **Match Is** каждое из ключевых слов, предусмотренных автором в ответе, должно быть помечено как правильное (Correct), неправильное (Incorrect) или не влияющее на результат проверки (Not Judged). Также автор тестового задания должен определить, как CourseBuilder должен относиться к любым другим словам и словосочетаниям, не предусмотренным автором, но использованным учеником для ответа. Это можно сделать с помощью списка **Any Other Response Is**, в котором выбрать одну из трех установок: Correct (любые непредусмотренные ключевые слова считать допустимыми), Incorrect (ключевые слова непредусмотренные автором

будут считаться недопустимыми) или Not Judged (любые другие ключевые слова никак не повлияют на результат выполнения тестового задания).

Каждому ключевому слову в поле **Score** можно назначить свой балл (оценку), который затем использовать для интегральной оценки по результатам выполнения нескольких тестовых заданий.

7.4 Создание тестового задания на установление соответствия

Тестовые задания на установление соответствия можно реализовать, используя элементы на основе перетаскивания (Drag-and-drop Interactions). Для создания такого тестового задания в списке **Category** диалогового окна CourseBuilder (рис. 7.14) необходимо выбрать группу **Drag-and-drop**. CourseBuilder позволяет создать различные тестовые задания, которые отличаются количеством элементов и направлением перетаскивания.

Рассмотрим реализацию классического тестового задания на установление соответствия с равным количеством элементов множеств и перетаскиванием в одном направлении. Элементы управления для настройки тестового задания расположены на появившихся закладках **General**, **Elements**, **Pairs**, **Action Mgr**.

Закладка **General** (рис. 7.15) практически не отличается при настройке различных тестовых заданий. Для тестовых заданий на основе перетаскивания на закладке **General** появились дополнительные элементы управления, озаглавленные **Reaction**. Если включить флажок **Snap back if incorrect**, то в случае неправильного указания соответствия между элементами множеств перетаскиваемый элемент будет возвращаться на свое исходное место.

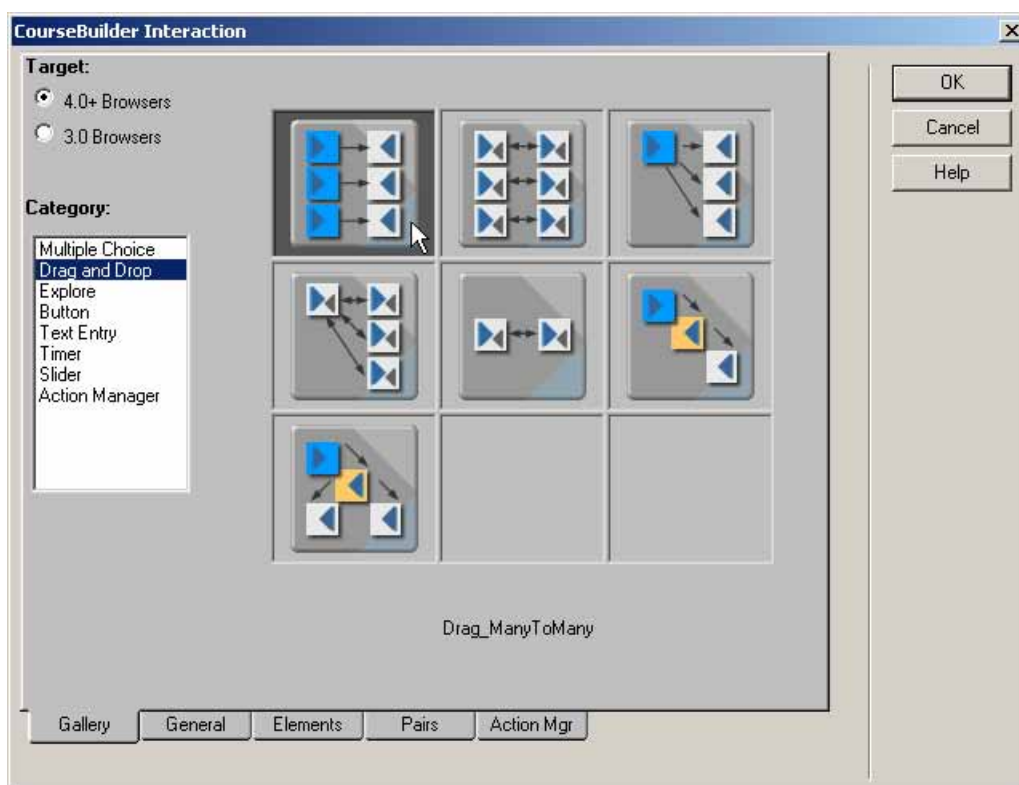


Рис. 7.14. Диалоговое окно CourseBuilder (закладка Gallery)

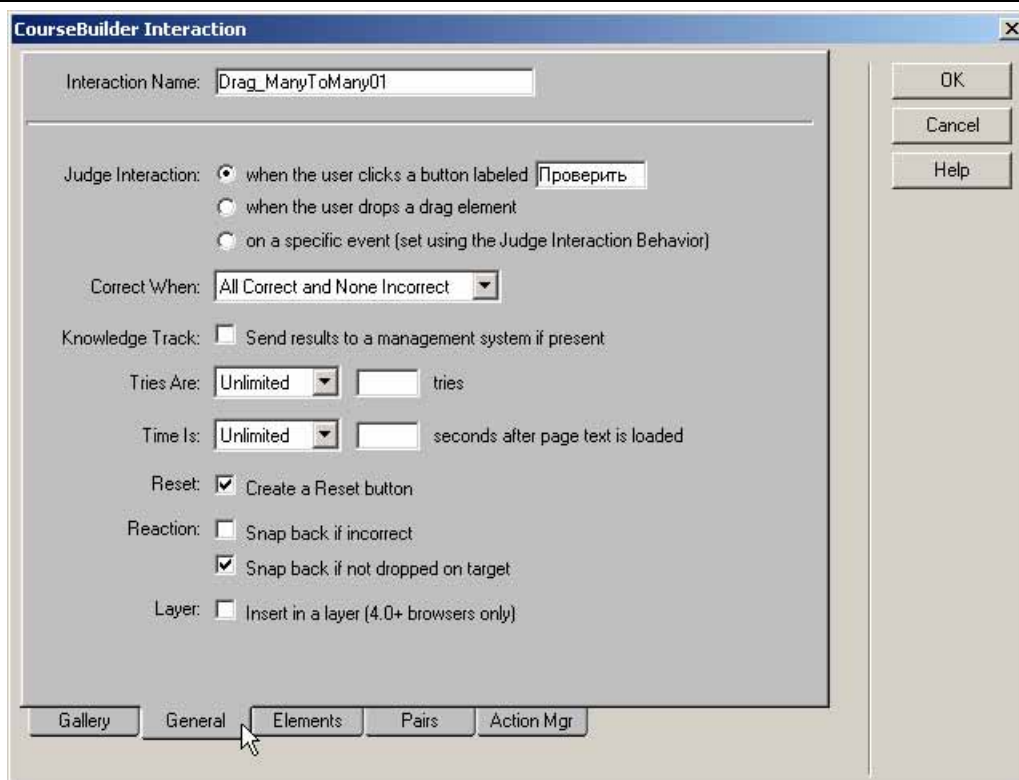


Рис. 7.15. Диалоговое окно CourseBuilder (закладка General)

Выполнить такое тестовое задание неправильно будет просто невозможно, поэтому этот режим используется только при организации обучающего тестирования и не используется при аттестации. Флажок **Snap back if not dropped on target** включает режим, в котором перетаскиваемый элемент необходимо точно совместить с перетаскиваемым. Оставить элемент в произвольном месте окна браузера система не позволит.

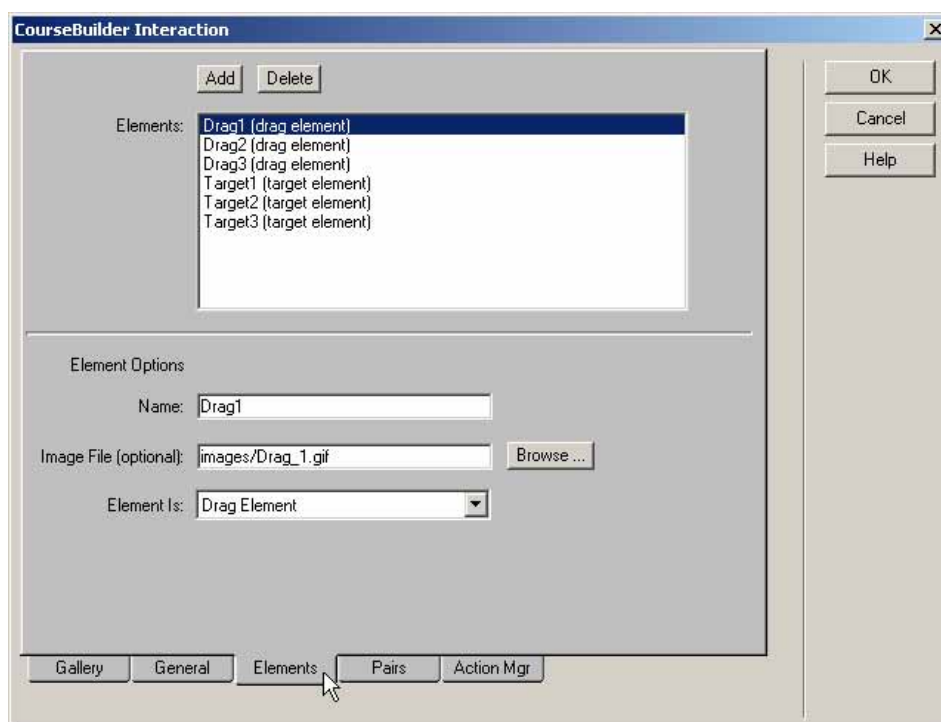


Рис. 7.16. Диалоговое окно CourseBuilder (закладка Elements)

Формулировку тестового задания, созданного на основе перетаскиваемых элементов, записывают в обычном режиме редактирования.

Закладка **Elements** (рис. 7.16) предназначена для определения параметров элементов множеств. Созданные элементы отображаются в списке **Elements**. Изменить их количество можно с помощью расположенных выше кнопок **Add** и **Delete**. Каждый из элементов должен иметь уникальное имя (**Name**), которое используется JavaScript-программами, и поэтому должно состоять из латинских символов. Последовательно выбирая имена элементов из списка **Elements**, можно определить их параметры. Во-первых, тип элемента (**Element Is**): Drag – перетаскиваемый элемент, Target – целевой элемент или Drag and Target, когда автор разрешает элементу играть обе роли (быть перетаскиваемым и целевым одновременно). Во-вторых, автор может задать имя файла (**Image File**) с изображением, которое будет использоваться в качестве перетаскиваемого или целевого элемента при выполнении данного тестового задания. Путь к файлу с изображением можно указать в специальном диалоговом окне, которое появляется по кнопке **Browse...** При создании “скелета” тестового задания данного типа в качестве элементов CourseBuilder использует заготовки, которые следует просто заменить. Если в качестве элемента вы не предполагаете использовать изображение, то поле **Image File** следует очистить и оставить пустым.

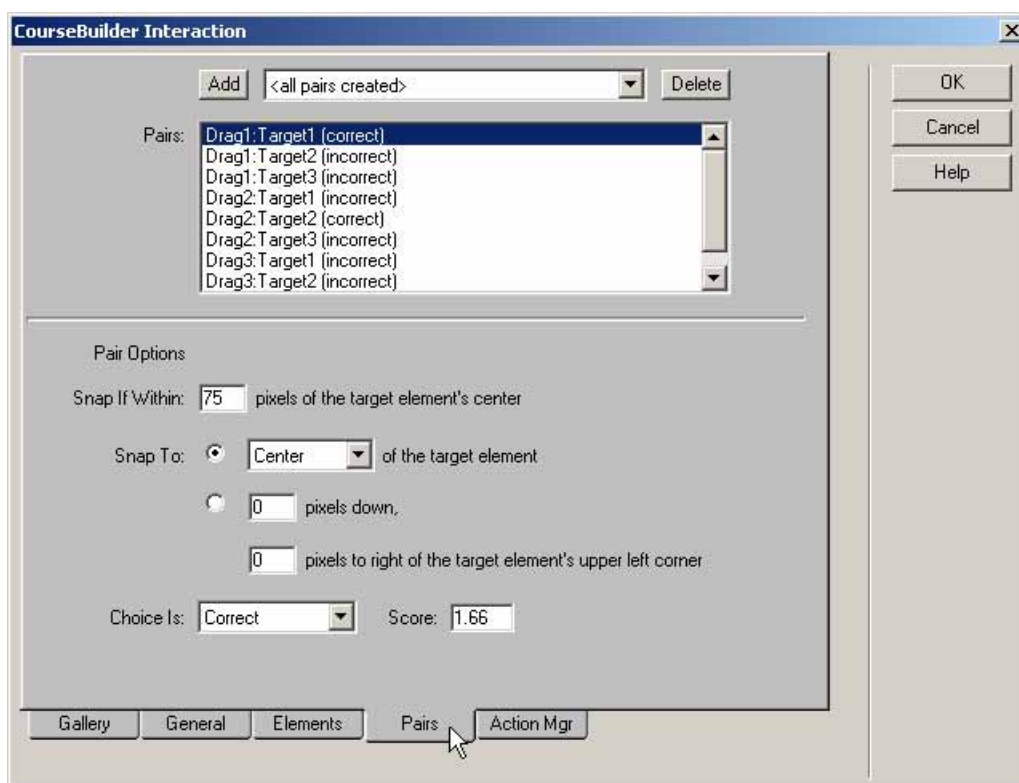


Рис. 7.17. Диалоговое окно CourseBuilder (закладка Pairs)

На закладке **Pairs** (рис. 7.17) задаются параметры всех возможных соответствий между всеми возможными элементами. Добавить и удалить пары можно с использованием кнопок **Add** и **Delete**, а выбрать пары элементов, соответствия между которыми еще не были рассмотрено, можно из расположенного между кнопками списка.

Последовательно выбирая пары элементов из списка **Pairs**, следует определить параметры каждой из них. В первую очередь, – является ли данное соответствие верным или нет. Как и при конструировании других тестовых заданий в списке **Choice Is** необходимо выбрать значение **Correct**, если соответствие является правильным, значение **Incorrect**, если соответствие некорректно, и **Not Judged**, если соответствие является нейтральным.

Остальные элементы управления на закладке **Pairs** определяют параметры отображения элементов, приведенных в соответствие. В процессе выполнения задания перетаскиваемый элемент как магнитом будет притягиваться к целевому элементу. Параметр **Snap If Within** определяет размеры области вокруг целевого элемента, в которой будет выполняться притягивание. CourseBilder по умолчанию задает размер этой области, равный 75 пикселям. Если в качестве элементов используются небольшие изображения или обычный текст, то размеры области притягивания следует уменьшить. Также это следует сделать, если целевые элементы расположены достаточно плотно друг к другу, иначе области притяжения элементов будут перекрываться.

Группа параметров **Snap To** определяет положение перетаскиваемого элемента относительно целевого после указания соответствия, то есть после выполнения задания. По умолчанию положение перетаскиваемого элемента будет выравниваться относительно центра целевого элемента. Если в качестве элементов используется текст, то в этом случае после притягивания их содержимое будет сливаться. Чтобы этого не происходило, лучше выбрать одно из предусмотренных значений (**Left**, **Right**, **Top**, **Bottom**) или в полях для ввода, расположенных ниже, явно указать координаты точки, относительно которой осуществлять выравнивание.

При создании тестовых заданий на основе перетаскивания в качестве элементов может использоваться текст. Для этого в режиме обычного редактирования надо выделить и удалить содержимое элемента, а затем вписать необходимый текст (рис. 7.18).



Рис. 7.18. Тестовое задание на установление соответствия в окне редактора

Однако, наиболее эффектно тестовые задания с использованием механизма перетаскивания выглядят, когда в качестве перетаскиваемых элементов используются графические изображения (рис. 7.19).

Приведите в соответствие авторов пьес и их названия.

Проверить

Повторить

	 Бедность не порок
	 Горе от ума
	 Ревизор
	 Недоросль

Рис. 7.19. Тестовое задание на установление соответствия в окне браузера

Перетаскиваемые и целевые элементы реализуются с использованием слоев, поэтому их положение автор может поменять, перетаскивая за левый верхний угол с помощью мыши в обычном режиме редактирования.

7.5 Создание тестового задания на установление правильной последовательности

Тестовые задания на установление правильной последовательности можно реализовать аналогично на основе механизма перетаскивания (Drag-and-drop Interactions).

Восстановить историческую очередность основных этапов развития первобытной культуры.



Рис. 7.20. Тестовое задание на установление правильной последовательности в окне браузера

При этом, в качестве перетаскиваемых элементов можно использовать графические изображения номеров последовательности (рис. 7.20) и т.п.

7.6 Создание последовательности тестовых заданий

Для создания системы тестирования тестовые задания должны быть объединены в цепочку. При помощи CourseBuilder такую последовательность можно построить несколькими способами.

Простейший способ создания набора тестов – размещение их на одной web-странице непосредственно друг за другом. В данном случае существенными недостатками являются невозможность контроля за порядком ответа на отдельные задания и невозможность запрета пропуска заданий. К положительным свойствам можно отнести возможность подсчета итогового результата.

Второй способ подразумевает создание каждого тестового задания в отдельном файле. При организации прохождения такой цепочки тестовых заданий необходимо после проверки задания и предъявления реакции вставить действие **Go To URL** (переход к файлу или на указанный Internet-ресурс). Для этого в списке в верхней части закладки ActionMgr окна CourseBuilder Interaction необходимо выбрать действие **Go To URL**. Затем выбрать в списке то действие, после которого следует осуществить переход к следующей странице, и нажать кнопку **Add**. В появившемся диалоговом окне в поле **URL** (рис. 7.21) необходимо указать имя файла, к которому должен осуществляться переход, и нажать кнопку **OK**.

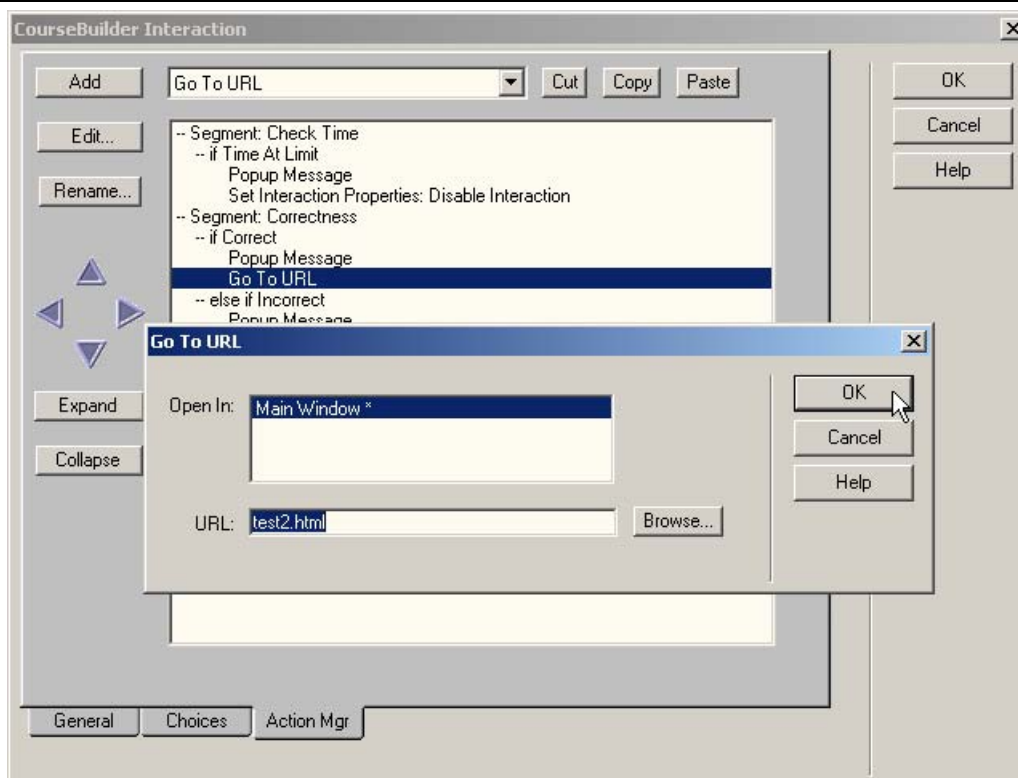


Рис. 7.21. Вставка действия Go To URL

Такой способ достаточно прост и позволяет построить эффективную цепочку. Однако, в данном случае исключается возможность подсчета итогового результата (так как при переходе из одного HTML-файла в другой данные JavaScript программ теряются).

Следующий вариант лишен недостатков первых двух: он позволяет построить достаточно эффективную цепочку заданий и, вместе с тем, организовать подсчет результатов. Однако с точки зрения реализации он несколько сложнее, так как подразумевает использование слоев.

Для организации подобной цепочки требуется создать все задания в одном файле, но каждое – на своем слое. Для этого, при создании отдельных заданий, необходимо устанавливать опцию **Layer** на закладке **General**. Слои имеют свойство, позволяющее делать их невидимыми. Чтобы сделать слой невидимым необходимо на панели свойств слоя (рис. 7.22) для параметра **Vis** выбрать значение **hidden**.

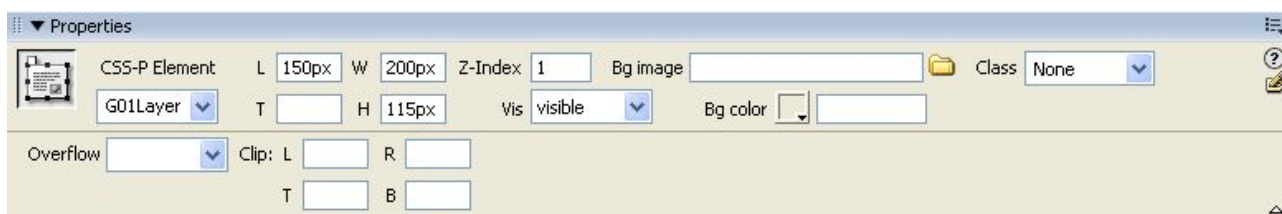


Рис. 7.22 Панель свойств слоя

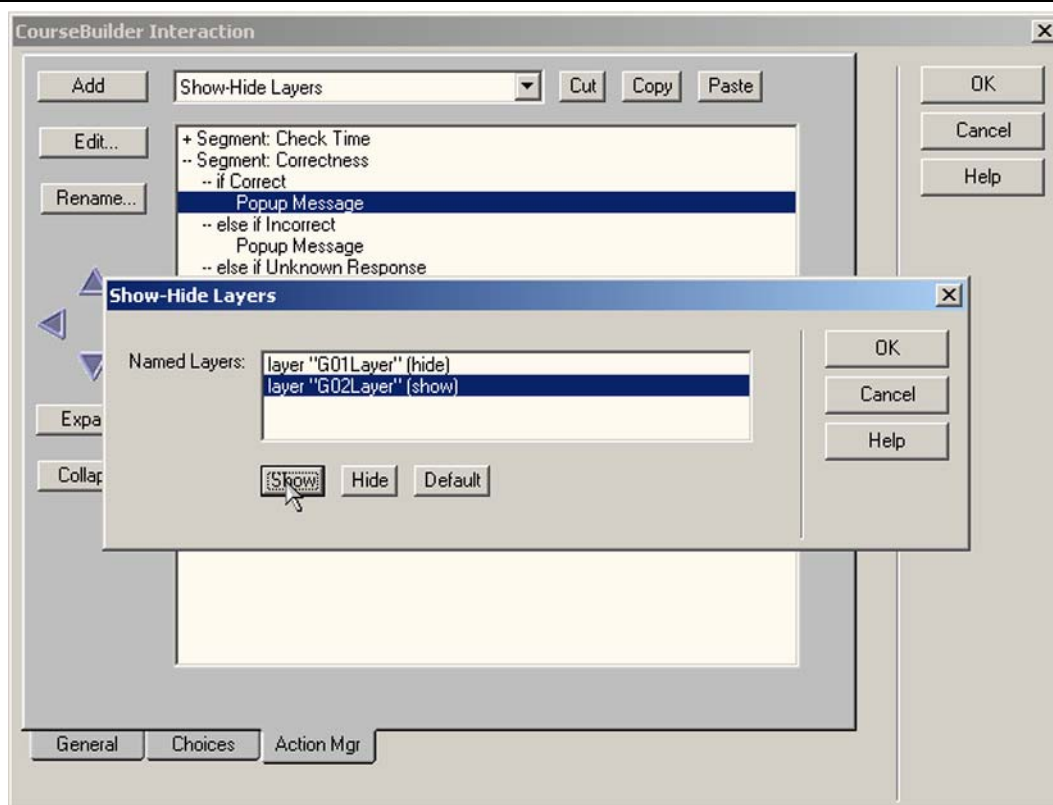


Рис. 7.23 Вставка действия Show-Hide Layers

Видимым нужно оставить только первое задание в цепочке, а все остальные спрятать. Затем, аналогично тому, как делались переходы из одного файла в другой во втором методе, нужно организовать последовательность смены слоев. Для этого необходимо после проверки задания и предъявления реакции вставить действие **Show-Hide Layers** (показать/спрятать слои). В появившемся диалоговом окне (рис. 7.23) в списке **Named Layers** перечислены все слои, присутствующие в документе. В этом списке нужно выделить тот слой, который нужно спрятать (т.е. слой с текущим тестовым заданием), и нажать кнопку **Hide** (спрятать). Слой со следующим вопросом нужно наоборот сделать видимым – его надо выделить и нажать кнопку **Show** (показать).

7.7 Организация подсчета и вывода результатов

Для того, чтобы организовать вывод интегрированной оценки после выполнения учеником нескольких тестовых заданий, необходимо, чтобы все задания находились в одном файле. В каждом тестовом задании, для каждого из вариантов ответов должен быть указан балл в поле **Score** (для элементов **Multiple Choice** – закладка **Choices**; для элементов **Text Entry** – закладка **Responses**; для элементов **Drag-and-Drop** – закладка **Pairs**). В конце HTML-документа после всех тестовых заданий необходимо вставить новый элемент CourseBuilder. Из списка Category следует выбрать тип элемента – **Action Manager** (рис. 7.24).

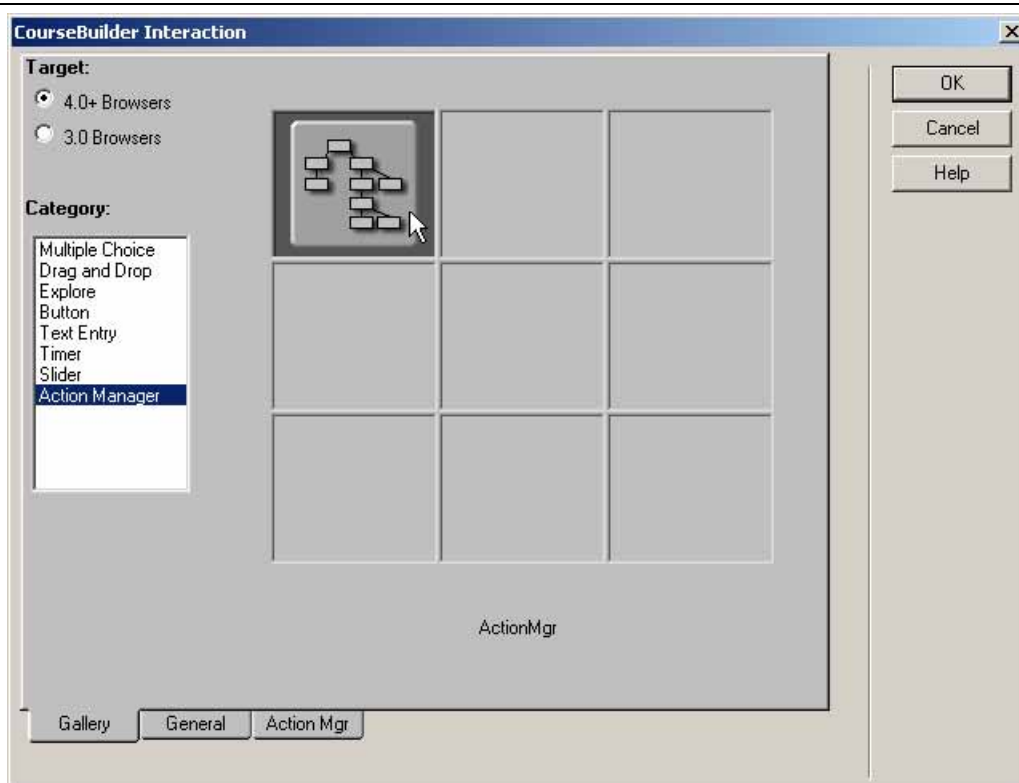


Рис. 7.24. Диалоговое окно CourseBuilder (закладка Gallery)

Далее на закладке **General** (рис. 7.25) необходимо организовать выполнение действий по нажатию на кнопку. Для этого необходимо выбрать переключатель **when the user clicks a button labeled**, а в тестовом поле справа задать надпись на кнопке (например, “Результат”). Остальные элементы управления на этой закладке следует оставить без изменений.

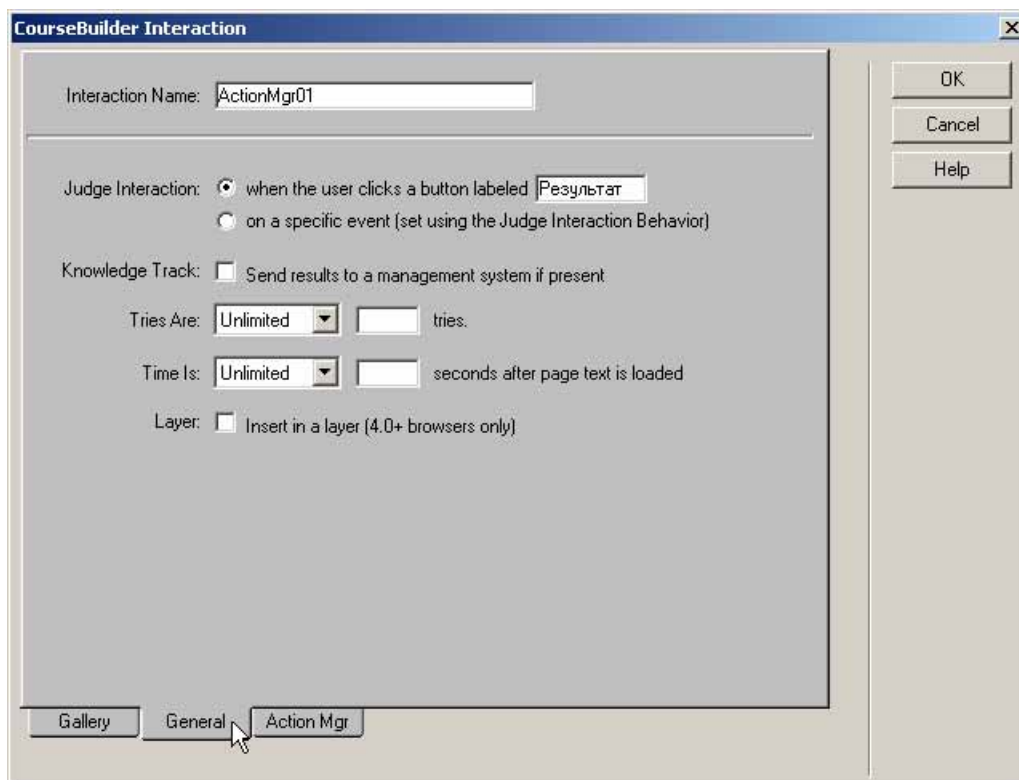


Рис. 7.25. Диалоговое окно CourseBuilder (закладка General)

Действия для вычисления интегрированной оценки необходимо задать с использованием элементов управления, расположенных на закладке **Action Mgr**. Во-первых, необходимо создать новый сегмент. Для этого в списке в верхней части диалогового окна выбрать пункт **Segment** и нажать кнопку **Add**.

На появившемся диалоговом окне **Segment Editor** (рис. 7.26) в поле **Segment Name** необходимо задать имя сегмента (например, "Segment : Result") и нажать **OK**.

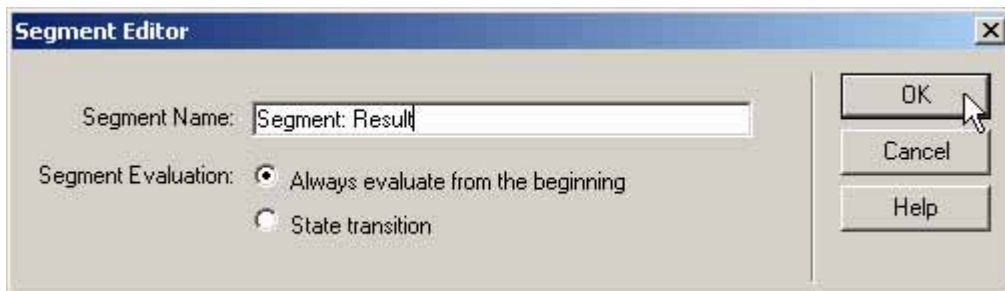


Рис. 7.26. Диалоговое окно для создания нового сегмента

А во-вторых, осуществить в этом сегменте вызов диалогового окна с сообщением. Для этого необходимо выбрать из списка в верхней части диалогового окна значение **Popup Message** и нажать кнопку **Add**. На появившемся диалоговом окне (рис. 7.27) в поле **Message** необходимо написать текст сообщения и формулы для вычисления результатов тестирования. Например, для теста, состоящего из пяти заданий с суммарной оценкой для каждого задания в 5 баллов, в поле **Message** можно написать следующее:

Вы набрали

`{G01.score+G02.score+G03.score+G04.score+G05.score}`

баллов, что составляет

`{(G01.score+G02.score+G03.score+G04.score+G05.score)*100/25} %`.

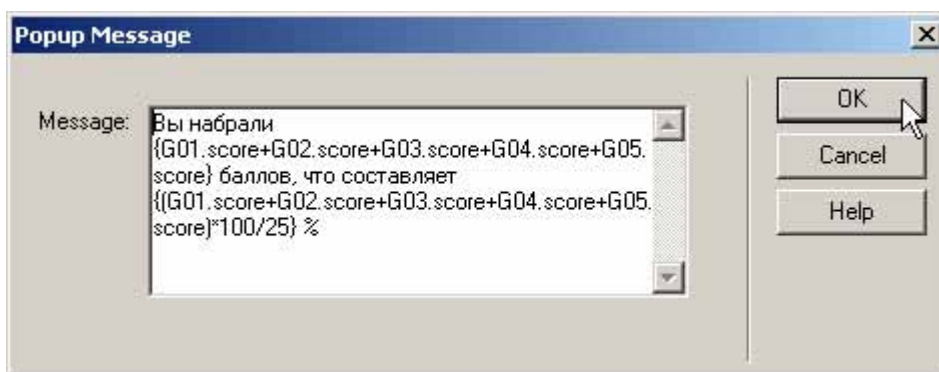


Рис. 7.27. Организация вывода результата в диалоговое окно с сообщением

Математические выражения заключаются в фигурные скобки { }. G01, G02, G03, G04 и G05 – имена интерактивных объектов CourseBuilder, которые можно узнать, если заглянуть в HTML-код, с помощью которого он реализован. Так как интерактивные элементы CourseBuilder являются объектами, доступ к их параметрам можно получить, используя оператор . (точка). Таким образом,

G01.score, G02.score, G03.score, G04.score и G05.score – числовое значение, равное оценке за выполнение тестового задания.

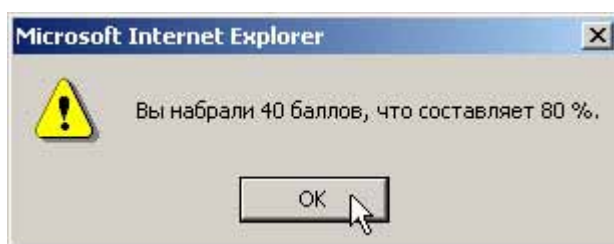


Рис. 7.28. Окно с сообщением результата выполнения тестовых заданий

После выполнения всех тестовых заданий по кнопке “Результат” ученику будет предъявляться окно с сообщением, примерный вид которого представлен на рисунке 7.28.

Приложение А. Краткая справка по тегам языка HTML

Структура документа HTML

Документ в формате HTML состоит из трех основных частей:

- строка, содержащая информацию о версии HTML;
- заголовочная часть (определяется тегом <HEAD>);
- тело, которое включает собственно содержимое документа (определяется тегом <BODY> или тегом <FRAMESET>).

Кроме того, вторая и третья части должны охватываться тегом <HTML>. На практике первая часть часто опускается.

Заголовочная часть содержит информацию о документе в целом, которая не отображается в окне браузера. В теле находится непосредственно информационное содержание страницы, вместе с описанием своего форматирования – то что выводится в окне браузера. Вместо тела документ может содержать описание фреймов (тег <FRAMESET>). Следующий пример демонстрирует теги, образующие скелет простейшего документа HTML:

```
<html>
<head>
  <title>Простой HTML-документ</title>
</head>
<body text="#0000ff" bgcolor="#ffffff">
  <h1>Заголовок документа</h1>
  <p> Пример простого документа</p>
</body>
</html>
```

Ниже приводится список основных тегов HTML в алфавитном порядке, содержащий их краткое описание, примеры использования и результаты интерпретации браузером.

Теги языка HTML

<a>

Якорь – создает ссылку на Internet-ресурс, указанный обязательным атрибутом **href**. Атрибут **name** задает имя якоря, так что он может служить пунктом назначения других ссылок. Ссылки и якоря не могут быть вложенными. Атрибут **title** может использоваться для сообщения информации о связанном ресурсе.

Текст, содержащий ссылку
--

Текст, содержащий <u>ссылку</u>



Заголовок ссылки

<abbr>, <acronym>

Эти теги позволяют явно показывать использование аббревиатур и акронимов. Западные языки широко используют акронимы, такие как "GmbH", "НАТО" и "ФБР", а также аббревиатуры типа "М.", "Inc.", "etc.", "WWW", "HTTP", "URI". Разметка этих конструкций предоставляет броузерам и вспомогательным средствам, таким как средства проверки орфографии, синтезаторы речи, переводящие системы и индексаторы поисковых машин, полезную информацию. Атрибут **title** этих элементов может использоваться для указания полной развернутой формы выражения.

<address>

Может использоваться для указания контактной информации.

Контактная информация: <address>Россия, 197101, Санкт-Петербург, ул.Саблинская, д.14</address>

Контактная информация: <i>Россия, 197101, Санкт-Петербург, ул.Саблинская, д.14</i>

<area>

Определяет чувствительную к перемещению мыши область изображения. Используется совместно с тегом <MAP>. Атрибут **href** указывает URL связанного с данной зоной документа. Параметр **shape** задает форму активной зоны (**rect** – определяет прямоугольную область, **circle** – круглую, **poly** – многоугольную), а **coord** – указывает (через запятую) список зависящих от формы координат, которые определяют ее границы.

<pre> <map name="Navigation"> <area shape="rect" coords="44,16,64,48" href="page4.html" alt="Следующая страница" title="Следующая страница"> </map></pre>
--



Заклученный в теги текст отображается жирным шрифтом.

Пример жирного текста

Пример жирного текста

<base>

В HTML ссылки на внешние изображения, апплеты, программы для обработки форм, таблицы стилей и т.д. задаются с помощью адресов URL. Относительные адреса разрешаются в соответствии с базовым URL. Тег <BASE> при помощи атрибута **href** позволяет явно указать базовый URL документа.

<big>

Заклученный в теги текст отображается шрифтом большего размера.

Пример <big>увеличенного</big> текста

Пример увеличенного текста

<blockquote>

Тег предназначен для длинных цитат. Заклученный в теги текст размещается с новой строки. При этом поля справа и слева от текста увеличиваются. При использовании этого тега несколько раз текст все больше сжимается к центру.

Пример использования длинных цитат: <blockquote>цитата <blockquote>вложенная цитата </blockquote> </blockquote>

Пример использования длинных цитат:

цитата

вложенная цитата

<body>

Определяет тело документа. Атрибут **text** позволяет задать цвет основного текста в документе. Атрибуты **bgcolor** и **background** задают цвет фона и фоновое изображение соответственно. Атрибуты **link**, **alink** и **vlink** устанавливают цвет обычных, активных и посещенных гипертекстовых ссылок соответственно.

Осуществляет принудительный перевод строки. Текст, расположенный после этого тега, отображается с новой строки.

Пример принудительного перехода на следующую строку
--

Пример принудительного перехода на следующую строку
--

<button>

Элемент позволяет создавать кнопки в формах, действующие аналогично кнопкам, создаваемым с помощью элемента <INPUT>, но обеспечивающим более богатые возможности представления. Кнопки могут иметь содержимое в виде изображения или текста. Атрибут **name** присваивает имя элементу управления. Параметр **value** присваивает кнопке начальное значение. Атрибут **type** позволяет задать тип кнопки (**submit** – значение по умолчанию, отправка данных, введенных пользователем web-серверу; **reset** – очистка формы и восстановление первоначальных значений параметров; **button** – настраиваемая кнопка, используемая совместно с JavaScript).

```
<form name="test_form" method="post" action=
"cgi-bin/action.cgi">
  <button type="button" name="Message" onClick=
"alert('Сообщение')"></button>
  <button type="submit" name="Submit"> Отправить
</button>
  <button type="reset" name="Reset"> Сбросить
</button>
</form>
```



Отправить

Сбросить

<caption>

Задаёт заголовок таблицы.

<cite>

Заклученный в теги текст представляет собой цитату и отображается наклонным шрифтом.

Пример <cite>цитаты</cite>

Пример *цитаты*

<code>

Заклученный в теги текст представляет собой отрывок текста программы и отображается шрифтом фиксированной ширины.

Пример фрагмента кода:<code>
текст</code>

Пример фрагмента кода: текст

`<col>`

Определяет свойства столбца (или столбцов) в группе столбцов (тег `<COLGROUP>`). Столбцы могут иметь общие атрибуты (например, выравнивание), не будучи частью формальной структурной группы. Атрибут **span** определяет число столбцов, охваченных элементом `<COL>`. Параметр **align** задает способ выравнивания текста, а **valign** – способ вертикального выравнивания в ячейках таблицы. Задать ширину (в пикселах, процентах или относительную) каждого столбца, охваченного элементом `<COL>` можно при помощи атрибута **width**.

`<colgroup>`

Создает группу столбцов. Этому структурному образованию внутри таблицы могут быть назначены атрибуты при помощи стилей CSS или HTML.

`<dd>`

Задаёт описание термина в списке определений (`<DL>`).

`<dfn>`

Тег указывает, что текст является определением вложенного термина.

Пример <code><dfn></code> определения <code></dfn></code> вложенного термина
--

Пример <i>определения</i> вложенного термина
--

`<div>`

Определяет информацию уровня блока, не налагая при этом никаких других ограничений для представления контекста. Авторы могут использовать этот элемент с таблицами стилей, атрибутами **lang** и т.д. Атрибут **align** позволяет задавать выравнивание.

<code><p></code> Текст для примера <code><div class="caption"></code> Текст для примера <code></div></code> Текст для примера

Текст для примера Текст для примера Текст для примера

`<dl>`

Задаёт список определений, состоящий из терминов (тег `<DT>`) и определений (`<DD>`).

<code><p></code> Пример списка определений: <code></p></code> <code><dl></code> <code><dt></code> термин <code></dt></code> <code><dd></code> определение <code></dd></code> <code></dl></code>

Пример списка определений:

термин	определение

<dt>

Обозначает термин, включаемый в список определений (<DL>).

В теги заключается текст требующий смыслового усиления и отображается наклонным шрифтом.

Текст требующий смыслового усиления .
--

Текст требующий <i>смыслового усиления.</i>

Заключенный в теги текст будет отображаться размером шрифта, указанным атрибутом **size** и цветом, указанным атрибутом **color**. Значением атрибута **size** может быть задано абсолютным значением (от 1 до 7, по умолчанию 3) или как относительное при помощи знаков плюс или минус. Атрибут **face** позволяет установить гарнитуру шрифта (устанавливается первый существующий из указанных в разделенном запятыми списке имен шрифтов).

Текст для примера .

Текст для примера.

<form>

Обозначает границы формы. В одном HTML-документе может быть несколько форм, но они не могут быть вложенными и не должны перекрываться. Атрибут **action** задает URL приложения, которое будет обрабатывать форму, а обязательный параметр **method** определяет какой метод HTTP (**get** или **post**) будет использован для обработки данных формы.

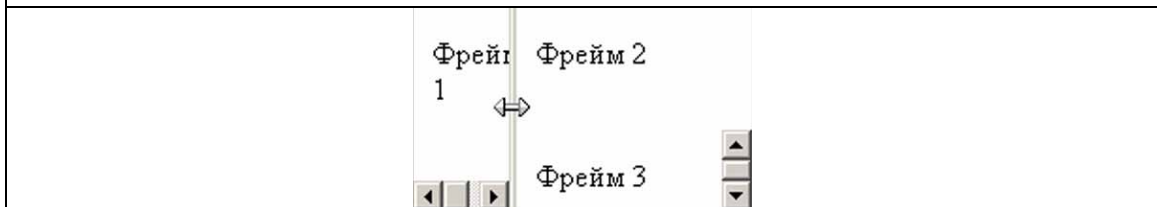
<frame>

Определяет один фрейм в составе тега <FRAMESET>. Атрибут **src** задает путь к HTML-файлу, который должен быть загружен во фрейм. Назначить имя фрейму можно при помощи атрибута **name**. Параметр **frameborder** задает отображение границ текущего фрейма (значение **1** или **yes** включает, а **0** или **no** выключает отображение границ). Если границы отображаются, то их цвет можно задать при помощи атрибута **bordercolor**. Параметр **scrolling** задает появление полосы прокрутки (значение **yes** указывает, что полоса прокрутки присутствует всегда; **no** – никогда; **auto** – появляется автоматически при необходимости). Атрибут **noresize** предотвращает изменение размера фрейма пользователем.

<frameset>

Определяет образующее группу фреймов множество фреймов или других групп фреймов.

```
<frameset cols="30%,*" rows="*" frameborder =  
"YES">  
  <frame src="menu.htm">  
  <frameset rows="60%,*" frameborder="NO"  
border="0" framespacing="0" cols="*">  
    <frame name="topFrame" scrolling="NO"  
src="top_menu.htm" >  
    <frame src="page1.htm">  
  </frameset>  
</frameset>
```



<h1...h6>

Заклученный в теги текст представляет собой заголовок соответствующего уровня (от 1 до 6). Атрибут **align** позволяет задать выравнивание заголовка.

```
<h1>Заголовок 1-го уровня.</h1>  
<h2>Заголовок 2-го уровня.</h2>  
<h3>Заголовок 3-го уровня.</h3>
```

Заголовок 1-го уровня.

Заголовок 2-го уровня.

Заголовок 3-го уровня.

<head>

Определяет заголовочную часть документа.

<hr>

Завершает текстовую строку. Отображает горизонтальную линию и с новой строки продолжает отображение текста.

```
Текст для примера .  
<hr>  
Текст для примера .
```

Текст для примера.

Текст для примера.

<html>

Обозначает начало и конец HTML-документа.

<i>

Заклученный в теги текст отображается наклонным шрифтом.

Пример <i> наклонного </i> текста

Пример <i>наклонного</i> текста

Вставляет изображение в текст. обязательный атрибут **src** указывает имя графического файла. Атрибуты **width** и **height** определяют ширину и высоту отображения в пикселах. если они не указаны, то изображение отображается с оригинальными размерами. Атрибут **align** определяет способ выравнивания изображения относительно текста. Атрибут **alt** позволяет задать альтернативный текст. Толщину рамки (в пикселах) у изображений, содержащихся в гиперссылках, позволяет указать атрибут **border**. Атрибут **usemap** связывает навигационную карту с изображением.

<pre>Текст для примера</pre>



Текст для примера

Ссылка с изображения

<input>

Создает управляющие элементы форм, тип которых определяется атрибутом **type**:

- **button** – настраиваемая кнопка;
- **checkbox** – элемент ввода флажок (флажки работают как переключатели, имеющие состояние "включен" и "выключен", причем одновременно может быть включено несколько элементов);
- **file** – позволяет передавать на обработку вместе с формой внешние файлы;
- **hidden** – элемент формы, не отображаемый браузером (может использоваться для передачи на сервер специальной информации, связанной с формой, которую не может изменить пользователь);
- **image** – позволяет использовать изображение в качестве замены кнопки типа **submit**;

- **password** – элемент ввода текста – вводимый текст представляется таким образом, чтобы не отображать символы (например, в виде ряда звездочек), часто используется для ввода паролей;
- **radio** – кнопка-переключатель, в отличие от элемента **checkbox** в группе переключателей во включенном состоянии может находиться только один (группу составляют элементы, имеющие одно имя);
- **reset** – кнопка, очищающая содержимое элементов формы (или восстанавливающая их значения по-умолчанию);
- **submit** – создает кнопку, которая отправляет информацию, содержащуюся в форме, на обработку серверу;
- **text** – элемент ввода текста.

Атрибут **name** определяет имя управляющего элемента, а **value** позволяет задать его начальное значение.

```
<form name="test" method="post" action="cgi-bin/script.cgi">
  <p><input type="hidden" name="hiddenField" value="mail@mail.ru">
    Текстовое поле: <input type="text" name="textfield"><br>
    Переключатели: <input type="radio" name="radiobutton" value="radiobutton" checked> 1
      <input type="radio" name="radiobutton" value="radiobutton"> 2<br>
    Флажки: <input type="checkbox" name="checkbox" value="checkbox" checked> 1
      <input type="checkbox" name="checkbox2" value="checkbox"> 2
      <input type="checkbox" name="checkbox3" value="checkbox"> 3<br>
    Файл: <input type="file" name="file"><br>
    <input type="image" border="0" name="imageField" src="images/msg.gif" width="18" height="12" alt="Отправить">
    <input type="submit" name="Submit" value="Submit">
    <input type="reset" name="Reset" value="Reset"></p>
</form>
```

<kbd>

В теги заключается текст, который должен ввести пользователь (машинописный текст). Текст отображается шрифтом фиксированной ширины.

Пример `<kbd>машинописного</kbd>` текста

Пример машинописного текста

<label>

Используется для связывания информации с элементами управления. Каждый элемент `<LABEL>` должен быть ассоциирован только с одним элементом управления формы. Атрибут **for** позволяет явно связать метку с другим управляющим элементом: значение этого атрибута должно совпадать со значением атрибута **id** связанного управляющего элемента.

Обозначает элемент списка (используется внутри тегов списков `<OL>` и `<UL>`).

<link>

Этот элемент определяет связь. В отличие от тега `<A>`, он может присутствовать только в разделе `<HEAD>` документа (неограниченное число раз). Элемент не имеет содержимого, он содержит информацию об отношениях.

```
<link rel="stylesheet" href="css/styles.css"
type="text/css">
```

<map>

Содержит описание клиентской карты на изображении. Обязательный атрибут **name** задает имя карты, на которое впоследствии ссылаются в теге `<IMG>`.

<meta>

Обеспечивает предоставление дополнительной информации о документе. Обязательный атрибут **content** определяет значение метаинформации, а атрибут **name** позволяет указать ее имя.

```
<meta http-equiv="Content-Type" content=
"text/html; charset=windows-1251">
```

<nobr>

Запрещает разрывы в заключенном в теги блоке текста.

`<nobr>Текст для примера.</nobr>`

Текст для пр

<noframes>

Определяет содержимое, которое будет выведено на экран браузером, не поддерживающим фреймы.

<object>

Размещает на web-странице объект (апплет, интерактивную анимацию Flash, мультимедийные файлы и т.п.). Атрибут **classid** определяет адрес URL, по которому находится реализация объекта (программа). Атрибут **data** может использоваться для указания местоположения данных объекта. Если в нем задается относительный адрес URL, то он будет интерпретирован относительно атрибута **codebase**, определяющего базовый путь. Параметры **height** и **width** позволяют задать размеры объекта.

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=4,0,2,0" width="100" height="22">  
  <param name=movie value="flash/button1.swf">  
  <param name=quality value=high>  
</object>
```

Back

Определяет начало и конец нумерованного списка, состоящего из элементов . Атрибут **start** позволяет начать нумерацию в списке с определенного номера, а не с 1. Параметр определяет тип нумерации (возможны варианты: **1** – арабские цифры, **A** – заглавные буквы, **a** – строчные буквы, **I** – римские цифры в верхнем регистре, **i** – римские цифры в нижнем регистре).

```
<ol>  
  <li>Элемент списка</li>  
  <li>Элемент списка</li>  
  <li>Элемент списка</li>  
</ol>
```

1. Элемент списка
2. Элемент списка
3. Элемент списка

<optgroup>

Определяет логическую группу тегов <OPTION> (группы не могут быть вложенными). Обязательный атрибут **label** задает метку для группы.

<option>

Определяет вариант в элементе выбора (меню со множественным выбором или прокручиваемом списке). Атрибут **selected** позволяет отобразить элемент как выбранный при первоначальном отображении формы. Параметр **value** позволяет вернуть вместо содержимого тега <OPTION> заданное значение приложению, обрабатывающему форму.

<p>

Заключенный в теги текст представляет собой абзац. Необязательный атрибут **align** позволяет задать выравнивание и может принимать значения **left** (по левому краю), **center** (по центру), **right** (по правому краю), **justify** (по левому и правому краю). По умолчанию для этого атрибута используется значение **left**.

<pre><p>Первый абзац.</p> <p align="center">Второй абзац.</p></pre>
Первый абзац. Второй абзац.

<param>

Позволяет задать параметры в тегах <APPLET> и <OBJECT>.

<pre>

Заключенный в теги текст будет отображаться так, как он был отформатирован предварительно, с точным соблюдением переносов строки и пробельных символов.

<pre><pre>Пример использования предварительно отформатированного текста.</pre></pre>
Пример использования предварительно отформатированного текста.

<q>

Определяет короткую встроенную цитату. Может использоваться с кавычками.

<samp>

Заключенный в теги текст представляет собой название команды, оператора, пример вывода из программ и т.п., и отображается шрифтом фиксированной ширины.

Пример вывода: <code><samp>Hello!</samp></code>
Пример вывода: Hello!

<script>

Предназначен для использования в документе сценария (скрипта), написанного на поддерживаемом языке описания скриптов. Обязательный атрибут **type** задает язык сценария (например, "text/javascript"). Атрибут **src** определяет местоположение внешнего скрипта.

```
<script language="JavaScript" src="js/script.js">
</script>
```

<select>

Определяет меню со множественным выбором или прокручиваемый список. Элементы списка определяются при помощи тега <OPTION>. Обязательный атрибут **name** определяет имя списка, которому будет присвоено значение выбранного элемента. Атрибут **multiple** позволяет пользователю выбирать более одного элемента в списке. Параметр **size** управляет отображением списка. Когда он равен 1 (и атрибут **multiple** не задан), список выводится в виде выпадающего меню, а когда больше 1 – варианты отображаются как прокручиваемый список с заданным числом видимых элементов.

```
<select name="select" size="2">
  <option value="1">Элемент 1</option>
  <option value="2">Элемент 2</option>
  <option value="3">Элемент 3</option>
</select>
```



<small>

Заклученный в теги текст отображается шрифтом меньшего размера.

Пример <small>уменьшенного</small> текста.

Пример уменьшенного текста.

Определяет встраиваемую информацию, не налагая при этом никаких других ограничений для представления контекста. Может использоваться с таблицами стилей, атрибутами **lang** и т.д.

Текст для примера.

Текст для *примера*.

Заклученный в теги текст, требующий более четкого, усиленного выделения, отображается жирным шрифтом.

Пример усиленного выделения.
--

Пример усиленного выделения.

<style>

Позволяет помещать описание каскадных таблиц стилей в раздел <HEAD> документа. Обязательный атрибут **type** определяет язык таблиц стилей для содержимого элемента (например, "text/css"). Вокруг содержимого обычно помещают теги комментариев – это скрывает информацию о стилях от браузеров, которые не распознают тег <STYLE>.

<pre><style type="text/css"> <!-- .selection { font-family: Arial, Helvetica, sans-serif; font-style: italic; color: #00CCFF} --> </style></pre>
--

<sub>

Заклученный в теги текст представляет собой нижний индекс, поэтому смещается вниз и отображается шрифтом меньшего размера.

Пример <sub>нижнего</sub> индекса
--

Пример нижнего индекса

<sup>

Заклученный в теги текст представляет собой верхний индекс, поэтому смещается вверх и отображается шрифтом меньшего размера.

Пример <sup>верхнего</sup> индекса

Пример верхнего индекса

<table>

Определяет начало и конец таблицы. Атрибуты **height** и **width** позволяют задать общую высоту и ширину таблицы соответственно (в пикселах или процентах от размера окна браузера). Параметр **border** задает ширину (в пикселах) границы вокруг таблицы и ее ячеек. Задание **border =0** делает границу невидимой. Атрибут **cellpadding** определяет величину интервала (в пикселах) между границей ячейки и ее содержимым, а **cellspacing** – интервал между соседними ячейками таблицы. Цвет фона таблицы позволяет задать атрибут **bgcolor**, а фоновое изображение определяет атрибут **background**.

<pre><table width="80%" border="1" cellspacing="0" cellpadding="2"> <caption>Заголовок таблицы</caption> <tr></pre>

```

<th width="250">Заголовок столбца 1</th>
<th>Заголовок столбца 2</th>
<th>Заголовок столбца 1</th>
</tr>
<tr>
<td rowspan="2">Ячейка 1</td>
<td align="center" bgcolor="#00CC99">Ячейка
2</td>
<td>Ячейка 3</td>
</tr>
<tr>
<td height="50" valign="top">Ячейка 4</td>
<td height="50">Ячейка 5</td>
</tr>
</table>

```

Заголовок таблицы		
Заголовок столбца 1	Заголовок столбца 2	Заголовок столбца 1
Ячейка 1	Ячейка 2	Ячейка 3
	Ячейка 4	Ячейка 5

<td>

Определяет ячейку таблицы. Ячейка может содержать любую информацию, в том числе и другую таблицу. Атрибуты **width** и **height** определяют ширину и высоту ячейки соответственно. Параметры **align** и **valign** задают способ горизонтального и вертикального выравнивания содержимого ячейки соответственно. Атрибут **bgcolor** определяет цвет фона ячейки. Параметр **colspan** позволяет задать количество столбцов, а **rowspan** – количество строк, которые будет объединять данная ячейка.

<textarea>

Задаёт окно многострочного текстового ввода в формах. Текст, заключённый в теги **<TEXTAREA>**, будет выводиться в указанном поле при первоначальном отображении формы. Обязательный атрибут **name** задаёт имя параметра, которое будет передаваться приложению, обрабатывающему форму. Обязательный атрибуты **rows** и **cols** задают высоту окна текстового ввода количеством строк и его видимую ширину поля ввода в символах соответственно.

```

<textarea name="textfield" cols="30" rows="2">
Текст для примера</textarea>

```

Текст для примера

<th>

Определяет заголовочную часть таблицы. Обычно браузеры выделяют ее содержимое жирным шрифтом и центрируют его по вертикали и горизонтали. Тег имеет те же атрибуты, что и <TD>.

<title>

Каждый документ HTML должен иметь элемент <TITLE> в разделе <HEAD>, который используется для идентификации его содержимого. Название документа обычно появляется в заголовке окна браузера.

<code><title>Заголовок документа</title></code>


<tr>

Задаёт строку ячеек в таблице. Строка не имеет никакого содержания, кроме совокупности ячеек таблицы (<TD>).

<tt>

Форматирует текст как машинописный (при выводе используется шрифт фиксированной ширины).

Пример <code><tt></code> машинописного <code></tt></code> текста.
Пример машинописного текста.

Определяет начало и конец маркированного списка, состоящего из элементов . Атрибут **type** определяет тип маркера элементов списка.

<code></code> <code></code> Элемент списка <code></code> <code></code> Элемент списка <code></code> <code></code> Элемент списка <code></code> <code></code>
• Элемент списка • Элемент списка • Элемент списка

<var>

Заключенный в теги текст представляет собой имя переменной и отображается наклонным шрифтом.

Пример переменной: <code><var></code> x=7 <code></var></code>
Пример переменной: x=7

Приложение Б. Задания для закрепления материала

Основы работы в редакторе Dreamweaver

1. Настройте программу редактора Dreamweaver, установив требуемую кодировку создаваемых Web-страниц.
2. Определите рабочую папку для локального сайта и параметры удаленного доступа к серверу.
3. Создайте HTML-документ в соответствии с образцом, представленным ниже:

Основы работы в HTML-редакторе Dreamweaver

Форматирование текста, создание списка

Создатели программы Dreamweaver постарались сделать удобный и простой в использовании инструмент, облегчающий процесс создания HTML-страниц, их редактирования и управления большим количеством связанных друг с другом документов. В данном учебном пособии рассматриваются основы работы с пакетом Adobe Dreamweaver CS3. Основное внимание уделяется инструментам и приемам работы, позволяющим создавать качественные образовательные ресурсы. Вы научитесь создавать и форматировать web-страницы, использовать интерактивные элементы, познакомитесь со средствами автоматизации разработки и управления файлами сайта.

При организации файлов сайта **рекомендуется** следовать следующим общепринятым принципам:

- в корневом каталоге сайта обязательно должен быть файл *index.html*, который является первой (стартовой) страницей сайта или фреймосодержащим документом;
- графические файлы размещаются в отдельных папках (обычно *images*);
- если имеются дополнительные файлы, используемые как компоненты страниц (например, скрипты JavaScript, стили CSS, анимации Flash), их размещают в отдельных папках в соответствии с типом;
- если сайт содержит несколько тематических разделов, то каждый раздел размещается в своей папке, куда также входят все дополнительные материалы, используемые в этом разделе;
- для документов, входящих в состав сайта, как правило, используют информативные названия, например *links.html*, *page1.html*, *background.gif* и т.д. (а не имена по умолчанию, такие как *Untitled1.htm* или *Movie1.swf*).

4. Сохраните созданный документ в рабочую папку, задав ему имя (например, **text.htm**).
5. Создайте HTML-документ в соответствии с образцом, представленным ниже:

<i>№</i>	<i>Автор</i>	<i>Название</i>
1	Дронов В.А.	Самоучитель Macromedia Dreamweaver 8
2	Кирсанов Д.	Веб-дизайн
3	Пауэрс Ш.	Динамический HTML
4	Уорнер Д., Гарднер С.	Dreamweaver MX 2004 для "чайников"
5	Шафран Э.	Создание Web-страниц: самоучитель
<i>№</i>	<i>Автор</i>	<i>Название</i>

6. Сохраните созданный документ в рабочую папку, задав ему имя (например, **table.htm**).
7. С помощью поисковой системы **Яндекс** найдите картинки среднего размера на любую тематику (например, **животные**). Сохраните несколько картинок в рабочую папку.
8. Создайте HTML-документ в соответствии с образцом, представленным ниже:



Зайцеобразные (лат. *Lagomorpha*) - отряд плацентарных млекопитающих.

Общие сведения

Зайцеобразные распространены в почти всех частях света. Раньше их не было в Южной Америке, на Мадагаскаре, островах Юго-восточной Азии и в Австралии. Однако



благодаря человеческому вмешательству сегодня они в наличии везде, а в Австралии, в связи с их невраждебностью в пищевую цепь распространение кроликов приобрело характер национального бедствия.

Зайцеобразные - звери небольшого размера, с коротким хвостом, либо он отсутствует вовсе. Их зубы имеют некоторое сходство с зубами грызунов. Питание зайцеобразных растительное, в том числе травы, корешки, ветки и кора. Многие виды отличаются чрезвычайно высокой плодовитостью.

Классификация

В отличие от ранних классификаций зайцеобразные не рассматриваются более как часть грызунов, хотя остаётся бесспорным тот факт, что с грызунами у них состоит близкое генетическое родство. Отряд зайцеобразных разделён на два семейства:

- семейство зайцы, *Leporidae*. Населяют практически все части света.
- семейство пищухи, *Ochotonidae*. Населяют Сибирь и Северную Америку.

9. Сохраните созданный документ в рабочую папку, задав ему имя (например, **picture.htm**).
10. Создайте HTML-документ в соответствии с образцом, представленным ниже:

[Ссылка на web-сайт](#) (откроется в новом окне)

[Ссылка на web-страницу](#)

[Ссылка на изображение](#)



11. В качестве **ссылки на web-сайт**, укажите любой интернет-адрес (например, **http://spb.fio.ru**).
12. Задайте открытие **ссылки на web-сайт** в новом окне.
13. В качестве **ссылки на web-страницу**, укажите любой из созданных ранее HTML-документов.
14. В качестве **ссылки на изображение**, укажите любую из сохраненных ранее картинок.
15. Сохраните созданный документ в рабочую папку, задав ему имя (например, **links.htm**).
16. Создайте несколько HTML-документов по образцу, представленному на рисунке:

ЛОГОТИП	ЗАГОЛОВОК СТРАНИЦЫ
страница 1 страница 2 страница 3 страница 4 страница 5	Основной текст страницы

17. Сохраните документы в рабочую папку под разными именами (**index.htm**, **page1.htm**, **page2.htm** и т. д.)
18. В каждом документе, текст *страница 1*, *страница 2*, *страница 3* и т.д. сделайте ссылками на соответствующие страницы сайта.
19. В левом верхнем углу каждой страницы расположите логотип – рисунок отражающий тематику сайта. Картинку-логотип сделайте ссылкой на главную страницу.
20. Разместите созданный Web-сайт на FTP-сервере, обратитесь к нему с помощью программы Internet Explorer и проверьте работу ссылок.
21. Наполните и оформите страницы в соответствии с темой сайта.

Работа в редакторе DREAMWEAVER (дополнительные возможности)

1. Настройте программу редактора DREAMWEAVER, установив требуемую кодировку создаваемых Web-страниц.
2. Определите рабочую папку для локального сайта и параметры удаленного доступа к серверу.

3. Создайте HTML-документ в соответствии с образцом, представленным ниже:

Окружность	
Содержание (маркированный список)	Содержание (нумерованный список)
• Связанные определения • Свойства • Уравнения ◦ Декартовы координаты ◦ Полярные координаты ◦ Комплексная плоскость ◦ Как график ◦ Параметрическое представление • См. также	1. Связанные определения 2. Свойства 3. Уравнения 1. Декартовы координаты 2. Полярные координаты 3. Комплексная плоскость 4. Как график 5. Параметрическое представление 4. См. также
Уравнение:	
На плоскости (O, x_1, x_2) уравнение $(x_1)^2 + (x_2)^2 = R^2$ определяет окружность с радиусом R и центром в точке с координатами $x_1=0, x_2=0$.	

4. В конец документа добавьте таблицу в соответствии с образцом, представленным ниже:

ЛЕВО			ПРАВО		
ВЕРХ	1	1	НИЗ	1	1
	2	2		2	2
	3	3		3	3
	4	4		4	4
	5	5		5	5

5. После таблицы добавьте несколько пустых параграфов и с помощью якоря создайте ссылку на начало документа.
6. Используя действие **Go To URL** создайте HTML-документ в котором щелчок левой кнопки мыши на ссылке открывает интернет-ресурс.
7. Используя действие **Go To URL** создайте HTML-документ в котором двойной щелчок левой кнопки мыши по ссылке открывает новую страницу.
8. Используя действие **Open Browser Window** создайте HTML-документ в котором щелчок левой кнопки мыши по ссылке открывает новое окно, размером 200x100 px, с возможностью изменения размера окна вручную.
9. Используя действие **Open Browser Window** создайте HTML-документ в котором щелчок левой кнопки мыши по ссылке открывает интернет-

ресурс в новом окне, размером 600x400 px, с полосой прокрутки (при необходимости) и пунктами меню.

Шаблоны, CSS-стили, ручное форматирование HTML-кода

1. Настройте программу редактора Dreamweaver, установив требуемую кодировку создаваемых Web-страниц.
2. Определите рабочую папку для локального сайта и параметры удаленного доступа к серверу.
3. С помощью CSS-стилей создайте HTML-документ в соответствии с образцом, представленным ниже (в качестве графического файла используйте любое изображение, найденное и сохраненное из сети интернет; текст страницы может быть произвольным, но по смыслу согласующимся с изображением):

Работа с HTML-редактором Dreamweaver

№	Автор	Название
1	Белунцов В.	Новейший самоучитель по созданию Web-страниц
2	Браун М., Ханикатт Д.	HTML 3.2 в подлиннике
3	Гончаров А.	Самоучитель HTML
4	Исагулиев К.П.	Самоучитель Macromedia Dreamweaver 4
5	Кирсанов Д.	Веб-дизайн

Ячейка1

Ячейка2

Ячейка3



Создатели программы Dreamweaver постарались сделать удобный и простой в использовании инструмент, облегчающий процесс создания HTML-страниц, их редактирования и управления большим количеством связанных друг с другом документов. В данном учебном пособии рассматриваются основы работы с пакетом Macromedia Dreamweaver 4. Вы научитесь создавать и форматировать web-страницы, использовать интерактивные элементы, познакомитесь со средствами автоматизации разработки и управления файлами сайта.

При организации файлов сайта рекомендуется следовать следующим общепринятым принципам:

- в корневом каталоге сайта обязательно должен быть файл *index.html*, являющийся первой (стартовой) страницей сайта или документом с фреймами;
- графические файлы размещаются в отдельных папках (обычно *images*);
- если имеются дополнительные файлы, используемые как компоненты страниц (например, скрипты *JavaScript*, стили *CSS*, анимации *Flash*), их размещают в отдельных папках в соответствии с типом;
- если сайт содержит несколько тематических разделов, то каждый раздел размещается в своей папке, куда также входят все дополнительные материалы, используемые в этом разделе;
- для документов, входящих в состав сайта, как правило, используют информативные названия, например *links.html*, *page1.html*, *background.gif* и т. д. (а не имена, назначаемые по умолчанию, такие как *Untitled1.htm* или *Movie1.swf*).

4. Используя действие **Play Sound** создайте HTML-документ в котором звук проигрывается при наведении курсора мыши на ссылку.
5. Используя действие **Play Sound** создайте HTML-документ в котором щелчок левой кнопки мыши по ссылке проигрывает звук в новом файле с момента загрузки документа.
6. Используя действие **Popur Message** создайте HTML-документ в котором щелчок левой кнопки мыши по ссылке открывает окно сообщения.
7. Используя действие **Set Text of Text Field** создайте HTML-документ с текстовым полем в котором текст появляется при щелчке левой кнопки

мышь внутри текстового поля и исчезает при щелчке левой кнопки мыши вне текстового поля.

8. Используя технологию работы с шаблонами создайте несколько страницы с одинаковой структурой и оформлением.
9. Используя редактор Notepad (Блокнот) или любой другой простой текстовый редактор создать HTML-документ по образцу, представленному ниже (в качестве графического файла можно использовать любое изображение, найденное и сохраненное из сети интернет):

Пример простого документа

Подзаголовок

Проба пера.



	<i>Курсив</i>	Светлый
		Жирный
	Крупный	

Создание тестовых заданий с использованием дополнения CourseBuilder

1. Создайте тест, состоящий из 5 вопросов различного типа с выдачей сообщений о правильности ответов. Вопросы должны находиться в одном HTML-документе, и располагаться друг за другом. Установите ограничение времени для каждого. Организуйте подсчет результата тестирования.
2. Создайте тест, состоящий из 5 вопросов различного типа с выдачей сообщений о правильности ответов. Вопросы должны находиться в одном HTML-документе, и располагаться в таблице в два столбца. Установите ограничение времени для каждого. Организуйте подсчет результата тестирования.
3. Создайте тест, состоящий из 5 вопросов различного типа с выдачей сообщений о правильности ответов. Вопросы должны находиться в одном HTML-документе, и располагаться на отдельных слоях. Установите ограничение времени для всего теста. Организуйте подсчет результата тестирования.
4. Создайте тест, состоящий из 5 вопросов различного типа без выдачи сообщений о правильности ответов. Вопросы должны находиться в разных HTML-документах. Установите ограничение времени для каждого вопроса. Организуйте условные переходы между документами по следующему принципу: при правильном ответе на

вопрос устанавливается переход к файлу со следующим вопросом, при неправильном ответе – к файлу, содержащему информацию об ошибке и объясняющему правильный ответ.

5. Создайте тест, состоящий из 5 вопросов различного типа с выдачей сообщений о правильности ответов. Вопросы должны находиться в одном документе, и располагаться на отдельных слоях. Установите ограничение времени для каждого вопроса. Организуйте подсчет результата тестирования.

Литература

Белунцов В. *Новейший самоучитель по созданию Web-страниц.* М.: Изд-во “ДЕСС КОМ”, 2000, 448 с.

Браун М., Ханикатт Д. *HTML 3.2 в подлиннике.* СПб: BHV-СПб, 1998, 1040 с.

Бурсов М.В., Домненко В.М., Гаврилин Д.А., Николаев Д.Г. *Основы работы с HTML-редактором Dreamweaver. Учебно-методическое пособие.* СПб.: СПбГИТМО(ТУ), 2002. - 104 с.

Гончаров А. *Самоучитель HTML.* СПб: Питер, 2000, 240 с.

Интернет. Энциклопедия. Под. ред. Л. Мелиховой, СПб: ЗАО “Питер Бук”, 2000, 528 с.

Дронов В.А. *Самоучитель Macromedia Dreamweaver 8.* СПб.: БХВ-Петербург., 2006 - 320 с.

Кирсанов Д. *Веб-дизайн.* СПб: Символ-Плюс, 1999, 376 с.

Крейнак Дж., Хебрейкен Дж. *Интернет. Энциклопедия.* СПб: Изд-во “Питер”, 2000, 560 с.

Круг С., Блэк Р. *Веб-дизайн: книга Стива Круга или "не заставляйте меня думать!"* СПб: Символ-Плюс, 2001, 195 с.

Матросов А.В., Сергеев А.О., Чаунин М.П. *HTML 4.0 в подлиннике.* СПб: БХВ-Петербург, 2000, 672 с.

Мейнджер Дж. *JavaScript: основы программирования.* К.: BHV, 1997, 512 с.

Мэтьюз Р.Д., Джонс П., Магид Дж., Болл Д.А., Хаммел М. Дж. *WEB-сервер под UNIX.* СПб: Изд-во “Символ-Плюс”, 1998, 560 с.

Нидерст Д. *Web-мастеринг для профессионалов.* СПб: Изд-во “Питер”, 2001, 576 с.

Нильсен Я. *Веб-дизайн.* СПб: Символ-Плюс, 2000, 450 с.

Пауэрс Ш. *Динамический HTML.* М.: Изд-во ЛОРИ, 1999, 364 с.

Симонович С.В., Евсеев Г.А. *Новейший самоучитель по работе в ИНТЕРНЕТЕ.* М.: Изд-во “ДЕСС КОМ”, 2000, 528 с.

Спейнауэр С., Экштейн Р. *Справочник вебмастера. 2-е издание.* СПб: Символ-Плюс, 2001, 608 с.

Уорнер Д., Гарднер С. *Dreamweaver MX 2004 для "чайников".* М.: Издательский дом "Вильямс", 2004, 352 с.

Хольцнер С. *Dynamic HTML: руководство разработчика.* К.: Издательская группа BHV, 2000, 400 с.

Шафран Э. *Создание Web-страниц: самоучитель.* СПб: Изд-во “Питер”, 2001, 320 с.

Для заметок

Для заметок

**Александр Владимирович Белозубов
Дмитрий Геннадьевич Николаев**

**Основы работы с HTML-редактором
Macromedia Dreamweaver CS3**

Учебно-методическое пособие

В авторской редакции
компьютерный набор и верстка А.В. Белозубов, Д.Г. Николаев
Редакционно-издательский отдел СПбГУ ИТМО
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99
Подписано к печати
Тираж 100 экз. Заказ №
Отпечатано на ризографе.