

Курс лекций по дисциплине «Web-дизайн»

Лекция 1

Тема: Основные понятия Web-дизайна

Цели: *Ознакомить учащихся с основными понятиями web-дизайна; основными задачами web-дизайна в Интернете функции web-дизайнера; определить роль и предмет изучения дисциплины web-дизайна.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Понятие web-дизайна
 - 2) Задачи web-дизайна
 - 3) Функции web-дизайнера
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Понятие web- дизайна

Термин Web-дизайн состоит из двух частей. Первая часть - Web - это сокращенное написание слов World Wide Web - Всемирная Паутина. Вторая часть термина - слово "дизайн" - происходит от английского слова design, что означает: проектировать, конструировать, планировать, чертить, создавать рисунок. Таким образом, предварительный перевод термина Web-design с английского языка на русский язык может быть таким: "проектирование для Интернета".

Для уточнения смысла этого термина рассмотрим существующие понятия и определения.

Дизайн в широком смысле слова - любое проектирование, т. е. процесс создания новых предметов, инструментов, оборудования. Дизайн в узком смысле слова - это художественное конструирование. Его цель - проектирование предметов, в которых форма соответствует их назначению.

Изобразительными средствами дизайна являются: точка, линия, фактура, текстура, цвет, форма, пропорция и др. Кратко рассмотрим следующие термины.

Текстура - характер поверхности какого-либо объекта, обусловленный его внутренним строением, структурой, объективными физическими свойствами (дерево, металл, стекло, ткань).

Фактура - внешние свойства поверхности объекта (гладкая, шероховатая, зеркальная). Фактура характеризует качество обработки поверхности объекта.

Форма - внешнее очертание, наружный вид, контур объекта.

Цвет - свойство тел вызывать определенное зрительное ощущение в соответствии со спектральным составом и интенсивностью отражаемого или испускаемого ими видимого излучения. Основные параметры цвета - цветовой тон,

насыщенность и светлота. Цветовой тон определяется спектральным составом света, насыщенность - количеством добавляемого серого цвета, а светлота - яркостью.

Теперь дадим определение Web-дизайна. Заметим, что в литературных источниках встречаются другие написания этого термина: Веб-мастеринг, Веб-дизайн.

Web-дизайн - новый жанр изобразительного искусства, основанный на применении специфических инструментов (компьютера и программ-редакторов), специальных языков программирования (HTML, JavaScript, Java, Perl, C++, PHP и др.) и сетевых технологий (CGI, CSS, SSI и др.). Целью Web-дизайна является проектирование (создание) объектов (Web-страниц, баннеров, апплетов, скриптов, отдельных элементов Web-страниц), размещаемых в Интернете.

Web-дизайнер (или несколько русифицированные термины: Web-мастер, Веб-мастер) - художник-программист, занимающийся Web-дизайном.

Кроме обязанностей, связанных с дизайном, Веб-мастеру нередко приходится также выполнять обязанности системного администратора и программиста (по крайней мере, иметь хорошее представление о сетевых технологиях и языках программирования). Порой Web-дизайнеру приходится генерировать идеи, воплощать их в реальную жизнь и быть, к сожалению, единственным зрителем своего произведения. В других крайних случаях дизайнеров ждет головокружительный успех, как это случилось с американскими студентами - разработчиками каталога Yahoo.

Начинающие Web-дизайнеры (любители) размещают на своих домашних страничках собственные литературные, графические и музыкальные произведения, результаты научных исследований, фотографии, репродукции любимых картин, кулинарные рецепты, основные сведения о себе, о своих увлечениях и т. п. Это позволяет им отыскать в сети единомышленников, людей со сходными интересами, общаться с интересными людьми.

Более квалифицированные дизайнеры создают сайты, посвященные некоммерческим организациям: учебным заведениям, добровольным объединениям, политическим партиям, любимым спортивным командам и музыкальным группам.

Профессиональные Web-дизайнеры разрабатывают коммерческие проекты, приносящие прибыль: виртуальные магазины, поисковые системы, каталоги, информационные сайты. Профессионально разработанный Web-сайт является одним из видов эффективной рекламы продукции или услуг, удобным средством для поиска деловых клиентов и партнеров, успешного проведения маркетинговой политики.

Разработка серьезного коммерческого проекта требует значительных затрат времени, вложения средств и применения квалифицированного труда. Чаще всего сложные Web-сайты разрабатываются коллективно, так как приходится решать множество сложных технических, изобразительных, информационных, психологических и других задач.

Основной задачей профессиональных Web-дизайнеров является привлечение большого числа посетителей на созданный ими сайт. И если на страницы дизайнеров-любителей заходят два-три посетителя в неделю, то коммерческие сайты посещают ежедневно тысячи людей. Этому способствует неразрывное единство художественного оформления сайта и его предметного содержания (единство формы и содержания). Не последнюю роль в создании популярности играет умение "раскрыть" сайт.

Для создания коммерческого сайта требуются профессиональное мастерство (знание теории, владение навыками), умение выделять актуальные проблемы и хороший художественный вкус. При разработке сайта дизайнеры используют закономерности, накопленные в живописи, графике, скульптуре, архитектуре, художественной фотографии, декоративно-прикладном искусстве, граффити. Однако многие приемы Web-дизайна являются принципиально новыми. Поэтому порой говорят о Web-дизайне как о новом жанре искусства, который опирается на свои изобразительные законы. Эти законы в настоящее время создаются нашими современниками и проходят практическую апробацию в сети.

В отличие от классических произведений изобразительных искусств, в Web-дизайне не используются карандаши, кисти, мольберты, краски. Здесь нужны иные инструментальные средства (HTML-редакторы, графические, видео и звуковые редакторы, аниматоры, редакторы скриптов, специальные утилиты, программы для создания фотоальбомов, разнообразные сетевые технологии, языки программирования, цифровые фотоаппараты, сканеры и др.).

На Web-страницах присутствуют не только текст и статические изображения, но и анимация, видео- и аудиоклипы. Внешний вид просматриваемых Web-страниц динамично изменяется при выполнении пользователем некоторых действий. Это делает Web-страницу сложным произведением искусств, создаваемым на стыке литературы, журналистики, изобразительного искусства, кино, телевидения, радио, фотоискусства. При этом опытный профессиональный разработчик хорошо представляет контингент людей, чаще других посещающих его сайты, учитывает психологию пользователей, подбирает темы, которые наиболее популярны в сети в данный момент времени. Подавляющее большинство Web-страниц является продуктом мультимедиа.

Проектирование страниц или сайтов осложняется еще и тем обстоятельством, что одна и та же страница отображается по-разному разными браузерами (Internet Explorer, Netscape Navigator, Opera) и в окнах разного размера. Во многих случаях браузер самостоятельно "решает" как будет выглядеть загружаемая страница. Изменить внешний вид страницы может и пользователь, например, путем отключения опции загрузки рисунков, анимации.

Для создания динамичных Web-страниц (обладающих элементами искусственного интеллекта) загрузка отдельных элементов производится в зависимости от времени суток или с помощью генератора случайных чисел. Таким образом, изменяются проигрываемые музыкальные клипы, рисунки, загружаемые анекдоты и афоризмы.

При проектировании сайта дизайнеру приходится обращать внимание на правильный выбор цвета шрифта и цвета фона. Рекомендуется отдавать предпочтение синим буквам на белом фоне, черным на желтом, зеленым на белом, черным на белом, белым на синем. При выборе цветовой палитры следует учитывать, что наибольшую чувствительность глаз имеет к желто-зеленому цвету, а наименьшую к фиолетовому и красному цветам. Замечено, что разные цвета оказывают разное воздействие на человека: красный цвет возбуждает, а синий цвет успокаивает.

Данные психологических исследований говорят о том, что человеческий глаз воспринимает красный, оранжевый, желтый цвета (и их оттенки) как теплые. Синий, фиолетовый цвета и их оттенки воспринимаются человеком как холодные. Это

связано, видимо, с тем, что солнце и огонь содержат красные и желтые цвета, а снег, лед - голубые и синие цвета.

Как правило, теплые тона на рисунке обладают свойством "приближаться", а холодные - "удаляться". Другими словами, предметы, нарисованные теплыми тонами, будут казаться расположенными ближе предметам, изображенных холодными тонами.

Специалистами художественной фотографии установлено, что глаз зрителя обращается прежде всего к светлым тонам снимка и к тем местам, где имеется наивысший тональный контраст.

Белый, черный и серый цвета называются ахроматическими, все прочие - хроматическими. Особенности воздействия цвета на человека изучает наука колористика.

Перечислим некоторые важные рекомендации по созданию Web-сайта. Целесообразно не перегружать страницу большим числом деталей. Это связано с тем, что человек за короткое время способен воспринять и запомнить не более 7-8 объектов.

Считается, что чем меньше размер объекта, тем больше должна быть его контрастность. Замечено, что четко выраженный фоновый рисунок на Web-странице повышает утомляемость глаз и снижает эффективность восприятия текста. Для исключения пестроты страницу рекомендуется создавать с использованием не более четырех основных цветов.

На всех страницах одного сайта рекомендуется использовать одинаковые панели навигации, размещенные в одинаковых местах. Каждая Web-страница сайта должна иметь ссылку на главную страницу сайта.

На последней стадии проектирования сайта целесообразно проверить работоспособность всех сделанных гиперссылок. Завершая проектирование сайта, желательно просмотреть его с помощью различных браузеров (MS Internet Explorer, Netscape Navigator, Opera).

Кратко рассмотрим вопрос оценки допустимого объема Web-страницы. В современном динамичном мире каждому человеку нужно успеть многое сделать. Поэтому люди экономят каждое мгновение дарованной им жизни. Пользователи, путешествующие по сети, "проскакивают" мимо тех Web-страниц, которые недопустимо долго загружаются на их компьютеры. По этой причине при проектировании сайта дизайнеру необходимо уметь заранее оценить примерное время загрузки и допустимый объем создаваемой ими страницы. Считая, что предельно допустимое время ожидания загрузки Web-страницы составляет 30-40 секунд, а средняя скорость поступления информации по коммутируемому каналу - 10-20 Кбит/с, получаем желательный объем проектируемой Web-страницы - 300-800 Кбит, т. е. 37,5-100 Кбайт. Естественно, что при использовании выделенного канала скорость загрузки, как правило, возрастает в 3-4 раза, и поэтому допустимый объем страницы в перспективе может быть увеличен.

Краткие итоги

- Web-дизайн - новый жанр изобразительного искусства, использующий собственные средства, закономерности и приемы.
- Web-дизайнер применяет специфические инструменты: компьютер, прикладные программы и языки программирования.

Определения в web-дизайна

Web-дизайнер - это специалист, который занимается проектированием и разработкой web-узла; созданием его структуры, подбором цветовой палитры и разработкой пользовательского интерфейса.

Web-мастер - это специалист, основной задачей которого является поддержка существующего web-сайта. Поддержка заключается в информационном наполнении страниц сайта и в виртуальном взаимодействии с пользователями интернета, которые посещают этот сайт.

Web-сайт - это совокупность web-страниц, объединенных между собой единой композицией, темой информационного содержания, структурой, набором цветов, и находящихся на одном или нескольких серверах.

Web-страница - это текстовый документ, составленный с использованием языка HTML-разметки. Web-страница является простым файлом с расширением html или htm. Этот файл содержит текст и изображения.

Браузер - это прикладная программа, которая позволяет запрашивать, получать с сервера и отображать web-страницы. Самым популярным браузером в настоящее время считается Microsoft Internet Explorer.

Статический web-узел - это web-сайт, содержимое которого фиксировано и не изменяется до тех пор, пока мастер не заменит данные на сайте.

Динамический web-узел - это web-сайт, в котором информация хранится в базе данных, а web-страницы генерируются динамически в ответ на запрос клиента.

Модем - это устройство, позволяющее передавать цифровую информацию по телефонным линиям.

Сервер - это компьютер, подключенный к сети и предназначенный для управления ресурсами. Если он подключен к глобальной сети интернета, то на нем могут быть размещены web-документы, которые передаются в ответ на запросы клиентов.

Электронная почта - это система, предназначенная для обмена электронными сообщениями по глобальной сети интернета.

Сетевой робот - это компьютерная программа, которая сканирует различные web-сайты, извлекает информацию о ключевых словах и сохраняет ее в базе данных поискового сервера.

Поисковый сервер - это компьютерная программа, которая организует поиск по ключевым словам в глобальной сети интернета либо на конкретном web-узле. В ответ на запрос пользователя поисковый сервер возвращает список ссылок на документы. В качестве примера поискового сервера можно назвать "yandex" и "rambler".

Доменное имя - это название web-сайта, представленное в виде символов на английском языке. Например, www.designer.ru, www.site.com и другие.

Протокол интернет (IP) - это протокол передачи информационных пакетов, который лежит в основе функционирования глобальной сети интернета.

Web-палитра - это набор из 216 цветов. Web-ориентированная палитра гарантирует, что цветное изображение будет отображаться в неизменном виде большинством браузеров, работающих на разных платформах.

Протокол - это набор правил, регламентирующих обмен информацией между компьютерами в сети. В качестве примеров сетевых протоколов можно привести IP, HTTP и FTP.

Гипертекст - это информационная структура, которая обеспечивает навигацию между web-документами посредством гипертекстовых ссылок. Фрагменты текста или

изображения, при активации которых отображаются связанные с ними документы, называются гипертекстовыми ссылками.

3. Вопросы для закрепления.

- 1) Что такое «web-дизайн»?
- 2) Дайте понятие «web-страница»
- 3) Назовите допустимый объем web-страницы.
- 4) Назовите основные рекомендации при создании web-страниц

4. Домашнее задание: Подготовить доклад на тему: «История развития сети Интернет»,
«Основные web-технологии»

Лекция 2

Тема: Технологии web-дизайна

Цели: *Ознакомить учащихся с технологическими и художественными решениями применяемых в современном web-дизайне.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Логическая и физическая структура сайта
 - 2) Особенности верстки документов сайта.
 - 3) Виды модульных сеток
3. Закрепление изученного.
4. Итог урока.

Ход урока.

4. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

5. Изложение нового материала.

Логическая и физическая структура сайта

Каждый ресурс Интернета, от любительской домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой странице, которой присваивается имя `index.htm` (`.html`). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся `index.html`.

ПРИМЕЧАНИЕ Такое имя файла рекомендуется присваивать всем стартовым документам сайта, поскольку в противном случае при обращении к какому-либо разделу посредством сокращенного URL без указания названия стартовой страницы (например, `http://www.mysite.ru/photos/` вместо `http://www.mysite.ru/photos/startpage.html`) браузер отобразит не саму web-страницу, а перечень хранящихся в данной папке файлов.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется логической структурой сайта. Физическая структура, напротив, подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован ваш сайт. Пример сравнения логической и физической структур одного и того же ресурса Интернета показан на рис. 2.1.

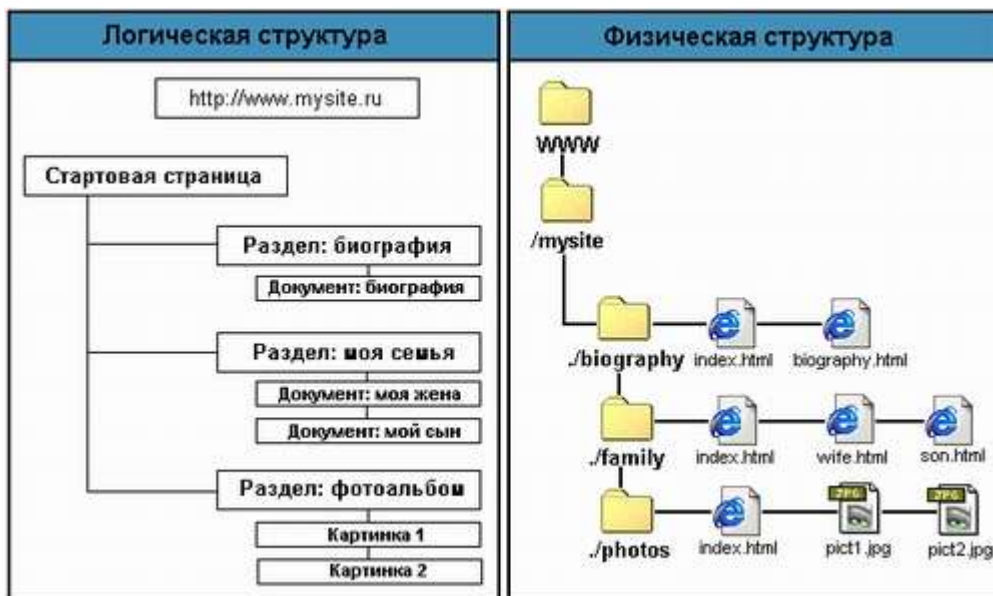


Рис. 2.1. Сравнение логической и физической структуры сайта

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволит вам избежать путаницы при последующем дополнении и обновлении материалов.

СОВЕТ Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием "Images", расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории.

Для того чтобы все гиперссылки на вашей домашней страничке или web-сайте работали корректно, все документы открывались правильно и браузер не выдавал ошибок при обращении к каким-либо разделам ресурса, при создании его физической структуры следует соблюдать несколько простых правил.

СОВЕТ Назначайте имена директорий, имена и расширения документов HTML и графических файлов с использованием символов только латинского алфавита и только в строчном регистре. Старайтесь, чтобы имена созданных вами файлов и директорий не превышали по длине восьми символов.

СОВЕТ При присвоении имен файлов документам HTML старайтесь следить за тем, чтобы эти имена были "смысловыми": впоследствии вы легко можете забыть содержимое и назначение какой-либо web-страницы, если имена файлов будут выглядеть, например, как 1.htm, 2.htm, 3.htm и т. д.

Для того чтобы облегчить процесс обновления web-страниц, дополнения разделов или создания новых рубрик, заведите средство документирования проекта - любую электронную таблицу, созданную, например, в Microsoft Excel, или просто разграфленную тетрадку, в которую записывайте соответствие элементов физической структуры вашего проекта его логической структуре. До тех пор пока количество составляющих ваш сайт файлов относительно мало, это может показаться излишним, когда же оно перевалит за первые два десятка, в обилии html-документов и графических элементов будет легко запутаться, особенно если вы создаете несколько

проектов одновременно. Пример оформления такого средства документирования показан в табл. 2.2.

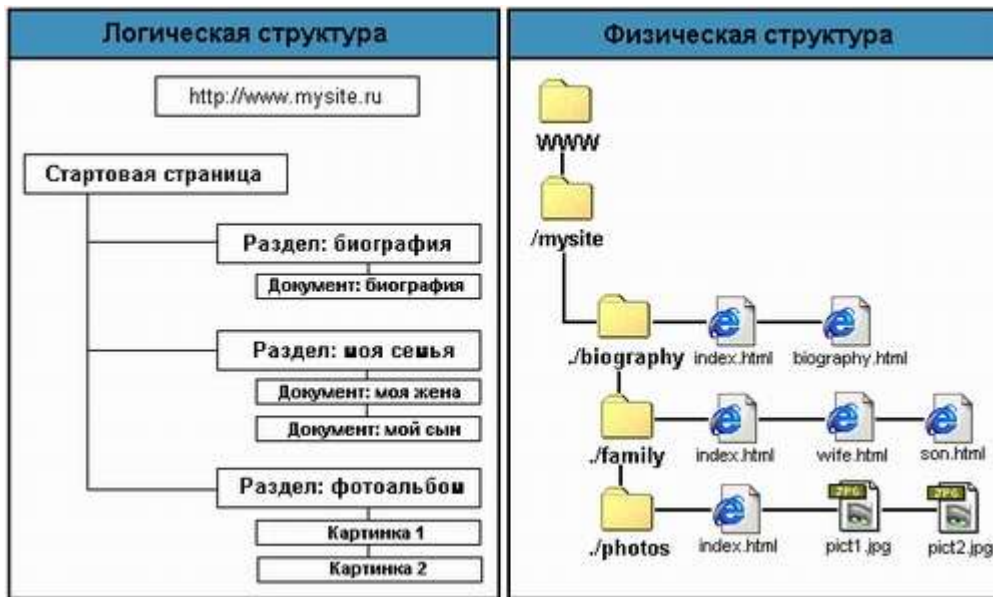


Таблица 2.2. Пример оформления средства документирования проекта

Имя файла	Директория	Описание	Дата создания/последнего изменения
index.html	/mysite	Стартовая страница сайта http://www.mysite.ru	1.07.2000
index.html	/mysite/family	Стартовая страница раздела <моя семья>	5.07.2000
wife.htm	/mysite/family	Рассказ о моей жене	5.07.2000
son.htm	/mysite/family	Рассказ о моем сынишке	5.07.2000
pid-jpg	/mysite/photos	Моя фотография на пляже в Сочи	6.07.2000

Из всего сказанного становится очевидным, что физическая структура сайта скрыта от посетителей вашего ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации. Отсюда следует вполне логический вывод: строение системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной вами логической структуре сайта.

Заглавная страница

Один из критериев, руководствуясь которым можно разделить различные сайты на две основные категории, - это наличие заглавной страницы (splash) или отсутствие таковой.

Заглавная страница представляет собой html-документ, который не включает в себя какую-либо содержательную информацию и элементы навигации. Файлу заглавной страницы присваивается имя index.html, при этом стартовая страница называется иначе и вызывается посредством организации гиперссылки с заглавной страницы, загружающейся при обращении к сайту первой. Заглавная страница содержит, как правило, логотип компании-владельца данного ресурса, счетчик

посещений и предложение выбора кодировки кириллицы, либо выбора между английской и русской версиями сайта. Пример web-ресурса, оснащенного такой страницей, показан на рис. 2.3.

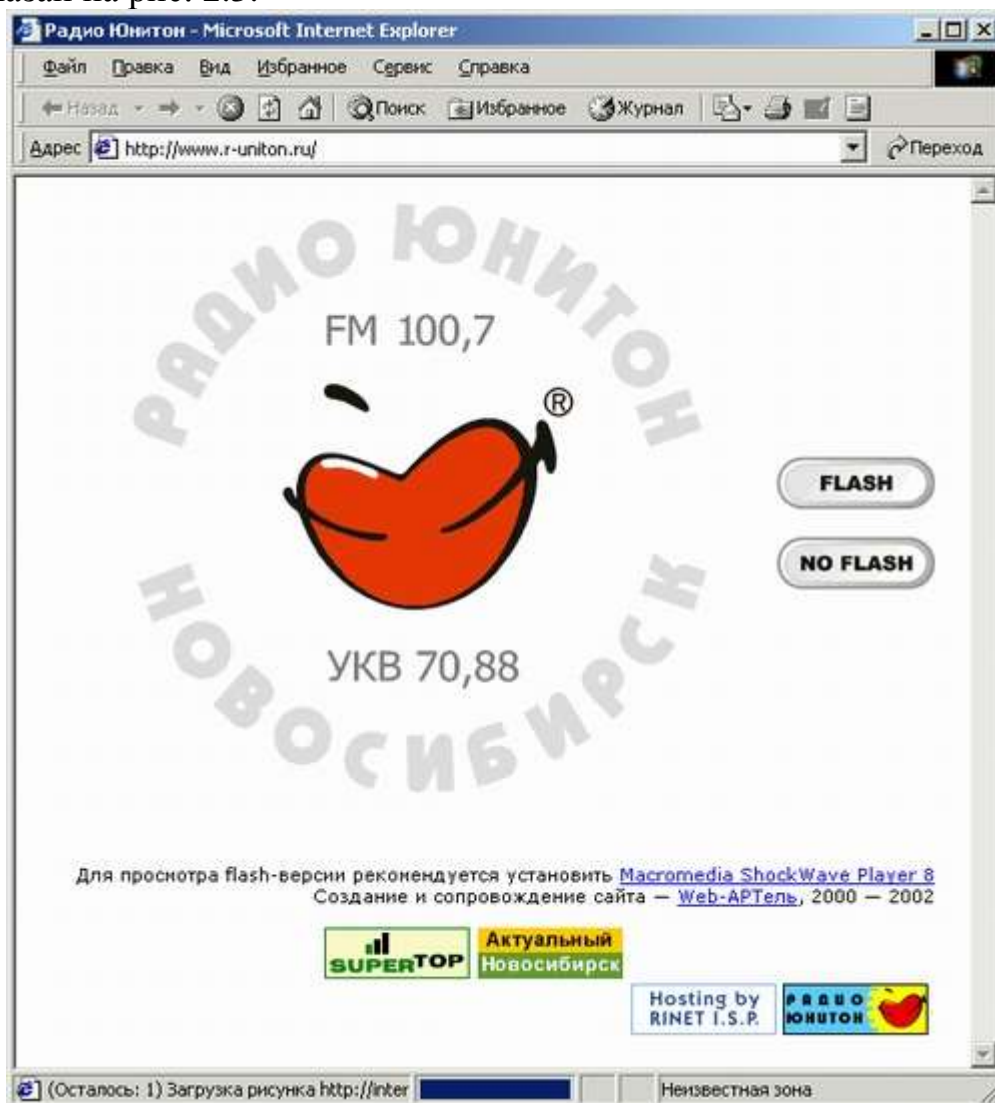


Рис. 2.3. Пример сайта, оснащенного заглавной страницей

При обращении к сайтам, не оснащенным заглавной страницей, первой отображается стартовый документ, включающий какое-либо информационное наполнение, элементы навигации и иногда анонсы составляющих данный ресурс тематических рубрик. На рис. 2.4 показан пример сайта, не оснащенного заглавной страницей.

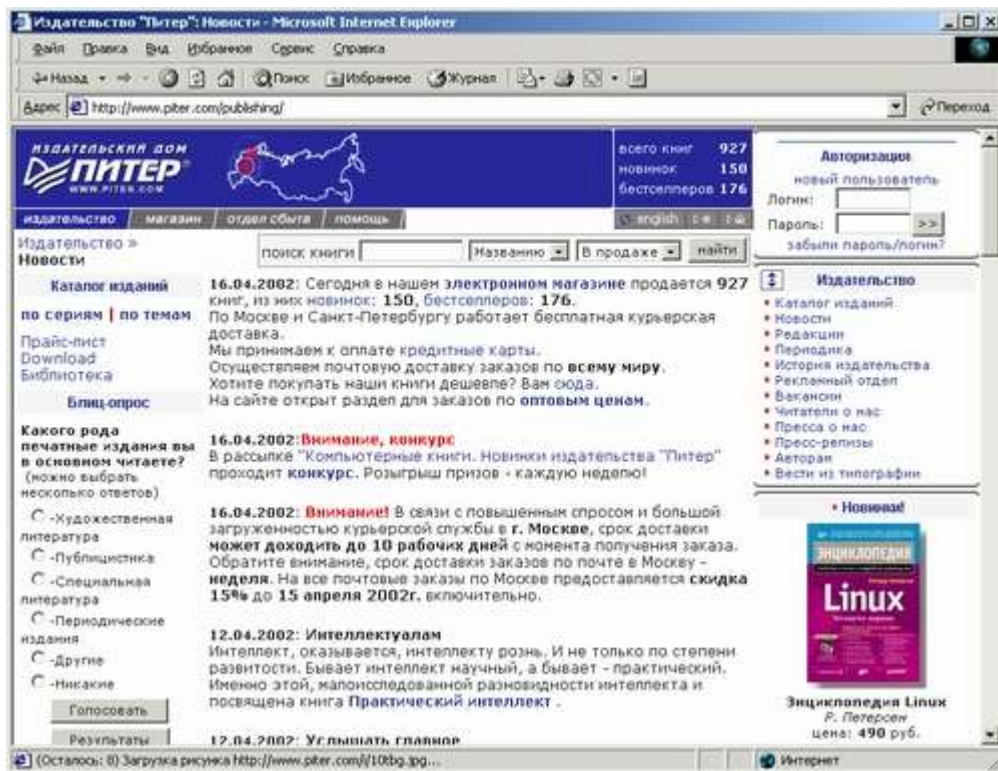


Рис. 2.4. Пример сайта, не оснащенного заглавной страницей

Использовать или не использовать заглавную страницу при создании собственного проекта в сети Интернет - дело вкуса каждого web-мастера. Дать какие-либо исчерпывающие рекомендации на этот счет трудно, поскольку окончательное решение зависит прежде всего от ваших художественных предпочтений и иногда - от желания заказчика, оплачивающего вашу работу.

Верстка web страниц

Веб-страница запускается на компьютере под управлением клиентской программы называемой браузером. Понятно, что операционная система, ее настройки и собственно сам браузер отличается от компьютера к компьютеру. Из чего напрашивается банальный вывод, что один и тот же документ сайта по-разному отображается у каждого пользователя.

Напрашивается вопрос, можно ли сделать так, чтобы веб-страница отображалась одинаково? Учитывая, сколько придется принять во внимание разных неоднозначных факторов, следует сказать, что нет, нельзя. Поэтому задача верстки веб-страниц формулируется так: сформировать документ, который бы корректно отображался с небольшими различиями на основных платформах и в браузерах. Корректно означает, что соблюден исходный замысел автора, воплощен требуемый дизайн документа, и он показывается в браузере без ошибок.

Чтобы реализовать указанную задачу надо понимать, как вообще происходит верстка веб-страниц и соизмерять свои идеи с их исполнением.

Особенности верстки документов сайта.

Ширина документа

Изначально разработчику сайта ширина окна браузера пользователя неизвестна, поскольку она может меняться в самых широких пределах. Ширина зависит от разрешения монитора, длины его диагонали, размера окна и еще некоторых варьируемых данных. Иными словами предугадать ее заранее простыми средствами

не представляется возможным. С учетом этой особенности утвердилось два способа верстки: фиксированный и «резиновый».

Фиксированный макет

В данном случае действуем от обратного и устанавливаем общую ширину макета жестко заданной и равной определенной величине. Если взять некоторую обобщенную статистику посетителей сайтов и посмотреть, какое разрешение монитора они преимущественно используют, то узнаем, что это 800 x 600 и 1024 x 768 пикселей. Получается, что ширина монитора пользователей в основном 800 и 1024 пиксела. Возьмем за ориентир 800 пикселей, тогда общая ширина макета за вычетом вертикальной полосы прокрутки и рамки браузера окажется 770–790 пикселей. На этот размер ориентируемся и устанавливаем ширину макета, например 770 пикселей.

Преимущество такой схемы следующее. Раз общая ширина макета точно известна, то мы можем легко подгонять под нее дизайн и делать изображения уже известной ширины. В целом подобная верстка приближается к верстке печатного буклета, и в том и другом случае ширина носителя информации строго задана, за счет чего верстка хоть частично, но упрощается.

Недостаток, который инкриминируют этому виду верстки фактически один — недостаточно эффективное использование свободной площади. Действительно, для монитора с большо-о-о-й диагональю или высоким разрешением экрана документ будет смотреться по-другому, чем на предполагаемых 800 пикселах. На рис. 2.5 показано, как выглядит макет в таком случае.

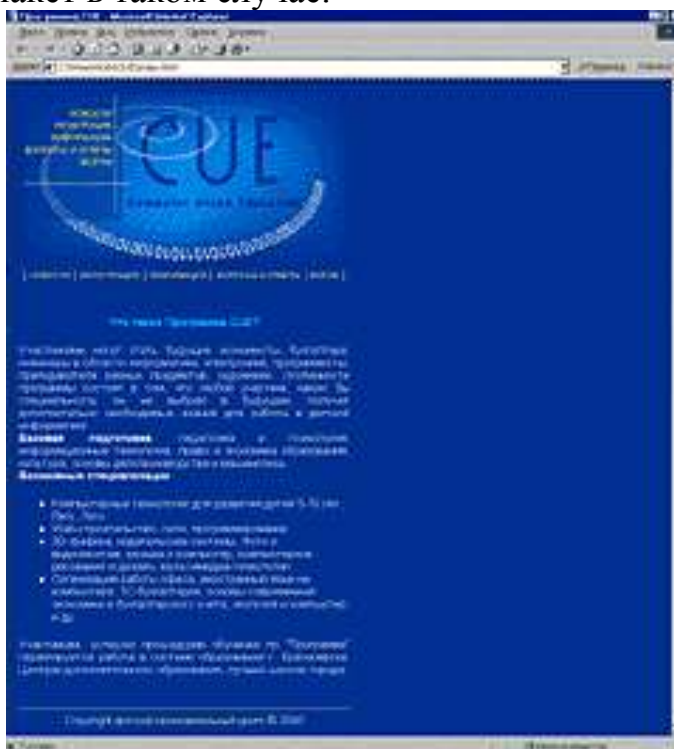


Рис. 2.5. Макет фиксированной ширины с размещением по левому краю

Справа появляется широкая пустая полоса, размер которой зависит от разрешения экрана пользователя и диагонали его монитора. Чтобы хоть как-то уменьшить пустое пространство, макет обычно помещают по центру окна браузера (рис. 2.6).



Рис. 2.6 Макет фиксированной ширины с размещением по центру «Резиновый» макет

Этот вид макета основывается на том, что в качестве одной из единиц измерения выступают проценты. Общая рабочая ширина окна браузера — 100%, и колонки макета в сумме не должны ее превышать, поэтому для удобства, как правило, везде применяют процентную запись. При изменении размеров окна происходит переформатирование данных страницы, чтобы они вписались в новую ширину (рис. 2.7).



Рис. 2.7. Веб-страница занимает всю отведенную ширину

Этот вид верстки набирает все большую популярность и практически все известные сайты выбрали именно его в силу того, что эффективно задействуется вся

площадь веб-страницы. Но следует сразу отметить некоторые особенности и недостатки присущие «резиновой» верстке.

- Хотя веб-страница и подстраивается под ширину окна браузера, при достижении некоторой величины читать текст становится неудобно — строки слишком длинные и глаза устают по ним бегать. Впрочем, браузер можно свернуть в окно, подобрав его комфортный размер.

- Верстать «резиновый» макет сложнее, чем аналогичный, но фиксированной ширины. Это связано с тем, что приходится учитывать множество дополнительных факторов и знать некоторые приемы верстки. К тому же, популярные браузеры неоднозначно трактуют некоторые параметры и в них «резиновый» макет может отображаться по-разному. Говоря проще, знать о правилах верстки надо больше.

- Любой макет имеет некоторую минимальную ширину, при достижении которой веб-страница «рассыпается» или появляется горизонтальная полоса прокрутки. Например, если в документ вставлен рисунок шириной 600 пикселей, то при уменьшении окна до этой величины браузер начнет отображать полосу прокрутки.

- «Резиновый» дизайн характеризуется активным использованием фоновых изображений, которые по горизонтали собираются без швов встык. Действительно, изменить без потери качества ширину рисунков не получится, а вот «подложить» под них фон — всегда пожалуйста.

Высота документа

Исторически сложилось, что пролистывание большого документа на компьютере происходит сверху вниз. Для удобства листания предназначены вертикальные полосы прокрутки, клавиатурные комбинации, колесо прокрутки мыши. А вот перемещение по горизонтали происходит не так удобно, поэтому горизонтальной полосы прокрутки быть не должно. Из чего следует, что веб-страница должна вписываться в окно браузера по ширине, но никак не по высоте, которая может изменяться в очень широком диапазоне. На рис. 2.8 видно, что реальная высота документа не ограничена рамками браузера, но в окне показывается лишь часть страницы.



Рис. 2.8. Отображение документа в окне браузера по высоте

В принципе, чем больше на странице информации, и, следовательно, больше высота документа, тем сложнее находить нужные данные. Поэтому текст структурируют, разбивают на блоки и каждому из них дают свой заголовок, чтобы взгляду читателя было за что зацепиться.

Также следует учесть, что объем статей на сайте может достаточно сильно различаться между собой. При этом будет меняться и высота страницы, поэтому следует заранее побеспокоиться о том, чтобы макет отображался без ошибок, несмотря на различное значение высоты.

Объекты веб-страницы прямоугольны

Все объекты на веб-странице имеют прямоугольную форму. Этот простой постулат не всегда согласуется с тем, что мы видим в действительности, поэтому следует сделать пояснения. В отличие от традиционной верстки (речь идет о полиграфических материалах), где в документ можно вставлять любые объекты, в том числе и векторные фигуры произвольной формы, верстка веб-страниц имеет ряд ограничений. К числу ограничений относится и то, что добавляемые объекты прямоугольны. Причем это относится к их форме, но не к содержимому, благодаря чему требуемый нам дизайн можно конструировать с помощью набора изображений. На рис. 2.6 показана картинка весьма неправильной формы. Но из-за того, что фон у этого рисунка совпадает с фоном веб-страницы, исходная прямоугольность изображения не видна.



Рис. 2.6. Изображение на рисунке может иметь произвольную форму

Однако стоит добавить вокруг рисунка рамку, как становится понятно, что изображение все-таки прямоугольно (рис. 2.7). Например, если включить обтекание картинки текстом, то он будет обходить ее именно по границе рамки.



Рис. 2.7. Но сам рисунок остается прямоугольным

Данная особенность породила некоторые техники связанные с версткой, которые перечислены далее.

Активное использование рисунков

Рисунки не только применяются для иллюстрации текста, но и выполняют на сайте много различных ролей, например, используются для создания привлекательного дизайна, служат распоркой между ячейками таблицы, создают градиентные заливки, фоновые изображения и т.д.

Разрезание изображения на фрагменты

Один рисунок может занимать слишком большую исходную площадь. Разрезав его на прямоугольники, получим замечательный конструктор, в котором одни фрагменты изображения допускается заменять текстом, другие анимацией, а третьи модифицировать «на лету». Таким образом, имеем в своем распоряжении средство для обхода прямоугольной природы изображений, ведь в «склеенном» рисунке может не хватать некоторых фрагментов, уголков, например.

Применение фонового рисунка

Фоновый рисунок удобен тем, что он может заполнять всю отведенную ширину или высоту под блоком. Это позволяет создавать линии или другие декоративные элементы, которые привязываются к ширине или высоте текста и не зависят от размеров окна. К тому же поверх фона можно накладывать текст, что также расширяет возможности по дизайну веб-страниц.

Картинки вместо текста

Если средства верстки имеют определенные ограничения, то почему бы не создать текст в графическом редакторе и не вставить его в качестве картинки или Flash? Это гарантирует, что текст сохранит свой вид и начертание несмотря ни на какие внешние условия. Однако здесь имеется и обратная сторона — рисунки занимают больший объем, чем рядовой текст, их сложнее править, они не индексируются поисковыми машинами, их показ пользователи могут отключить. В общем, недостатков масса, поэтому изображения на сайте хотя и применяют вместо текста, но достаточно ограниченно. Например, для заголовка сайта.

МОДУЛЬНАЯ СЕТКА

Модульная сетка представляет собой набор невидимых направляющих, вдоль которых располагаются элементы веб-страницы. Это облегчает размещение данных в документе, обеспечивает визуальную связь между отдельными блоками и сохраняет преемственность дизайна при переходе от одной страницы к другой.

Веб-страница фактически рассматривается как набор прямоугольных блоков, которые выкладываются в определенном порядке. При этом, как правило, данные располагаются по колонкам, поэтому при верстке применяют термин одно-, двух-, трехколонный макет и т.д. Для примера рассмотрим главную страницу сайта deviantart.com (рис. 2.8).

Каждый блок этой страницы четко отделен от других с помощью пустого пространства, рамки или разделителя, в качестве которого выступает цветной прямоугольник с текстом заголовка. Напрашивается только вопрос, действительно ли мы имеем дело с тремя колонками? В некоторых случаях сразу определить, сколько колонок содержит макет и впрямь затруднительно. В таком случае следует понимать, что колонки могут объединяться, а также содержать не только сплошной текст, но и графические вставки. Если представить основные блоки страницы в виде однотонных прямоугольников, то получим наглядную модульную сетку, по которой сразу становится понятно, как сверстан документ (рис. 2.9).

По данному рисунку видно, что верхний блок с названием сайта, формой поиска и кнопками навигации занимает всю ширину страницы. Далее идут три колонки, причем первые две колонки предварительно объединены в одну для удобного представления необходимой информации. Завершает макет блок с контактной и правовой информацией.

Перейдем к принципам построения модульной сетки. Вначале макет веб-страницы разрабатывают на листе бумаги. Это позволяет, не тратя зря времени, быстро сделать серию набросков и уже из них выбрать подходящий эскиз. Сами посудите, сколько вариантов можно создать за десять минут в графическом редакторе и сколько за это же время с помощью карандаша и бумаги? При этом не имеет значения степень владения программой, все равно на бумаге выйдет быстрее. Тем более что набросок может быть и корявым, главное чтобы автор сам понял, что он нарисовал. Обычно вместо текста и рисунков применяют схематические значки. Например, текст обозначается несколькими горизонтальными линиями (рис. 2.10), а рисунки изображаются затемненными блоками или перечеркнутыми прямоугольниками (рис. 2.11).



Рис. 2.8. Главная страница deviantart.com

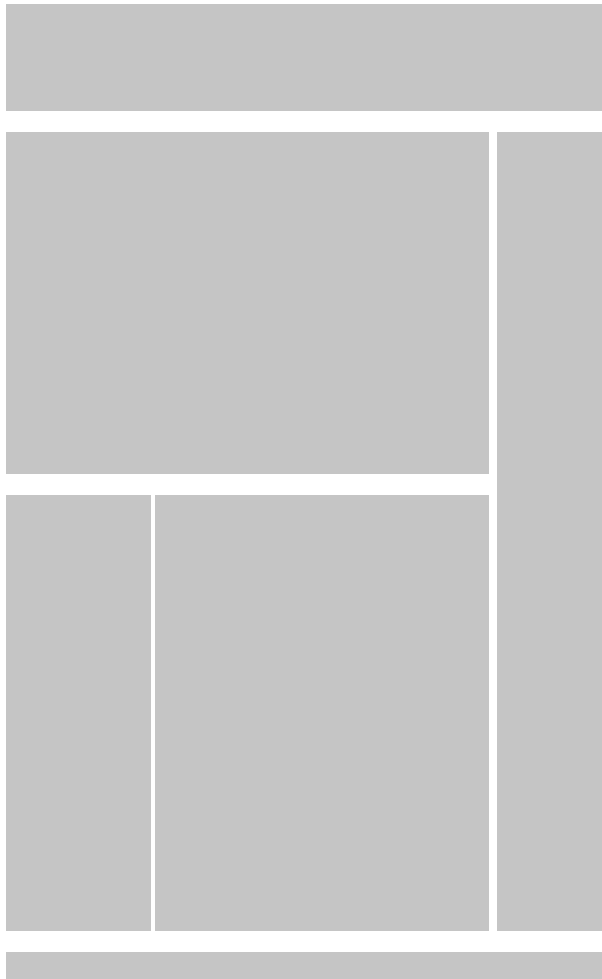


Рис. 2.9. Модульная сетка для главной страницы deviantart.com



Рис. 2.10. Обозначение текста в макетах

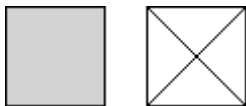


Рис. 2.11. Обозначение изображений в макетах

Далее эти обозначения будем применять при рассмотрении наиболее распространенных модульных сеток.

Одноколодная сетка

Текст в одну колонку чаще всего встречается в академическом дизайне, при фиксированном макете и публикации большого текста.

Замечание

Академический дизайн характеризуется минимализмом оформления и даже аскетизмом. Основной упор делается на содержательную часть, а дизайну как таковому практически не уделяется внимания. Преимущественно академический дизайн встречается в научной среде.

На рис. 2.12 показана типичная схема одноколодной модульной сетки. Как правило, наблюдается четыре основных блока — заголовок страницы, набор ссылок на другие страницы сайта (навигация), собственно сам текст и в самом низу

контактная информация. Если высота страницы достаточно велика, то блок навигации дублируют внизу или делают ссылку «Наверх», которая перемещает к началу документа.



Рис. 2.12. Одноколонная модульная сетка

Иллюстрации в тексте встречаются по ходу, при этом текст их обычно обтекает по контуру. При активном применении изображений на сайте удобнее воспользоваться фиксированным макетом, ширина которого точно известна. Тогда рисунки можно готовить заданного размера, которые точно впишутся в макет страницы фиксированной ширины и практически полностью состоит из набора рисунков.

Модульная сетка для этого сайта представлена на рис. 2.13. Вначале идет заголовок сайта с формой поиска, ниже следует навигация, а отдельные фотографии представляют собой саморекламу со ссылками на соответствующие разделы сайта. В самом низу расположены ссылки на информацию о сайте.

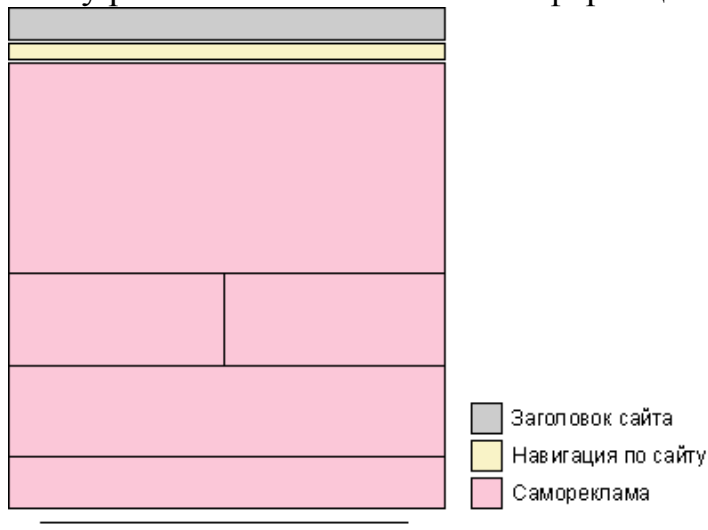


Рис. 2.13. Модульная сетка для главной страницы сайта victoriasecret.com

Двухколонная сетка

Это один из самых распространенных вариантов при использовании на сайтах. При такой модульной сетке используется две колонки — одна отводится под основной текст, а вторая используется для навигации и другой полезной информации (рис. 2.14).



Рис. 2.14. Двухколонная модульная сетка

Принципиального значения не имеет, слева или справа располагается колонка с навигацией, встречается и тот и другой вариант.

В каком-то смысле двухколонная сетка стала стандартом де-факто для информационных сайтов из-за своего удобства. Действительно, все «под рукой» — и текст и ссылки, к тому же данная сетка не исключает подключения горизонтальной навигации, как это принято в одноколонной сетке.

Двухколонные сетки удобны при создании самых разнообразных сайтов и не требуют особых знаний по верстке веб-страниц. Единственный недостаток, который им вменяют, что подобные сайты выглядят достаточно однообразно. Но с другой стороны пользователям удобнее работать с сайтом привычного вида, без лишних «наворотов».

Трехколонная сетка

Такие сетки часто применяются на главных страницах сайтов, где одновременно требуется показать пользователю множество возможностей, которые он обнаружит на данном сайте. Также трехколонная сетка используется и на внутренних страницах, если для размещения различной информации двух колонок уже не хватает (рис. 2.15).

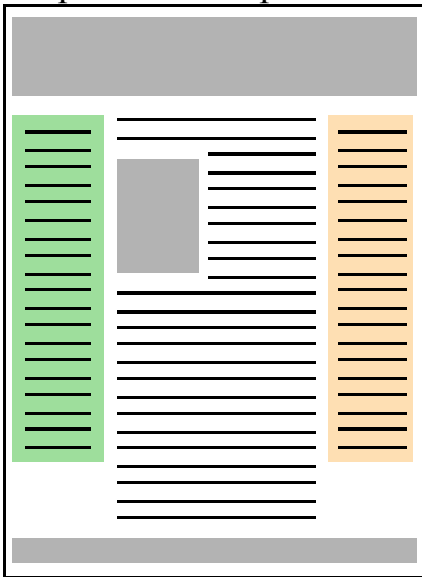


Рис. 2.15. Трехколонная модульная сетка

Одна из колонок отдается под навигацию, вторая, самая широкая — под основной текст, а в третью колонку добавляют рекламу, ссылки, текст и т.д.

Трехколонная сетка обеспечивает больше простора для дизайна, ведь в некоторых местах можно объединять колонки, разбивать материал на отдельные

фрагменты и визуально отделять один блок от другого. Макет при этом может получиться достаточно сложным, но результат обычно того стоит. Возвращаясь к главной странице сайта deviantart.com, приведем более детальную модульную сетку (рис. 2.16). Отдельные блоки выделены разным цветом.

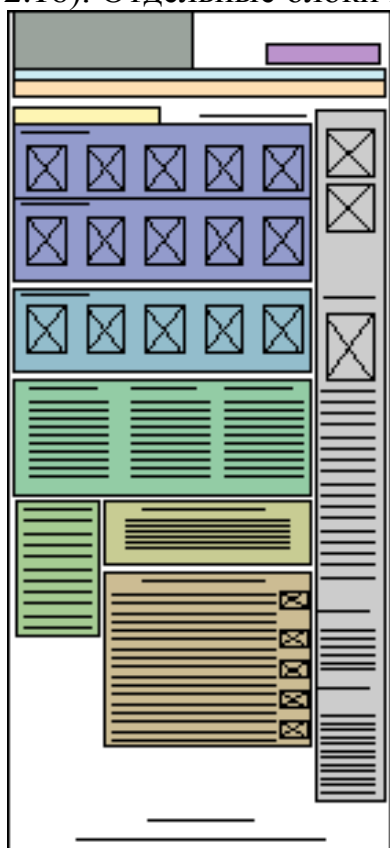


Рис. 2.16. Модульная сетка для главной страницы deviantart.com

На сайте deviantart.com применяется три колонки, две из них часто объединяются для получения более широкой области. Это оправданно, поскольку требуется разместить 5 фотографий или три колонки с текстом.

К недостаткам трехколонной сетки относится достаточная сложность верстки макета. Чтобы получить нужный результат приходится иной раз затратить много времени на создание стилевого файла и отладку документа в разных браузерах.

Конечно, существуют и другие виды модульных сеток, например, многоколонные. Однако они практикуются достаточно редко из-за ограниченной ширины окна браузера. При таком раскладе текст приходится делать мелким или применять другие способы, чтобы не возникло горизонтальной полосы прокрутки. От этого может пострадать удобство восприятия информации пользователем.

Модульная сетка не является единственным верным вариантом при верстке веб-страниц. Можно воспользоваться перспективой, хаотичностью или другой структурой, главное, чтобы она была эстетичной и привлекательной. Отказ от правил верстки характерен для дизайн-студий, задача которых состоит в том, чтобы поразить своим сайтом воображение посетителей и привлечь к себе их внимание. Но даже они делают для заказчика сайты на основе формальных сеток, поскольку так проще добавлять информацию и вносить разнообразие в дизайн макета.

Модульные сетки позволяют упростить верстку сайта, поскольку все материалы разбиваются на отдельные блоки, которые выравниваются по невидимым направляющим линиям. Такие блоки хотя и взаимосвязаны друг с другом, но обычно

дозволяют независимое редактирование данных, что упрощает оформление элементов.

Прежде чем переходить к созданию веб-страницы в HTML-редакторе, сделайте ее набросок на листе бумаги. Это позволит сократить время на выбор модульной сетки, которая будет применяться в дальнейшем. В зависимости от целей сайта и количества материала применяют одно-, двух, трех или многоколодную модульную сетку. Чем больше колонок применяется, тем шире возможности по управлению видом дизайна, но за это приходится платить сложностью верстки документа. Не стоит также забывать, что наиболее популярными на сайтах являются двух и трехколоночные сетки.

6. Вопросы для закрепления.

- 1) Чем отличается логическая и физическая структура сайта
 - 2) Дайте понятие фиксированному макету.
 - 3) Дайте понятие «резиновому» макету.
 - 4) Какие виды модульных сеток вы изучили?
- 7. Домашнее задание:** Составить схему (модульную сетку) для будущего сайта.

Лекция 3

Тема: Принципы компоновки и алгоритм создания web- сайта

Цели: *Ознакомить учащихся с принципами компоновки сайта: статической и динамической;; основными компонентами web – сайта.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Принципы компоновки сайта
 - 2) Динамический сайт
 - 3) Статический сайт
 - 4) Компоненты сайта
3. Закрепление изученного.
4. Итог урока.

Ход урока.

8. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

9. Изложение нового материала.

Принципы компоновки web- сайта

Современные видеокарты поддерживают несколько произвольно устанавливаемых видеорежимов, характеризующихся экранным разрешением и количеством цветов, используемых при отображении информации на экране компьютера. С помощью встроенных функций операционной системы пользователь может установить несколько стандартных значений экранного разрешения, например 640x480, 800x600, 1024x768, 1152x864, 1280x1024 или 1600x1200 точек. Разумеется, при открытии в одном и том же браузере какой-либо web-страницы она может отображаться не одинаковым образом в зависимости от используемого посетителем сайта экранного разрешения.

Для того чтобы избежать "съезжания" элементов html-документа друг относительно друга и, как следствие, деформации web-страницы в целом при изменении параметров экрана, применяется достаточно простой и действенный прием: все компоненты web-страницы заключаются в соответствующие ячейки невидимой таблицы, при этом каждому объекту назначается одно, строго определенное положение. Таким образом, появляется второй критерий, по которому можно разделить все существующие web-сайты на две условные категории. Данной таблице можно назначить строго определенную ширину в пикселях, например, 640 точек, после чего жестко позиционировать ее по центру экрана или "прижать" к левому его краю.

Для построения сложных таблиц можно применять вложение одной таблицы в другую: например, первичная таблица шириной 640 точек задает горизонтальное разбиение страницы, в каждую из ее ячеек вкладывается еще одна, равная ей по ширине невидимая таблица, обеспечивающая вертикальное разбиение. Та- кой подход позволяет дробить различные участки web-страницы независимо друг от друга, создавая, к примеру, в одном ее поле четыре вертикальные колонки различной ширины, а в другом — две, оперируя при этом различными параметрами вложенных

таблиц в пределах одного и того же документа — размером пространства между колонками, толщиной границ и т. д.

Такой вариант компоновки сайта можно назвать статическим, поскольку ширина таблицы не меняется в зависимости от экранного разрешения. Разумеется, при изменении параметров экрана не происходит ни малейшего смещения элементов дизайна страницы.

Иной подход - когда ширину невидимой таблицы, содержащей фрагменты web-страницы, задают в процентах от текущей ширины экрана. При увеличении экранного разрешения таблица "растягивается" по горизонтали, и все размещенные в ее ячейках элементы, позиционированные либо по центру, либо по краям столбцов, смещаются согласно установленному алгоритму. В силу того, что параметры таблицы изменяются в зависимости от настроек экрана, такой принцип компоновки html-документа можно назвать динамическим. И тот и другой подход обладает как достоинствами, так и недостатками, которые перечислены ниже.

Статическая компоновка страницы

Достоинства. Простота алгоритма верстки документа. Оптимизировав страницу для отображения с экраным разрешением 640x480 точек, вы можете быть уверены, что при изменении пользовательских экранных настроек элементы дизайна не "поплывут". Кроме того, данный вариант компоновки сайта в большинстве случаев (при соблюдении ряда дополнительных условий) позволяет добиться идентичности отображения html-документа в браузерах MicrosoftInternet Explorer и Netscape Navigator.

Недостатки. При отображении документа на компьютере с высоким экраным разрешением по краям экрана или с правой его стороны остается заметное пустое поле.

Динамическая компоновка страницы

Достоинства. Документ растягивается по всей ширине экрана, не остается неиспользованных пустых полей.

Недостатки. Сложность верстки и отладки страницы, весьма часто проявляется неадекватность отображения таких документов в браузерах Microsoft Internet Explorer и Netscape Navigator.

ПРИМЕЧАНИЕ Используемые в командах HTML значения пикселей (условных точек) подразумевают физические размеры точки экрана пользователя и зависят от типа его монитора.

Примеры реализации сайтов со статической и динамической компоновкой страниц приведены на рис. 3.1 и 3.2. Данные изображения сделаны с фиксированным экраным разрешением 800x600 точек.

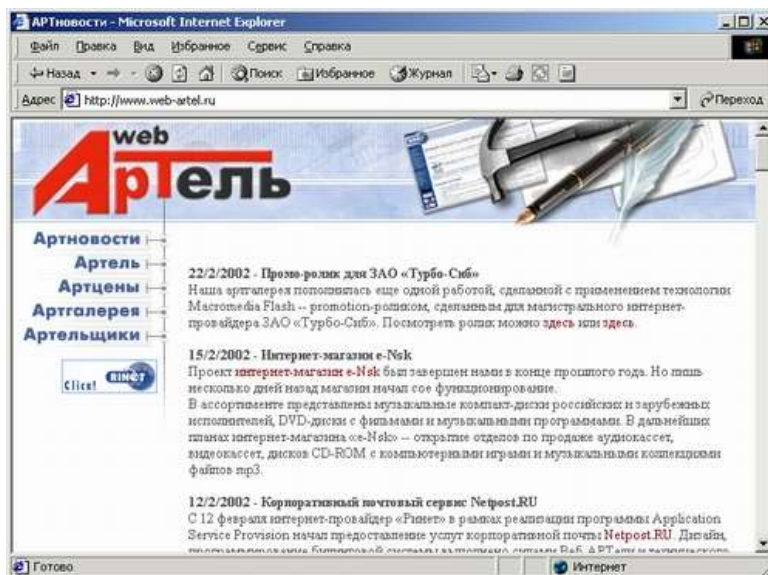


Рис. 3.1. Пример сайта со статической компоновкой страниц



Рис. 3.2. Пример сайта с динамической компоновкой страниц

Элементы web-страницы

Любая web-страница содержит определенный набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета. Безусловно, ассортимент и количество подобных объектов могут варьироваться в зависимости от тематической направленности сайта, объема опубликованных на нем материалов, а также от целей и задач, которые ставит перед собой создатель данного ресурса. Компоновка таких элементов, проектирование их взаимного расположения и составляет одну из главных задач web-мастера.

Первым элементом web-страницы, который нам предстоит рассмотреть, является ее заголовок. Он может быть выполнен как в текстовом, так и в графическом варианте, однако и в том и в другом случае он должен располагаться в верхней части документа. Иногда с заголовком совмещают меню выбора кодировки кириллицы и кнопки для перехода с русскоязычной на англоязычную версию сайта, если данный web-ресурс представлен на двух языках. Непосредственно подзаголовком документа, как правило, располагается пространство, отведенное для размещения рекламного баннера. Включение баннера именно в верхнюю часть web-страницы в большинстве случаев является обязательным условием регистрации сайта в службах баннерного

обмена - системах, рекламирующих созданный вами ресурс в обмен на показ на страницах вашего сайта рекламы других участников баннерообменной сети. Стандартный размер баннеров, публикуемых под заголовком документа, составляет обычно 468x60 точек. Если вы используете статический принцип компоновки страницы, ширина заголовка вашего документа будет составлять приблизительно 640 пикселей: это значение обусловлено, прежде всего, необходимостью обеспечить корректное отображение документа на мониторах с экранным разрешением 640x480 точек и избежать появления горизонтально полосы прокрутки, затрудняющей его просмотр. Очевидно, что ширина баннера в этом случае будет значительно меньше ширины заголовка, благодаря чему в той части страницы, где вы планируете отвести место под рекламу, образуется незанятое пространство, которое можно заполнить логотипом компании-владельца данного сайта или ссылкой на сервер, осуществляющий web-хостинг. Разумеется, логотип необходим далеко не всегда: как правило, он включается в состав web-страницы лишь в случае, если сайт имеет коммерческую направленность.

Основную часть документа занимает так называемое текстовое поле - участок, где и размещается смысловое наполнение страницы: содержательный информационный текст и иллюстрации. Перечисленные элементы еще называют "контент" (от англ. content - содержание). Расположение текстового поля зависит в первую очередь от того, каким образом web-дизайнер разместит остальные элементы документа.

Следующей обязательной составляющей частью web-страницы являются элементы навигации - гиперссылки, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, например Java-апплетов. Последние представляют собой те же кнопки, которые, в отличие от своих "традиционных" сестер, умеют реагировать на движения мыши, выполняя при наведении на них курсора какие-либо несложные действия (включение подсветки, создание эффекта "нажатия", изменение формы и т. д.). Как я уже упоминал в разделе "Основные постулаты" web-дизайна, располагать элементы навигации следует таким образом, чтобы они всегда были "на виду", "под рукой", то есть так, чтобы пользователю не приходилось "отматывать" страницу назад, если текстовое поле занимает по высоте несколько физических экранов, после чего подолгу искать ссылки на другие разделы. Наиболее устоявшимся подходом является размещение элементов навигации у левой границы страницы.

В нижней части документа принято публиковать информацию о разработчиках сайта и адрес электронной почты, по которому посетители ресурса могут направить владельцам странички свои отклики, предложения и пожелания. Если web-страница является стартовым документом, в нижней ее части также размещают счетчик посещений - небольшой сценарий, вызывающий установленный на сервере CGI-скрипт, который фиксирует каждое открытие документа в браузере пользователей, изменяя значение индикатора счетчика. Благодаря этому web-мастер без труда определит количество посетителей, навестивших его страничку в течение какого-либо времени. Отмечу, что счетчик посещений устанавливается только на первой странице, вызываемой при обращении к сайту, в остальных документах ресурса он отсутствует. Не рекомендуется также размещать на одной странице несколько разных счетчиков.

Итак, мы разобрали все основные компоненты web-страницы и их возможное расположение относительно друг друга. Пример компоновки сайта, содержащего полный набор описанных выше составляющих, показан на рис. 3.3. В нем выбрано позиционирование элементов навигации по левой границе документа.



Рис. 3.3. Пример компоновки web-страницы с левым позиционированием элементов навигации

На практике часто встречаются web-сайты, в дизайне которых элементы навигации позиционированы по правой границе экрана. В этом случае текстовое поле смещается налево, остальные компоненты документа располагаются, исходя из принципа максимальной эстетичности их сочетания. Пример такого исполнения сайта показан на рис. 3.4.



Рис. 3.4. Пример компоновки web-страницы с правым позиционированием элементов навигации

Как видно из рисунка, логотип в этом случае помещен на один уровень с заголовком документа, а рекламный баннер позиционирован относительно центра страницы. При таком подходе рекомендуется выдерживать графическое оформление заголовка, логотипа и поля для размещения рекламы в едином цветовом и художественном стиле - тогда несимметричность положения данных объектов будет не столь очевидна и не станет "резать глаз" сторонникам строгой, "табличной" эстетики дизайна.

Элементы навигации можно разместить не только вблизи правой и левой границ страницы, но и в верхней части документа. Такой вариант компоновки наиболее подходит, на мой взгляд, при создании домашних страничек: в этом случае все объекты страницы гармонично "вписываются" в заданную ширину невидимой таблицы, при этом подготовка самой таблицы значительно упрощается. Единственным недостатком подобного подхода является необходимость продублировать элементы навигации в нижней части документа, поскольку при вертикальной прокрутке страницы они исчезают за верхней границей экрана, и, чтобы добраться до них, пользователю придется "отматывать" экран назад, что, согласитесь, весьма неудобно. Пример оформления страницы с верхним расположением элементов навигации показан на рис. 3.5.



Рис. 3.5. Пример компоновки web-страницы с верхним позиционированием элементов навигации

Безусловно, все, что было сказано в данном разделе, является не панацеей, а руководством к действию. Я пытаюсь изложить лишь общие принципы, которые применяются при компоновке структуры сайта, окончательное же решение всегда остается за web-мастером. В конечном счете, какой бы дизайн вы не заложили в основу своего будущего проекта, результат ваших трудов все равно будет правильным: в Интернете нет ни цензуры, ни каких-либо регламентов, загоняющих создателя сайта в те или иные жесткие рамки. Примером дизайнерского решения, не попадающим ни в одну из указанных выше категорий, может служить так называемая смешанная компоновка, примерная схема которой приведена на рис. 3.6.

Как видно из рисунка, в данном примере часть управляющих элементов встроена непосредственно в заголовок страницы - речь идет о кнопках переключения между русской и английской версиями сайта, а также о кнопках навигации: это может быть ссылка на адрес электронной почты создателей ресурса, продублированная в нижней части документа, и ссылка на один из тематических разделов, например, страницу новостей. Основной блок элементов навигации позиционирован относительно левой границы документа, однако меню выбора кодировки кириллицы расположено непосредственно под рекламным баннером в верхней части страницы. Текстовое поле разделено на две несимметричные колонки, причем в правой размещены краткие анонсы составляющих ресурс тематических рубрик, включая ссылки на эти разделы.

Очевидно, что вариантов "смешанной компоновки" web-страницы может быть великое множество: конкретные решения зависят от количества составляющих ресурс разделов, объема подготовленного для размещения на сайте текста и, наконец, от фантазии самого дизайнера. Важно лишь, чтобы внешний вид сайта не вызывал нареканий у посетителей. В конце концов, только вы как разработчик, вы и никто другой, вправе проявить все свои способности и таланты и скомпоновать страницу по собственному вкусу. Создатели некоторых домашних страничек, не мучаясь сомнениями, размещают счетчик посещений в верхнем правом углу документа, название сайта пишут мелким убористым шрифтом и публикуют его под рекламным баннером, а элементы навигации почему-то неожиданно обнаруживаются прямо в середине текстового блока, между рассказом о себе и фотографиями любимой собаки автора проекта. На вкус и цвет, как говорится, товарищей нет. Но лично мне эта болезнь кажется неизлечимой.



Рис. 3.9

10. Вопросы для закрепления.

- 1) Что такое динамическая компоновка сайта?
- 2) Что такое статическая компоновка сайта?
- 3) Назовите основные элементы web-страницы.
- 4) Назовите классификацию web-сайтов по признакам их компоновки.

11. Домашнее задание: Подготовить рефераты на тему: «Типы веб-дизайна: текстовый, динамический, конструктивный, полиграфический, смешанный. Характеристики.
«Правила профессионального веб-дизайна»

Лекция 4

Тема: Графические изображения. Особенности веб-графики

Цели: *Ознакомить учащихся с основными понятиями графических форматов; возможностями оптимизации графических изображений;*

План урока.

1. **Организационный момент.**
2. **Изложение нового материала.**
 - 1) Виды графических форматов их особенности и отличия
 - 2) Оптимизация графических объектов
3. **Закрепление изученного.**
4. **Итог урока.**

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Графика для web

Оформление **Web-страниц** подразумевает необходимость размещения максимально возможного объема информации на сравнительно небольшом пространстве. Вы должны исходить из того, что среднестатистический пользователь работает в Интернете с разрешением экрана не более 800x600. Область, которая отображается в окне браузера при таком разрешении экрана, составляет примерно 790x430 пикселей. Чтобы представить величину этой области, вспомните, что значки на рабочем столе Windows имеют размер 32x32 пиксела.

Подготавливая графику для Web-страницы, необходимо помнить о небольших размерах области, с которой вы должны работать. Изображение не должно превышать 700 пикселей по ширине и 400 пикселей по высоте. Но файл даже такой картинке может оказаться слишком большим для передачи по каналам связи сети. Следует также помнить и о том, что часть пользователей могут видеть на своих мониторах только 256 цветов. Поэтому 24-битная графика, способная обеспечить 16,7 миллиона оттенков цвета, не сможет быть оценена некоторой частью зрителей. При этом пользователи не только не смогут увидеть на своих мониторах всей заложенной вами цветовой гаммы, но, кроме того, изображение может оказаться худшего качества, чем в случае использования 8-битной графики, отображающей 256 цветов. К тому же файл окажется в три раза больше по размеру, так как, его размер пропорционален глубине цвета, и, соответственно, будет требовать большего времени для загрузки на экран.

Использование корректной цветовой гаммы важно при разработке любого графического проекта. Для **графики Web-страниц необходимо применять цвета модели RGB. Так как для Web-страниц устройством вывода графики является экран монитора, то здесь, как правило, соблюдается условие WYSIWIG - What You See Is What You Get (Что видишь, то и получаешь).** Говоря «как правило», мы имеем в виду небольшие индивидуальные отличия мониторов пользователей.

На вопрос «**Каков должен быть размер графического файла?**», предназначенного для Web, трудно ответить однозначно. Давайте попробуем представить себе, что пользователь, обращающийся к вашему сайту, имеет модем с

пропускной способностью 14,4 Кбит/сек, хотя подобные модемы и считаются устаревшими. При использовании такого модема передача блока данных объемом 1 килобайт занимает примерно 1 секунду, так что при объеме графики в 30 килобайт для ее передачи по сети потребуется приблизительно 30 секунд. После таких расчетов возникает следующий вопрос: как долго посетитель, обратившийся к вашему сайту, готов ждать появления на экране загружаемого изображения? Если вы сможете ответить на этот вопрос, то сразу же определите и максимальный размер файла с графическим изображением. Считается, что файл должен иметь такой размер, при котором время его загрузки по сети не превышает одной минуты, что выражается в ограничении размера величиной примерно 60 килобайт.

Размер графического файла, в частности, полученного в результате сканирования изображения, можно определить по следующей формуле:

$$F = (A/25,4 \times B/25,4) \times D^2 \times C/8,$$

где

F - размер файла без сжатия в байтах;

A - ширина исходного изображения в мм;

B - длина исходного изображения в мм;

D - графическое разрешение изображения в dpi;

C - глубина цвета в битах.

Таким образом, если вы хотите отсканировать для своей Web-страницы фотографию

размером 10x15 см с разрешением 150 dpi и глубиной цвета 24 бита, то результирующий

файл будет иметь размер 1569378 байт или 1,49 мегабайта.

Любая компьютерная информация может храниться только в определенном формате. Каждый вид информации имеет собственные форматы. Для текста используются одни форматы, для электронных таблиц - другие, для графики - третьи. Формат графической информации обычно определяется программой, в которой она создана. Для **Web-графики стандартными считаются форматы JPEG, GIF и PNG.**

Формат JPEG (Joint Photographic Experts Group - Объединенная группа экспертов фотографии) - самый популярный формат для хранения фотографических изображений. JPEG является стандартом в Интернете и предназначен для сохранения растровых или точечных изображений со сжатием, уменьшающим размер файла от десятых долей процента до 100 раз, за счет отбрасывания избыточной информации, не влияющей на отображение документа. Но практически диапазон сжатия значительно уже: от 5 до 15 раз. Хотя алгоритм сжатия изображения этого формата и ухудшает качество, он может быть легко настроен на минимальные, практически незаметные для человеческого глаза, потери. Распаковка JPEG-файла происходит автоматически, во время его открытия. Между качеством изображения и степенью уплотнения существует обратная зависимость: чем более высокую степень сжатия для результирующего изображения вы зададите, тем хуже будет его качество.

Существует разновидность формата JPEG, именуемая «прогрессивный JPEG» (p-JPEG). Прогрессивный JPEG отличается от обычного тем, что при выводе на экран изображение появляется почти сразу, но плохого качества, а по мере загрузки качество улучшается. В формате JPEG можно хранить только полноцветные изображения с глубиной цвета 24 пиксела. Существует также версия JPEG для 32-

битных изображений, но она пока мало распространена. Обычно формат JPEG рекомендуют использовать для фотографий, на которых снижение качества, сопутствующее сжатию по этой технологии, наименее заметно.

Формат GIF (Graphics Interchange Format - Формат для обмена графической информацией) был разработан специалистами компьютерной сети CompuServe в 1987 году с целью ускорения пересылки файлов по телефонным линиям. Требовалось создать такой формат, который бы содержал достаточно хорошо упакованные данные. Алгоритм распаковки должен был быть при этом встроен во все программы, которые будут работать с этим форматом: утилиты просмотра, графические редакторы и т.п. Именно такой подход и реализован в формате GIF.

Хотя с тех времен пропускная способность модемов увеличилась во много раз, требования к компактности файлов, передаваемых по сетевым каналам, отнюдь не отошли на второй план. Web-страница, которая со всем своим текстом и графикой занимает свыше 40-50 килобайт, уже считается «большой» - и далеко не каждый посетитель дождется, пока она целиком загрузится в его браузер. Профессионализм Web-дизайнера как раз и заключается в том, чтобы отыскать должный баланс между художественностью, информативностью страницы и ее объемом. Вот почему формат GIF, главным достоинством которого является малый размер файлов, до сих пор сохраняет свое значение основного графического формата World Wide Web. Из распространенных графических форматов GIF однозначно уступает по степени сжатия только формату JPEG.

Конечно, GIF - не единственный формат, позволяющий сжимать графические данные; кое-какие способности к сжатию есть у форматов BMP, PCX, TIFF и многих других. Однако GIF - один из немногих форматов, использующих алгоритм сжатия, почти не уступающий по эффективности программам-архиваторам. Иными словами, GIF-файл не нужно архивировать, так как это редко дает ощутимый выигрыш в объеме.

Как и у программ-архиваторов, степень сжатия графической информации в формате GIF сильно зависит от уровня ее повторяемости и предсказуемости, а иногда еще и от ориентации картинка. Поскольку GIF сканирует изображение по строкам, то, к примеру, плавный переход цветов - градиент, направленный сверху вниз, сожмется куда лучше, чем тех же размеров градиент, ориентированный слева направо, а этот последний - лучше, чем градиент по диагонали.

В отличие от формата JPEG, который позволяет достичь серьезного выигрыша в размере при минимально заметных для глаза потерях в качестве, формат GIF пользуется обычным сжатием без потерь. С другой стороны, преимуществом JPEG по сравнению с GIF является возможность хранить полноцветные изображения с 16 миллионами цветов, тогда как GIF ограничен лишь 256-цветной палитрой.

В 1989 году появилась новая, усовершенствованная версия формата GIF, в которой используется «постепенный» показ изображения по мере его загрузки из сети. Эта технология заключается в том, что при показе специально записанных GIF-файлов строки изображения выводятся на экран не подряд, а в определенном порядке: сначала каждая 8-я, затем 4-я и т.д., то есть полностью изображение показывается в четыре прохода. Такая система показа позволяет понять, что изображено на картинке еще до того, как она полностью загрузится, что очень важно на медленных линиях связи. Нечто подобное используется в телевидении для передачи одного полного кадра за два прохода луча - так называемая «чересстрочная развертка». Поэтому такие

изображения были названы чересстрочными (interlaced). Приняв из сети очередной проход чересстрочного GIF-файла, браузер не только рисует каждую его строку на своем месте, но и заполняет копиями предыдущей строки все промежутки между строками, иначе после первого прохода изображение было бы почти невидимым. Поэтому сначала чересстрочный GIF состоит из широких горизонтальных полос, которые постепенно сужаются по мере проявления изображения.

Формат PNG (Portable Network Graphics - Переносимая сетевая графика) поддерживается поздними версиями браузеров, например, Netscape Navigator, начиная с версии 4, но широкого распространения пока не получил. Этот формат, основанный на вариации алгоритма сжатия без потерь качества, в отличие от GIF сжимает растровые изображения и по горизонтали, и по вертикали, что обеспечивает более высокую степень сжатия; кроме того, он поддерживает цветные графические изображения с глубиной цвета до 48 бит включительно.

PNG реализует гораздо более эффективный алгоритм чересстрочности. Достаточно сказать, что первый проход, дающий общее представление об изображении, занимает в PNG не одну восьмую исходного файла, как в GIF, а всего лишь одну шестьдесят четвертую — и, тем не менее, распознаваемость картинки при этом заметно лучше.

PNG позволяет хранить полную информацию о степени прозрачности в каждой точке изображения в виде так называемого альфа-канала. **Каждый пиксел PNG-файла, вне зависимости от его цвета и местоположения, может иметь любую градацию прозрачности:** от нулевой - полная непрозрачность, до абсолютной невидимости.

Чтобы сгладить различия в яркости дисплеев, существующие между разными типами компьютеров и, соответственно, между создаваемыми на них изображениями, PNG позволяет хранить в файле исходный коэффициент яркости того дисплея, на котором изображение было создано. Это дает возможность Программам просмотра перед выводом на экран регулировать яркость изображения, приспособив ее к особенностям данной видеосистемы - так называемая гамма-коррекция.

PNG-формат не дает возможности создавать анимационные ролики. Поэтому для хранения анимации или нескольких изображений в одном файле используется вариация формата PNG - формат MNG (Multiple Network Graphics - Множественная сетевая графика). В настоящее время разработчики формата PNG работают над новым форматом PNP(пинап) (Portable Network Photography - Переносимая сетевая фотография), произносится «пинап», который предназначен для хранения фотографических изображений и обладает мощным алгоритмом сжатия с потерей качества.

Возможности оптимизации

Чтобы любое изображение можно было использовать на Web-странице, его необходимо предварительно оптимизировать, уменьшив до минимума размер файла и, следовательно, время загрузки по сети так, чтобы качество изображения при этом заметно не ухудшилось. Оптимизация это, по сути, компромисс между размером файла и качеством изображения.

В формате JPEG изображение сохраняется с глубиной цвета 24 бит и используется сжатие с потерей информации. У него не так уж много способов оптимизации, точнее сказать, два:

- использование прогрессивной развертки;
- оптимальный коэффициент сжатия.

Использование прогрессивной развертки не сказывается на размере изображения, но заметно при загрузке. В случае записи изображения в стандартном формате вывод на экран осуществляется горизонтальными полосами, и до загрузки всего изображения невозможно определить его смысл. Напротив, при записи файла в прогрессивном формате изображение появляется сразу целиком, но в грубой форме, и затем постепенно улучшается. Это дает посетителям сайта возможность сразу же оценить изображение и решить, стоит ли ожидать его окончательной загрузки, что позволяет сэкономить немного времени при просмотре страниц. Общее время загрузки страницы при этом не уменьшается, возникает лишь иллюзия его уменьшения. Прогрессивная развертка JPEG поддерживается всеми браузерами, но не каждый графический пакет может записывать в этом формате. Последние версии Photoshop умеют это делать.

Оптимальный коэффициент сжатия. Это - другая возможность оптимизации изображения в формате JPEG. Для осуществления такой оптимизации нужен графический пакет или утилита, которая позволяет регулировать степень сжатия изображения. Adobe Photoshop позволяет выполнить такую регулировку. Другие программы, например, Photo- Paint и PaintShopPro, также позволяют плавно регулировать степень сжатия в интервале значений от 1 до 255.

Как определить оптимальную степень сжатия? Для этого нужно сохранить изображение при нескольких значениях коэффициента сжатия, а затем внимательно рассмотреть его. Критерием потери качества будут так называемые «дрожащие контуры», наиболее заметные в местах контрастных переходов, и появление пятен в областях с плавными переходами. Тот коэффициент сжатия, при котором искажения еще не видны, но с увеличением коэффициента на одну ступеньку становятся заметными, и будет оптимальным. Размер файла при таком коэффициенте сжатия будет минимально возможным при сохранении качества.

Формат GIF сильно отличается от JPEG и использует совершенно другие методы оптимизации. Он предназначен для хранения изображений с количеством цветов не более 256 и использует сжатие без потерь по методу LZW, который применяется в программах-архиваторах. Методы оптимизации можно разделить на следующие типы, исключая оптимизацию файлов с анимацией:

- уменьшение количества цветов;
- оптимизация палитры изображения;
- стилизация изображения;
- изменение размеров изображения;
- фрагментарная оптимизация;
- оптимизация «прозрачных» изображений;
- использование чересстрочной развертки.

Теперь разберемся по порядку с этими методами и приемами оптимизации.

Уменьшение количества цветов. Для большинства изображений, не являющихся фотографическими, количество необходимых для воспроизведения цветов часто меньше 256. Лишние цвета можно убрать из изображения, тем самым уменьшив его размер. Это возможно потому, что формат GIF поддерживает размер палитры меньше 256 цветов. Теоретически мы можем задать любое число цветов палитры от 2 до 256,

например, 45 цветов или 99. Практически же количество цветов в изображении выбирается из ряда: 2, 4, 8, 16, 32, 64, 128, 256. Это связано с тем, что для представления цвета используется всегда целое число бит, а приведенный выше ряд является рядом максимального количества цветов при использовании от 1 до 8 бит на пиксел. В случае, если вы используете в рисунке, допустим, только 97 цветов, для сохранения одного пиксела все равно будет использовано 7 бит, как и при использовании 128 цветов. Поэтому при оптимизации количества цветов нужно ориентироваться на приведенный выше ряд. Это позволит получить более качественную картинку. Уменьшив количество цветов с 256 до 32, можно иногда добиться уменьшения размера файла в 2 раза без заметной потери качества.

Оптимизация палитры изображения. Преобразование полноцветного изображения в индексное, т.е. уменьшение количества цветов до 256 является важной операцией, и здесь следует сказать несколько слов о том, как это делается, современными графическими пакетами.

Палитра конечного изображения может быть либо фиксированной, либо оптимизированной. В первом случае графический редактор просматривает каждую точку изображения и подбирает ближайшую ей по цвету из палитры. Этот способ дает самые худшие результаты с точки зрения правильности воспроизведения цветов. Попробуйте, например, преобразовать фотографию красной розы в индексное изображение, используя палитру, содержащую оттенки зеленого. Это, конечно, крайний случай, но даст вам возможность оценить качество преобразования с использованием фиксированной палитры. Тем не менее, данный способ применяется в основном для того, чтобы изображения приемлемо выглядели на мониторах с малым количеством цветов - 16 или 256. Обычно в этом случае пользуются так называемой безопасной палитрой, состоящей из набора часто используемых цветов и их оттенков. Безопасная палитра гарантирует, что изображения, использующие эту палитру, будут одинаково показаны всеми браузерами.

Во втором случае, когда используется оптимизированная палитра, графический редактор вначале анализирует рисунок и составляет список всех используемых в изображении цветов. Далее, на основании частоты появления цветов, составляется палитра, которая называется оптимизированной. После этого рисунок снова просматривается, и цвет пиксела изменяется на ближайший из палитры. Этот способ дает гораздо лучшие результаты, и именно его следует применять, если у вас нет причин для использования фиксированной палитры.

При оптимизации конкретного изображения задача состоит в том, чтобы выбрать оптимальное количество цветов в палитре. Здесь уместно сказать о способе увеличения количества используемых цветов при их недостатке - так называемом смешивании (dithering). При этом области, которые в оригинале были залиты однородным цветом, после преобразования заполняются смесью пикселей разных цветов, разбросанных по случайному закону. Как правило, каждый отсутствующий в редуцированной палитре цвет передается пикселями двух самых близких к нему цветов новой палитры. В результате изображение приобретает характерную зернистую, шероховатую фактуру. Очень часто смешивание является единственным способом хоть сколько-нибудь адекватно передать исходные цвета с помощью палитры, в которой этих цветов уже нет. Например, у нас есть 16 стандартных чистых цветов, а нам нужен отсутствующий оранжевый цвет. В таком случае можно составить его из красных и желтых точек, разместив их в шахматном порядке.

Издавека будет казаться, что это сплошной оранжевый цвет. Это классический пример смешивания. Его использование при преобразовании изображений может дать очень хорошие результаты. Но с точки зрения оптимизации размеров файла происходит совсем обратное. Смешивание может увеличить размер файла, причем довольно существенно - он может стать больше, чем до оптимизации. А все дело в способе хранения изображения форматом GIF. Изображение перед сохранением на диске подвергается сжатию методом LZW, а основная особенность этого метода заключается в том, что сжатию лучше всего поддаются области, заполненные однородным цветом, и хуже всего - области, состоящие из набора разноцветных точек. Смешивание же как раз основано на том, что получает недостающие цвета путем «перемешивания» точек разных цветов.

Все существующие ныне Web-браузеры, включая Netscape Navigator и Microsoft Internet Explorer, вынуждены при выводе графических файлов на экран изменять их палитру. Во-первых, им приходится приводить к общему знаменателю палитру всех файлов, включенных в одну страницу. Поскольку некоторые компьютерные мониторы не могут отобразить больше 256 цветов одновременно, приходится идти на компромиссы. Во-вторых, в некоторых случаях браузеру приходится также приспособливать палитру файла к фиксированной системной палитре. Чтобы эти неизбежные преобразования не приводили к совсем уж неприемлемым результатам, все современные браузеры пользуются при замещении палитры смешиванием. Это значит, что даже если ваше изображение состоит из идеально гладких цветовых плоскостей, но его палитра почему-либо не может быть отображена напрямую, то в браузере это изображение неизбежно станет шероховатым.

Однако для очень многих изображений, составляющих нередко основу дизайна страницы, смешивание крайне нежелательно. Что же делать? Выход, хоть и не самый лучший, есть. Оказывается, существует набор из 216 цветов, которые с гарантией никогда не подвергаются смешиванию ни в Netscape Navigator, ни в Internet Explorer, ни в Windows, ни в Macintosh. Если ваше изображение будет содержать только цвета из этого набора или, по крайней мере, к этому набору будут относиться цвета, занимающие наибольшую площадь, в браузере оно будет выглядеть намного чище и привлекательнее. Эту палитру часто называют безопасной (browser-safe palette).

Воспользоваться безопасной палитрой не поздно, даже когда графика уже готова. Однако, если вы хотите иметь больший контроль над цветовой стороной своего дизайна, те графические элементы, для которых хотелось бы избежать смешивания, лучше с самого начала рисовать с использованием безопасной палитры. **Безопасная палитра поддерживается последними версиями Adobe Photoshop.**

Стилизация изображения. Для уменьшения количества используемых цветов в изображениях, содержащих фотографические сюжеты, возможно применение стилизации фотографии. Для этого можно воспользоваться такими приемами, как тонирование изображения и применение художественных фильтров, имитирующих рисунки. Фотографии, обработанные таким способом, будут сжиматься лучше и требовать палитру, содержащую менее 256 цветов.

Изменение размеров изображения. Этот метод оптимизации основан на изменении размеров рисунка. Его нельзя применять к обычным рисункам, но он очень хорошо подходит для рисунков, состоящих из элементов типа линий и однородных областей. Допустим, нам понадобился рисунок, представляющий собой

черный квадрат размером 100x100 пикселей. Для уменьшения размера файла достаточно сделать так называемый однопиксельный, т.е. размером 1x1 пиксел, рисунок GIF черного цвета и, как обычно, вставить его в HTML-файл, но размеры картинки указать те, что нужны нам - 100x100. Браузер сам растянёт наш пиксел до нужных размеров, и, таким образом, будет достигнута значительная экономия в объеме файла. Подобным же образом можно использовать одно-пиксельные рисунки других цветов, но наибольшую пользу может принести прозрачное однопиксельное изображение GIF. С его помощью удобно задавать «пустышки» на страницах, абзацные отступы, фиксировать минимальную ширину или высоту таблицы и делать другие полезные вещи.

Фрагментарная оптимизация. Фрагментарная оптимизация применяется в случае, если у вас есть некоторое большое изображение, содержащее сильно отличающиеся по количеству цветов области. В этом случае **изображение можно разрезать на фрагменты и поместить их в таблицу, тогда в браузере оно будет выглядеть как единое целое.** Теперь каждый фрагмент исходного рисунка оптимизируется отдельно. Для фрагментов, содержащих мало цветов, вполне может хватить палитры в 8 или 16 цветов, а для богатых цветами фрагментов можно использовать полную палитру в 256 цветов или сохранять их в формате JPEG. Этот способ позволяет уменьшить объем, занимаемый исходным файлом, примерно раза в 2-3, в зависимости от конкретного изображения. Кроме того, изображение будет появляться на экране более равномерно, так как все фрагменты будут загружаться параллельно. Однако, если количество фрагментов превысит 10, то возможного выигрыша не получится, так как каждый фрагмент должен будет хранить свою копию палитры и служебной информации.

Оптимизация «прозрачных» изображений. Формат GIF позволяет сохранять так называемые «прозрачные» изображения. На самом деле, GIF не поддерживает прозрачность в изображениях - альфа-канал - он лишь позволяет назначить одному любому цвету в палитре атрибут прозрачности. Если браузер рисует на экране такое изображение, то, встречая «прозрачный» пиксел, он просто игнорирует его и не показывает на экране. Само по себе это хорошо, но в случае использования метода сглаживания краев изображения (anti-aliasing) и последующего сохранения в формате GIF, возникает проблема появления ореола вокруг прозрачного изображения в случае, если GIF показывается на фоне, отличном от того, на котором происходило сглаживание и который стал прозрачным. Единственным способом уменьшения проявления этого эффекта является назначение прозрачности цвету, близкому к цвету фона, на котором будет использоваться GIF, и проведение сглаживания на этом же фоне.

Какой формат использовать?

Вопрос о том, в каком формате хранить нужное вам изображение, не так прост, как кажется. Очень часто можно прочесть, **что фотографические изображения, поскольку они имеют более 256 цветов, следует хранить в формате JPEG, а рисунки в формате GIF.** Это правило действительно верно в большинстве случаев. Однако могут быть и исключения. В ряде случаев, **если вам требуется особое качество изображения, следует предпочесть GIF.** Дело в том, что **главной помехой в использовании JPEG является то, что изображения, сохраненные в этом формате, в отличие от GIF имеют визуально заметные искажения.** Они тем заметнее, чем больше величина сжатия и меньше геометрические размеры деталей

изображения. Считается, что если исходный размер файла сжимается до 10 раз то искажения будут не очень заметны. В то же время, если в достаточно «интеллектуальной» программе уменьшить количество цветов фотографии до 256 этих 256 цветов в большинстве случаев хватит на то, чтобы изображение ж проиграло в визуальном восприятии.

Поэтому в том случае, когда для вас существенно, чтобы картинка воспроизводилась с минимальными искажениями, сделайте два файла - в GIF и JPEG-форматах и посмотрите, какой файл при устраивающем вас качестве займет меньше места на диске.

Во всяком случае, имейте в виду следующие обстоятельства:

- Чем больше на рисунке мелких и контрастных деталей, тем сильнее будут заметны искажения, вносимые при сжатии в формате JPEG.
- Чем крупнее геометрические размеры изображения в пикселах и чем ниже контрастность снимка, тем искажения JPEG будут меньше заметны.
- Чем больше на рисунке контрастных и мелких деталей, тем более крупный размер файла получится при сохранении в формате GIF.
- Чем большую площадь занимают однородно окрашенные участки и чем большую горизонтальную протяженность имеют детали рисунка, тем меньший размер займет на диске файл GIF.
- Файлы, записанные в чересстрочном (interlaced) формате GIF и в прогрессивном формате JPEG, то есть так, чтобы изображение «проявлялось» постепенно, будут, при прочих равных условиях, иметь больший размер.

Но есть две области, где GIF - вне конкуренции. Первая - это создание анимированных, то есть движущихся изображений. На самом деле анимированный GIF -это несколько графических изображений, записанных в один файл. Последовательный показ этих изображений создает иллюзию движения. В файле GIF хранится информация о том, сколько раз в секунду надо менять изображение, в какой последовательности показывать картинки и т.д. Для создания анимированного изображения вы в любом графическом редакторе рисуете картинки, из которых будет строиться анимация, а затем загружаете их в специальную программу подготовки анимации, где указываете, как эти картинки должны чередоваться. Таких программ существует большое количество.

Другое незаменимое свойство GIF - это возможность получения прозрачного фона изображения, то есть такого фона, сквозь который видно все, что под ним находится. Создавать GIF с прозрачным фоном может любой профессиональный графический редактор, например, Adobe Photoshop.

3. Вопросы для закрепления.

- 1) Какие виды графических форматов Вы знаете?
- 2) Какой объем графического файла допустим на web – странице? ».
- 3) Назовите способы оптимизации графических файлов
- 4) Чем отличается формат Gif от Jpg?

4. Домашнее задание: Подготовить графические файлы для сайта и оптимизировать их.

Подготовить рефераты на тему: «Форматы, применяемые для представления графики в сети Интернет», «Методы интеграции изображений в документ», «Приемы позволяющие осуществить фрагментацию изображений»,

«Принципы создания GIF-анимации», «Программы для оптимизации изображений».

Лекция 5

Тема: Структура HTML документа

Цели: Ознакомить учащихся с основными понятиями HTML; дать понятие основных элементов HTML; определить структуру HTML документа.

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Язык HTML
 - 2) Структура HTML документа
 - 3) Спецификации элементов
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

ОСНОВНЫЕ ПОНЯТИЯ HTML

HTML – язык гипертекстовой разметки документов (HyperText Markup Language). С помощью HTML создаются Web-страницы, которые находятся в глобальной компьютерной сети Интернет. HTML – это не язык программирования в традиционном смысле, он является языком разметки. С помощью HTML текстовый документ разбивают на блоки смысловой информации (заголовки, параграфы, таблицы, рисунки и т.п.).

Гипертекстовый документ – это документ, содержащий переходы (**гиперссылки**) на другие документы. При использовании гиперссылки происходит перемещение от одного документа к другому (как по цепочке) в Интернете. HTML-документ является гипертекстовым документом.

Особенности HTML-документа:

1. HTML-документ может содержать текст, графику, видео и звук.
2. В общем случае HTML-документ – это один или несколько текстовых файлов, имеющих расширение .htm или .html.
3. Создавать HTML-документ можно как с помощью специальных программ – редакторов HTML (например, FrontPage), так и с помощью любого текстового редактора (например, блокнота Windows).
4. Для просмотра HTML-документов существуют специальные программы-**браузеры**. Они **интерпретируют** HTML-документы, т.е. переводят текст документа в Web-страницу, и отображают ее на экране пользователя. Существует очень много различных браузеров, но наиболее распространенными браузерами являются Microsoft Internet Explorer, Netscape Navigator и Opera.
5. Если при интерпретации HTML-документа браузер чего-то не понимает, то сообщения об ошибке не возникает, а это место в HTML-документе игнорируется и не отображается браузером.

HTML-документ состоит из элементов HTML.

Элемент HTML – это чаще всего два тега (открывающий и закрывающий) и часть документа между ними. Кроме того, существуют элементы HTML, состоящие только из одного тега.

Тег – в переводе с английского – ярлык, этикетка. Тег определяет тип выводимого элемента HTML (например, заголовок, таблица, рисунок и т.п.). Сам тег не отображается браузером. Тег представляет собой последовательность элементов:

- символ левой угловой скобки (<) – начало тега;
- необязательный символ слеша (/) – символ используется, чтобы обозначить закрывающий тег;
- имя тега;
- необязательные атрибуты в открывающем теге;
- символ правой угловой скобки (>)

Атрибуты – необязательный набор параметров, определяющих дополнительные свойства элемента HTML (например, цвет или размер). Атрибут состоит:

- из имени атрибута;
- знака равенства (=);
- значения атрибута – строки символов, заключенной в кавычки

Пример элемента HTML:

```
<H1 ALIGN= "CENTER">Глава 1</H1>
```

В этом примере:

<H1 ALIGN= "CENTER"> – открывающий тег

</H1> – закрывающий тег

H1 – имя тега

ALIGN= "CENTER" – атрибут

ALIGN – имя атрибута

"CENTER" – значение атрибута

Правила создания HTML-документов:

1. Теги и атрибуты можно записывать в любом регистре, т.е. </H1> и </h1> – это одно и то же.

2. Несколько пробелов подряд, символы табуляции и перевода строки при интерпретации браузером заменяются на один пробел. Это позволяет писать хорошо структурированные исходные тексты файлов HTML.

3. Рекомендуется давать имена файлам HTML строчными английскими буквами. Длина имени – до восьми символов. В принципе, можно не придерживаться данной рекомендации, но тогда пользователи, работающие в операционных системах, отличных от Windows, не смогут воспользоваться вашими HTML-документами.

СТРУКТУРА HTML-ДОКУМЕНТА

Каждый HTML-документ должен начинаться тегом <HTML> и заканчиваться тегом </HTML>. Эти теги обозначают, что находящиеся между ними строки представляют единый HTML-документ. Кроме того, можно заметить, что файл HTML в целом является элементом языка HTML.

Также в HTML-документе должны присутствовать элементы HEAD (информация о документе) и BODY (тело документа).

Раздел документа HEAD

Раздел документа HEAD определяет его заголовок, а также содержит дополнительную информацию о документе для браузера. Этот раздел не является

обязательным, однако рекомендуется всегда использовать его в своих HTML-документах, так как правильно составленный заголовок может быть весьма полезен.

Раздел заголовка начинается тегом **<HEAD>** и следует сразу за тегом **<HTML>**. Между открывающим и закрывающим тегами элемента **HEAD** располагаются другие элементы заголовка.

Название документа TITLE

Для того чтобы дать название HTML-документу, предназначен тег **<TITLE>**. Это название будет выведено в заголовок окна браузера. Название записывается между тегами **<TITLE>** и **</TITLE>** и представляет собой строку текста. Длина этой строки может быть любой, но рекомендуется делать ее не больше 60 символов. Элемент **TITLE** должен находиться только в разделе **HEAD**.

Раздел документа BODY

В этом разделе документа располагается та информация, которая отображается в окне браузера. Раздел **BODY** должен начинаться тегом **<BODY>** и завершаться тегом **</BODY>**, между которыми располагаются элементы HTML, из которых и состоит содержание документа.

Спецификация элемента BODY

Тег **<BODY>** имеет ряд атрибутов, определяющих внешний вид документа. Ниже приводится спецификация тега **<BODY>**.

```
<BODY  
TEXT="цвет текста"  
BGCOLOR="цвет фона"  
BACKGROUND="адрес фонового рисунка"  
LINK="цвет непосещенной гиперссылки"  
VLINK="цвет посещенной гиперссылки"  
ALINK="цвет активной гиперссылки"  
>
```

Атрибут **TEXT** задает цвет шрифта для всего документа в формате **RGB** или в символьной нотации. По умолчанию (если не указан этот атрибут) используются настройки браузера.

Атрибут **BGCOLOR** задает цвет фона окна браузера документа в формате **RGB** или в символьной нотации. По умолчанию используются настройки браузера.

Атрибут **BACKGROUND** позволяет указать адрес и имя рисунка, используемого в качестве фона. Этот рисунок будет размножен и распределен на заднем плане документа.

Атрибуты **LINK**, **VLINK** и **ALINK** задают цвета гиперссылок в формате **RGB** или в символьной нотации. По умолчанию используются настройки браузера. Непосещенная гиперссылка – гиперссылка, которая еще не использовалась для перехода к другому документу. Посещенная гиперссылка – гиперссылка, которая уже использовалась для перехода к другому документу. Активная гиперссылка – гиперссылка на документ, к которому в данный момент происходит переход.

Советы по использованию атрибутов тега BODY

□ Если вы указываете хотя бы один цвет в теге **BODY**, то укажите и остальные. Это связано с тем, что пользователь может установить настройки цветов своего браузера как ему удобней. Указание только одного цвета может привести к ситуации,

что у некоторых пользователей текст сольется с цветом фона. В результате просмотр документа будет затруднен.

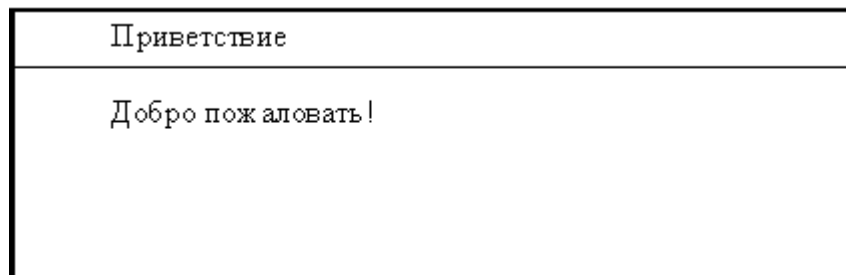
□ Выбирайте цвет текста так, чтобы он "работал" вместе с цветом фона или основными цветами изображения. Например, красное на зеленом может вызвать серьезные проблемы у значительного числа людей.

□ В элементе BODY можно задать как BGCOLOR, так и BACKGROUND. В этом случае браузер отдает предпочтение BACKGROUND, но если изображение фона невозможно загрузить, будет использовано BGCOLOR. Поэтому старайтесь задавать цвет фона похожим на цвет фонового рисунка, чтобы не нарушился цветовой баланс документа.

Примеры

Пример простого HTML-документа

```
<HTML>
<HEAD>
  <TITLE>Приветствие</TITLE>
</HEAD>
<BODY>
  <P>Добро пожаловать!</P>
</BODY>
</HTML>
```

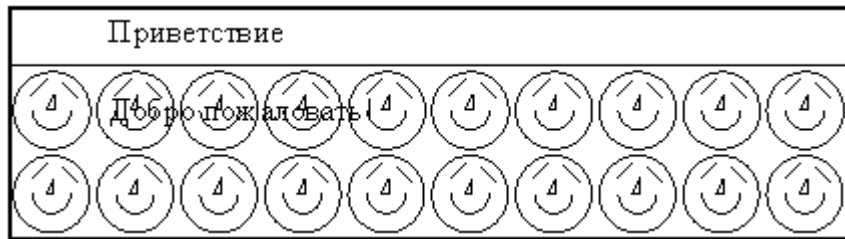


Этот документ отобразится в браузере так:

В этом примере обратите внимание на то, куда выводится браузером название документа в элементе TITLE.

Пример использования фонового рисунка

```
<HTML>
<HEAD>
  <TITLE>Приветствие</TITLE>
</HEAD>
<BODY BACKGROUND="smail.gif">
  <P>Добро пожаловать!</P>
</BODY>
</HTML>
```



Этот документ отобразится в браузере так:

3. Вопросы для закрепления.

- 1) Дайте понятия элемента HTML, тега, атрибутов.
- 2) Что такое браузер?
- 3) Каковы особенности и правила HTML-документа?
- 4) Какие элементы HTML входят в обязательную структуру HTML-документа?
- 5) Для чего используется элемент BODY, и какие он имеет атрибуты?

4. Домашнее задание: Создать две web-страницы в редакторе блокнот, используя язык HTML, с информацией о себе.

Лекция 6

Тема: Цветовые спецификации. Графические элементы

Цели: *Ознакомить учащихся с основными цветовыми спецификациями ;использование цветовых обозначений с помощью символьной нотации.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Цветовые спецификации
 - 2) Символьная нотация
 - 3) Добавление графических объектов с помощью языка HTML
3. Закрепление изученного.
4. Итог урока.

Ход урока.

12.Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

13.Изложение нового материала.

ЦВЕТОВЫЕ СПЕЦИФИКАЦИИ

Для определения цвета различных элементов HTML-документа необходимо указать значение соответствующих атрибутов. Указывать значения этих атрибутов можно двумя способами:

- определять цвет в формате RGB;
- определять цвет, используя символьную нотацию

Формат RGB

Формат RGB – это система указания цвета, которая базируется на смешении трех основных цветов: красном (RED), зеленом (GREEN) и синем (BLUE). Итоговый цвет определяется цифрами в шестнадцатеричном коде. Для каждого цвета задается шестнадцатеричное значение в пределах от 0 до FF, что соответствует диапазону 0-255 в десятичном исчислении. Затем эти значения объединяются в одно число, перед которым ставится символ #. Например, число #800080 обозначает фиолетовый цвет. Указывая цвет в формате RGB, можно определить более шестнадцати миллионов цветовых оттенков.

СИМВОЛЬНАЯ НОТАЦИЯ

Задание цвета в формате RGB имеет один недостаток – необходимо помнить совокупности цифр для указания цвета. Этому недостатка лишена символьная нотация. Можно указывать название цвета на английском языке. Но у этого способа указания цвета тоже есть недостаток – определено всего шестнадцать имен цветов.

СООТВЕТСТВИЕ ФОРМАТА RGB И СИМВОЛЬНОЙ НОТАЦИИ

Ниже приведена таблица соответствий указания цвета в символьной нотации и формате RGB.

Символьная нотация	Формат RGB	Цвет
Black	#000000	Черный

Silver	#C0C0C0	Серебро
Gray	#808080	Серый
White	#FFFFFF	Белый
Maroon	#800000	Темно-бордовый
Red	#FF0000	Красный
Purple	#800080	Фиолетовый
Fuchsia	#FF00FF	Розовый
Green	#008000	Зеленый
Lime	#00FF00	Известь
Olive	#808000	Оливковый
Yellow	#FFFF00	Лимонный
Navy	#000080	Темно-синий
Blue	#0000FF	Синий
Teal	#008080	Темно-бирюзовый
Aqua	#00FFFF	Бирюзовый

Таким образом, строка `ТЕХТ="#008000"` и строка `ТЕХТ="Green"` в теге `<BODY>` одинаково определяют цвет шрифта – зеленый.

. ГРАФИЧЕСКИЕ ЭЛЕМЕНТЫ

Одним из достоинств HTML-документа является возможность использования графических элементов в оформлении. Можно выделить три элемента, чаще всего используемых в HTML-документах: горизонтальные линии, таблицы и рисунки.

ГОРИЗОНТАЛЬНЫЕ ЛИНИИ

Горизонтальные линии визуально подчеркивают законченность той или иной области документа. Сейчас часто используют рельефную, вдавленную линию, чтобы обозначить “объемность” документа.

Тег `<HR>` позволяет провести рельефную горизонтальную линию в окне большинства браузеров. Этот тег не является контейнером, поэтому не требует закрывающего тега. До и после линии автоматически вставляется пустая строка. Спецификация тега `<HR>`:

```
<HR
ALIGN="выравнивание"
WIDTH="длина линии"
SIZE="толщина линии"
NOSHADE
>
```

Атрибут `ALIGN` определяет способ выравнивания линии. Он может иметь следующие значения:

- `LEFT` – линия выравнивается по левому краю окна браузера. Это значение используется по умолчанию.
- `CENTER` – линия выравнивается по центру окна браузера.
- `RIGHT` – линия выравнивается по правому краю окна браузера.

Атрибут `WIDTH` задает длину линии. Значением данного атрибута является число. Это число означает длину линии в пикселях. Если после числа стоит знак `%`, то это означает длину в процентах от ширины окна. Например:

`<HR WIDTH="400">` – линия длиной 400 пикселей.

`<HR WIDTH="50%">` – линия длиной 50 процентов от ширины окна.

Атрибут SIZE задает толщину линии. Значением этого атрибута является число. Это число означает толщину линии в пикселях.

Атрибут NOSHADE отменяет “трехмерность” линии.

РИСУНКИ

Без иллюстраций документ скучен, вял и однообразен. HTML позволяет использовать рисунки в формате JPG и GIF для оформления HTML-документов. Для вставки рисунков используется тег ****. Спецификация тега ****:

```
<IMG  
SRC="адрес рисунка"  
ALIGN="выравнивание"  
HEIGHT="высота рисунка"  
WIDTH="ширина рисунка"  
BORDER="толщина рамки"  
>
```

Атрибут SRC определяет URL-адрес рисунка, который будет отображаться броузером.

Атрибут ALIGN определяет способ выравнивания рисунка. Он может иметь следующие значения:

- TOP – рисунок выравнивается по верхнему краю текущей строки.
- MIDDLE – рисунок выравнивается серединой по текущей строке.
- BOTTOM – рисунок выравнивается по нижнему краю текущей строки.
- LEFT – рисунок прижимается к левому краю окна броузера и обтекается текстом.
- RIGHT – рисунок прижимается к правому краю окна броузера и обтекается текстом.

Атрибут HEIGHT определяет высоту рисунка в пикселях.

Атрибут WIDTH определяет ширину рисунка в пикселях.

Используя атрибуты HEIGHT и WIDTH можно увеличивать или уменьшать рисунок. Если указать только один из этих атрибутов, то рисунок будет увеличен или уменьшен пропорционально и по ширине, и по высоте.

Атрибут BORDER позволяет задавать рамку вокруг рисунка. Значение этого атрибута – толщина рамки в пикселях. По умолчанию – 1.

Пример выравнивания рисунков:

```
<HTML>  
<HEAD>  
<TITLE>Пример выравнивания</TITLE>  
</HEAD>  
<BODY>  
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="TOP">по  
верхнему краю</P>  
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="BOTTOM">  
по нижнему краю</P>  
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="MIDDLE">  
по середине</P>  
</BODY>  
</HTML>
```

Броузер отобразит данный пример так:

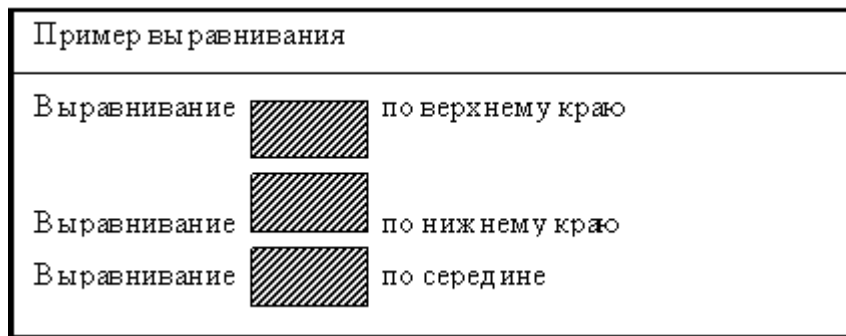


Рисунок-ссылка

Можно использовать рисунки в качестве гиперссылок. Для этого нужно включить тег `IMG` в содержание элемента `A`. Например:

```
<A HREF="glava5.htm"><IMG SRC="ris1.jpg"></A>
```

3. Вопросы для закрепления.

- 1) Чем отличается символьная нотация от формата RGB?
- 2) Какие значения в символьной нотации можно использовать для указания цвета?

4. Домашнее задание: Подготовить доклад на тему: «Дополнительные web-технологии» (Java, CGI, SSI, CSS, PHP, ASP, VBScript и др.)»

Лекция 7

Тема: Ввод текстовой информации

Цели: *Ознакомить учащихся с основными элементами форматирования текста: заголовков, абзац, перевод строки, списки, форматирование шрифта; способ создания таблиц с помощью языка HTML.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) форматирование текста
 - 2) форматирование шрифта
 - 3) создание таблиц
3. Закрепление изученного.
4. Итог урока.

Ход урока.

14.Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

15.Изложение нового материала.

Любые тексты, будь то школьные сочинения, заметка в газету или техническое описание устройства, имеют определенную структуру. Элементами такой структуры являются заголовки, подзаголовки, абзацы, списки и др.

Разбиение всего текста на структурные элементы называется логическим форматированием. В HTML-документе логическое форматирование достигается с помощью специальных тегов.

АБЗАЦЫ

Одним из первых правил составления любых документов является разбиение его текста на отдельные абзацы, выражающие законченную мысль. В HTML-документе разделение на абзацы производится с помощью специального тега **<P>**. Синтаксис этого тега таков:

```
<P  
ALIGN="выравнивание">
```

Атрибут ALIGN определяет способ выравнивания абзаца. Он может иметь следующие значения:

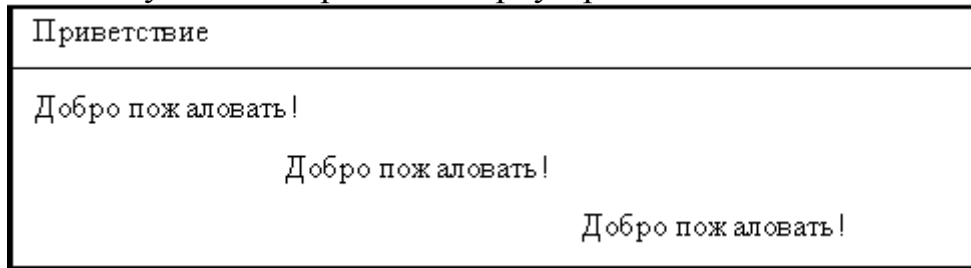
- LEFT – текст выравнивается по левому краю окна броузера. Это значение используется по умолчанию, т.е. когда атрибут не указан.
- CENTER – текст выравнивается по центру окна броузера.
- RIGHT – текст выравнивается по правому краю окна броузера.

Пример использования тега **<P>**:

```
<HTML>  
<HEAD>  
<TITLE>Приветствие</TITLE>  
</HEAD>  
<BODY>  
<P>Добро пожаловать!</P>  
<P ALIGN="CENTER">Добро пожаловать!</P>
```

```
<P ALIGN="RIGHT">Добро пожаловать!</P>
</BODY>
</HTML>
```

Этот документ отобразится в браузере так:



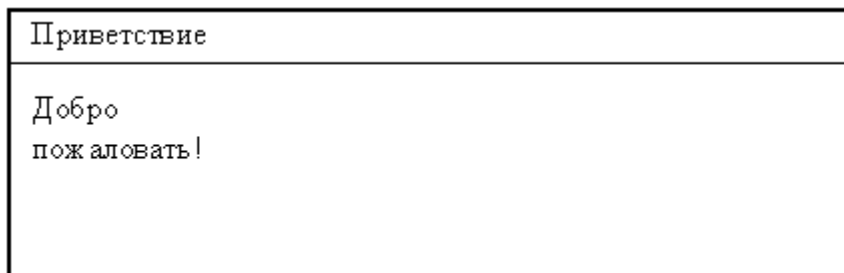
Браузер автоматически формирует абзацы в зависимости от ширины окна браузера или размера шрифта, перенося слова из строки в строку и отделяя абзацы друг от друга пустой строкой.

УПРАВЛЕНИЕ ПЕРЕВОДОМ СТРОКИ

Так как браузер автоматически определяет места переноса строк, иногда возникают ситуации запретить перевод строки в каком-нибудь месте или, наоборот, принудительно сделать перевод строки в каком-то определенном месте. Для этого существуют особые теги, управляющие переводом строк.

Когда необходимо сделать принудительный перевод строки, используют тег **
**. Этот тег не имеет атрибутов и закрывающего тега. Пример использования принудительного перевода строки:

```
<HTML>
<HEAD>
<TITLE>Приветствие</TITLE>
</HEAD>
<BODY>
<P>Добро<BR>позаловать!</P>
</BODY>
</HTML>
```



Этот пример будет выглядеть так:

При использовании тега **
** пустая строка не образуется, т.е. абзац не прерывается.

В некоторых случаях, наоборот, бывает необходимо сделать так, чтобы браузер не производил перевода строки. Например, не рекомендуется отрывать буквы инициалов от фамилии. В таких случаях тот участок текста, в котором нельзя переводить строку, следует поместить в элемент **NOBR**.

Пример:

```
<P>Это стихотворение написал <NOBR>А.С.
Пушкин</NOBR> – великий русский поэт.</P>
```

В браузере участок текста “А.С. Пушкин” всегда будет отображаться на одной строке.

Если получится так, что строка, расположенная в элементе NOBR, будет выходить за пределы окна браузера, то внизу окна появится горизонтальная полоса прокрутки.

Заголовки

Почти в каждом тексте используются заголовки для отдельных частей документа. Эти заголовки представляют собой фрагменты текста, которые выделяются на экране при отображении страницы браузером.

Для разметки заголовков используются теги <H1>, <H2>, <H3>, <H4>, <H5> и <H6>. Эти теги требуют соответствующего закрывающего тега. Заголовок с номером 1 является самым крупным (заголовок верхнего уровня), а с номером 6 – самым мелким. Теги заголовка нельзя использовать для выделения отдельных слов текста с целью увеличения их размера. При использовании тегов заголовков происходит вставка пустой строки до и после заголовка, поэтому тегов абзаца и перевода строки здесь не требуется.

Синтаксис тегов заголовков:

<Hn

ALIGN="выравнивание">

Атрибут ALIGN определяет способ выравнивания заголовка. Он может иметь те же значения, что и аналогичный атрибут у тега абзаца.

Пример использования разных заголовков:

<HTML>

<HEAD>

<TITLE>Пример</TITLE>

</HEAD>

<BODY>

<H1>Заголовок 1</H1>

<H2 ALIGN="CENTER">Заголовок 2</H2>

<P>Простой текст</P>

</BODY>

</HTML>

Пример
Заголовок 1
Заголовок 2
Простой текст

Вот так браузер отобразит данный пример:

СПИСКИ

В языке HTML предусмотрен специальный набор тегов для представления информации в виде списков. Списки являются одним из наиболее часто употребляемых форм представления данных как в электронных документах, так и печатных. В языке HTML предусмотрены маркированные, нумерованные списки и списки определений.

Маркированный список

Этот список еще называется нумерованным или неупорядоченным. В маркированном списке для выделения его элементов используются специальные символы, называемые маркерами списка. Вид маркеров определяется браузером, причем при создании вложенных списков браузеры автоматически разнообразят вид маркеров различного уровня вложенности.

Для создания маркированного списка необходимо использовать тег-контейнер `` ``, внутри которого располагаются все элементы списка. Открывающий и закрывающий теги списка обеспечивают перевод строки до и после списка, отделяя, таким образом, список от основного содержимого документа, поэтому нет необходимости использовать теги абзаца или принудительного перевода строки.

Каждый элемент списка должен начинаться тегом `` и заканчиваться тегом ``.

Спецификация элемента `UL`:

```
<UL
```

```
TYPE="вид маркера">
```

Атрибут `TYPE` задает вид маркера, которым выделяются элементы списка. Он может имеет следующие значения:

- `DISK` – маркеры отображаются закрашенными кружочками, это значение используется по умолчанию;

- `CIRCLE` – маркеры отображаются не закрашенными кружочками;

- `SQUARE` – маркеры отображаются квадратиками.

Спецификация элемента `LI` для маркированного списка:

```
<LI
```

```
TYPE="вид маркера">
```

Атрибут `TYPE` задает вид маркера, он может принимать такие же значения, что и одноименный атрибут элемента `UL`. Значение по умолчанию – `DISK`.

Пример использования маркированного списка:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Пример списка</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<UL>
```

```
Крупные города России:
```

```
<LI>Москва</LI>
```

```
<LI>Санкт-Петербург</LI>
```

```
<LI>Новосибирск</LI>
```

```
</UL>
```

```
</BODY>
```

```
</HTML>
```

Вот так браузер отобразит данный пример:

Пример списка
<p>Крупные города России:</p> <ul style="list-style-type: none"> • Москва • Санкт-Петербург • Новосибирск

Нумерованный список

Нумерованные списки иногда называют упорядоченными. Списки данного типа представляют собой упорядоченную последовательность отдельных элементов. Отличием от маркированных списков является то, что в нумерованном списке перед каждым его элементом автоматически проставляется порядковый номер. Вид нумерации зависит от браузера и может задаваться атрибутами тегов списка. В остальном реализация нумерованного списка во многом похожа на реализацию маркированного списка.

Для создания нумерованного списка следует использовать тег-контейнер `` ``, внутри которого располагаются все элементы списка.

Каждый элемент списка должен начинаться тегом `` и заканчиваться тегом ``.

Спецификация элемента OL:

`<OL`

`TYPE="вид нумерации"`

`START="начальная позиция">`

Атрибут `TYPE` задает вид нумерации, которой выделяются элементы списка. Он может иметь следующие значения:

A – маркеры в виде прописных латинских букв;

a – маркеры в виде строчных латинских букв;

I- маркеры в виде больших римских цифр;

i- маркеры в виде маленьких римских цифр;

1- маркеры в виде арабских цифр, это значение используется по умолчанию.

Атрибут `START` определяет позицию, с которой начинается нумерация списка. Используя этот атрибут, можно начать нумерацию, например, с цифры 5 или буквы E, в зависимости от вида нумерации. Значением атрибута `START` является число, вне зависимости от вида нумерации.

Спецификация элемента LI для нумерованного списка:

`<LI`

`TYPE="вид нумерации"`

`VALUE="номер элемента">`

Атрибут `TYPE` задает вид нумерации, он может принимать такие же значения, что и одноименный атрибут элемента OL. По умолчанию значение этого атрибута – 1.

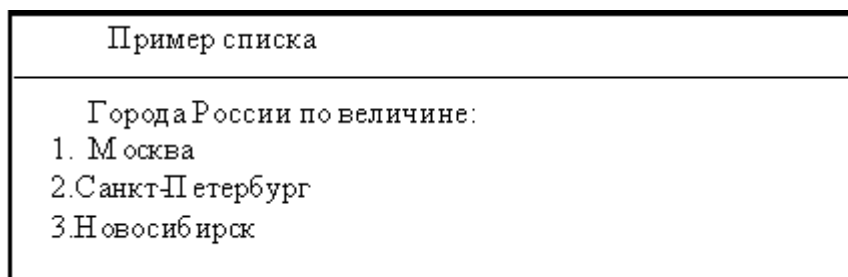
Атрибут `VALUE` позволяет изменить номер данного элемента, при этом изменятся номера и последующих элементов.

Пример использования нумерованного списка:

`<HTML>`

```
<HEAD>
<TITLE>Пример списка</TITLE>
</HEAD>
<BODY>
<UL>
Города России по величине:
<LI>Москва</LI>
<LI>Санкт-Петербург</LI>
<LI>Новосибирск</LI>
</UL>
</BODY>
</HTML>
```

Вот так браузер отобразит данный пример:



Список определений

Списки определений, также называемые словарями определений специальных терминов, являются особым видом списков. В отличие от других типов списков каждый элемент списка определений всегда состоит из двух частей. В первой части элемента списка указывается определяемый термин, а во второй части – текст в форме словарной статьи, раскрывающий значение термина.

Списки определений задаются с помощью тега-контейнера **<DL>**. Внутри него тегом **<DT>** отмечается определяемый термин, а тегом **<DD>** – абзац с его определением. Внутри элемента **<DT>** нельзя использовать абзацы (P) и заголовки (H1-H6), но их можно использовать внутри элемента **<DD>**. Атрибутов для элементов списка определений нет. В общем случае список определений записывается следующим образом:

```
<DL>
<DT>Термин</DT>
<DD>Определение термина</DD>
</DL>
```

Пример использования списка определений:

```
<HTML>
<HEAD>
<TITLE>Пример списка определений</TITLE>
</HEAD>
<BODY>
<DL>
<H2 ALIGN="CENTER">Состав Microsoft Office</H2>
<DT>Microsoft Word</DT>
```



```

<DD>Многофункциональный текстовый процессор</DD>
<DT>Microsoft Excel</DT>
<DD>Программа для работы с электронными
таблицами</DD>
<DT>Microsoft Access</DT>
<DD>Система управления базами данных</DD>
</DL>
</BODY>
</HTML>

```

Этот пример отобразится в браузере так:

Пример списка определений	
Состав Microsoft Office	
Microsoft Word	Многофункциональный текстовый процессор
Microsoft Excel	Программа для работы с электронными таблицами
Microsoft Access	Система управления базами данных

ФОРМАТИРОВАНИЕ ТЕКСТА

В языке HTML предусмотрены специальные теги, предназначенные для форматирования текста. Они позволяют изменять вид шрифта, цвет, размер и др.

Чтобы отобразить текст полужирным шрифтом, используют тег ****. Например:

<P>Это полужирный шрифт</P>

Тег **<I>** отображает текст курсивом. Например:

<P>Выделение <I>курсивом</I></P>

Используя тег **<TT>**, можно отобразить текст шрифтом, в котором все буквы имеют одинаковую ширину. Это так называемый моноширинный шрифт. Например:

<P>Это <TT>моноширинный</TT> шрифт</P>

Для отображения текста подчеркнутым используется тег **<U>**. Например:

<P>Пример <U>подчеркивания</U> текста</P>

Тег **<S>** отображает текст, перечеркнутый горизонтальной линией. Например:

<P>Пример <S>зачеркивания</S> текста</P>

Тег **<BIG>** выводит текст шрифтом большего (чем непомеченная часть текста) размера. Например:

<P>Шрифт <BIG>большого</BIG> размера</P>

Тег **<SMALL>** выводит текст шрифтом меньшего размера. Например:

<P>Шрифт <SMALL>меньшего</SMALL> размера</P>

Тег **<SUB>** сдвигает текст ниже уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для математических индексов. Например:

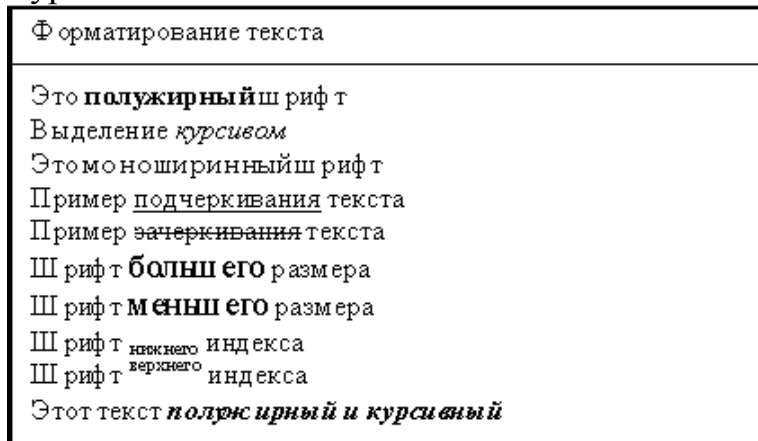
<P>Шрифт _{нижнего} индекса</P>

Тег **<SUP>** сдвигает текст выше уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для задания степеней чисел в математике. Например:

<P>Шрифт ^{верхнего} индекса</P>

Теги форматирования могут быть вложенными друг в друга. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере. Например:

<P>Этот текст <I>полужирный и курсивный</I></P>



Вот так браузер отобразит приведенные выше примеры:

Тег

Тег позволяет изменить шрифт для блока текста. Этот тег имеет следующую спецификацию:

```
<FONT  
FACE="тип шрифта"  
COLOR="цвет шрифта"  
SIZE="размер шрифта"  
>
```

Атрибут FACE служит для указания типа шрифта, которым браузер будет выводить текст (если такой шрифт имеется на компьютере). Значением данного атрибута служит название шрифта, которое должно в точности совпадать с названием шрифта, имеющимся у пользователя. Если такой шрифт не найдется, то данное указание проигнорируется и будет использован шрифт, установленный по умолчанию.

Можно установить как один, так и несколько названий шрифтов, разделяя их запятыми. Тогда список шрифтов будет просматриваться слева направо так: если на компьютере пользователя нет шрифта, указанного в списке первым, то делается попытка найти следующий и т.д. Если ни одного шрифта найти не удалось, то будет использоваться шрифт, установленный браузером по умолчанию.

Атрибут COLOR устанавливает цвет шрифта. Значение этого атрибута может быть указано в формате RGB или символьной нотацией.

Атрибут SIZE служит для указания размера шрифта. Указывать размер шрифта можно двумя способами: абсолютной величиной или относительной величиной. При указании размера абсолютной величиной значением атрибута является число от 1 до 7. 1 – самый маленький шрифт, 7 – самый большой. При указании размера относительной величиной значением атрибута является число со знаком + или -. В данном случае шрифт будет увеличен (+) или уменьшен (-) от размера, принятого по умолчанию.

Пример использования тега :

```
<HTML>  
<HEAD>  
<TITLE>Пример изменения шрифта</TITLE>
```

```
</HEAD>
<BODY>
<P>Шрифт по умолчанию<BR>
<FONT COLOR="GREEN">зеленый шрифт</FONT><BR>
<FONT FACE="Arial">другая форма шрифта</FONT><BR>
<FONT SIZE="6">размер шрифта – 6</FONT><BR>
<FONT SIZE="+1">размер увеличен на 1</FONT></P>
</BODY>
</HTML>
```

Этот пример отобразится так:

Пример изменения шрифта
Шрифт по умолчанию зеленый шрифт другая форма шрифта размер шрифта – 6 размер увеличен на 1

Контейнер DIV

Иногда бывает необходимо произвести выравнивание большого блока документа, содержащего не только текст, но и рисунки, таблицы и т.п. Для этих целей используется элемент-контейнер **DIV**. Спецификация элемента **DIV**:

```
<DIV
ALIGN="выравнивание">
```

Атрибут **ALIGN** определяет тип выравнивания содержимого и может иметь те же значения, что и элемент **P**.

Отступы

Иногда требуется отобразить блок текста с отступом. Для этого блок текста помещают в элемент-контейнер **BLOCKQUOTE**. Тогда содержимое этого элемента будет отображаться с небольшими отступами слева и справа, а также отделяться от остального текста пустыми строками.

ТАБЛИЦЫ

Одним из наиболее мощных и широко применяемых в **HTML** средств являются таблицы. Они используются не только традиционно как метод представления данных, но и как средство форматирования **Web**-страниц. Документ **HTML** может содержать произвольное число таблиц, причем допускается вложенность таблиц друг в друга.

Каждая таблица начинается тегом **<TABLE>** и заканчивается тегом **</TABLE>**. Внутри этой пары тегов располагается описание содержимого таблицы. Любая таблица состоит из одной или нескольких строк, в которых задаются данные для отдельных ячеек.

Каждая строка начинается тегом **<TR>** и заканчивается тегом **</TR>**. Отдельная ячейка в строке обрамляется парой тегов **<TD>** и **</TD>** или **<TH>** и **</TH>**. Тег **<TH>** используется для ячеек заголовка таблицы, а **<TD>** – для ячеек данных. Отличие

этих тегов в том, что в заголовке по умолчанию используется полужирный шрифт, а для данных – обычный.

Теги <TD> и <TH> не могут появляться вне описания строки таблицы <TR>.

Пример таблицы:

```
<HTML>
<HEAD>
  <TITLE>Пример таблицы</TITLE>
</HEAD>
<BODY>
<TABLE>
  <TR>
    <TD>Ячейка 1</TD>
    <TD>Ячейка 2</TD>
  </TR>
  <TR>
    <TD>Ячейка 3</TD>
    <TD>Ячейка 4</TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

Этот пример отобразится в браузере так:

Пример таблицы	
Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Спецификация тега <TABLE>:

```
<TABLE
ALIGN="выравнивание"
BORDER="толщина рамки"
CELLPADDING="расстояние"
CELLSPACING="расстояние"
HEIGHT="высота"
VALIGN="вертикальное выравнивание"
WIDTH="ширина"
>
```

Атрибут ALIGN определяет выравнивание таблицы в окне просмотра браузера. Он может иметь одно из двух значений: LEFT (по левому краю) и RIGHT (по правому краю). По умолчанию – LEFT.

Атрибут BORDER управляет толщиной рамки. Значением этого атрибута является число. Это число определяет толщину рамки таблицы в пикселях. По умолчанию толщина рамки – 1.

Атрибут CELLPADDING определяет расстояние в пикселях между рамкой и содержимым ячейки. По умолчанию – 1.

Атрибут **CELLSPACING** определяет расстояние в пикселях между ячейками таблицы. По умолчанию – 2.

Атрибут **HEIGHT** определяет высоту таблицы в пикселях.

Атрибут **VALIGN** определяет вертикальное выравнивание содержимого таблицы. Он может иметь следующие значения: **TOP** (по верхнему краю), **MIDDLE** (посередине) и **BOTTOM** (по нижнему краю). По умолчанию – **MIDDLE**.

Атрибут **WIDTH** определяет ширину таблицы в пикселях или процентах от ширины окна браузера.

Спецификация тега **<TR>**

<TR

ALIGN="выравнивание"

BGCOLOR="цвет фона"

VALIGN="вертикальное выравнивание"

>

Атрибут **ALIGN** определяет выравнивание содержимого всех ячеек строки. Он может иметь одно из трех значений: **LEFT** (по левому краю), **RIGHT** (по правому краю) и **CENTER** (по центру). По умолчанию – **CENTER**.

Атрибут **BGCOLOR** определяет цвет фона для всех ячеек строки. Его значение можно указывать в символьной нотации или в формате **RGB**.

Атрибут **VALIGN** определяет вертикальное выравнивание содержимого всех ячеек строки. Он может иметь следующие значения: **TOP** (по верхнему краю), **MIDDLE** (посередине) и **BOTTOM** (по нижнему краю). По умолчанию – **MIDDLE**.

Спецификация тега **<TD>**

<TD

ALIGN="выравнивание"

BGCOLOR="цвет фона"

COLSPAN="количество ячеек"

HEIGHT="высота ячейки"

ROWSPAN=" количество ячеек "

VALIGN="вертикальное выравнивание"

WIDTH="ширина ячейки"

>

Атрибут **ALIGN** определяет выравнивание содержимого ячейки. Он может иметь одно из трех значений: **LEFT** (по левому краю), **RIGHT** (по правому краю) и **CENTER** (по центру). По умолчанию – **CENTER**.

Атрибут **BGCOLOR** определяет цвет фона для ячейки. Его значение можно указывать в символьной нотации или в формате **RGB**.

Атрибут **COLSPAN** позволяет объединить несколько соседних ячеек по горизонтали. Значение этого атрибута – количество объединяемых ячеек.

Атрибут **HEIGHT** определяет высоту ячейки в пикселях.

Атрибут **ROWSPAN** позволяет объединить несколько соседних ячеек по вертикали. Значение этого атрибута – количество объединяемых ячеек.

Атрибут **VALIGN** определяет вертикальное выравнивание содержимого ячейки. Он может иметь следующие значения: **TOP** (по верхнему краю), **MIDDLE** (посередине) и **BOTTOM** (по нижнему краю). По умолчанию – **MIDDLE**.

Атрибут WIDTH определяет ширину ячейки в пикселях.

3. Вопросы для закрепления.

- a. Чем отличаются абзацы от заголовков?
- b. Какие бывают списки?
- c. Какие теги используются для изменения шрифта? Что такое «информация»?

4. Домашнее задание:

1. Создайте таблицу, которая в первой строке содержит одну ячейку, во второй – две, в третьей – три ячейки.
2. На практике наблюдайте эффект использования различных значений атрибута ALIGN тега .

Лекция 8

Тема: Гиперссылки

Цели: *Ознакомить учащихся с основными понятиями гиперссылки; правилами создания ссылок на различные типы документов.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Гиперссылки
 - 2) Универсальный идентификатор ресурсов URL
 - 3) Правила записи ссылок
3. Закрепление изученного.
4. Итог урока.

Ход урока.

16.Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

17.Изложение нового материала.

Как уже было сказано, гиперссылки – переходы к другим документам. Они являются основой HTML. Гиперссылки можно использовать для перехода не только к другим HTML-документам, но и другим объектам, которые можно разместить на компьютере, например, к видео- и аудиофайлам, архивам, рисункам и т.п.

Каждая ссылка состоит из двух частей. Первая – это то, что отображается браузером. Она называется **указателем ссылки**. Вторая часть – адресная, содержащая адрес объекта, к которому должен происходить переход. Эту часть называют **URL** (универсальный идентификатор ресурсов).

Когда вы щелкаете мышью по указателю ссылки, браузер загружает документ, адрес которого указан в URL.

Указателем ссылки может быть слово, группа слов или рисунок. Если указатель текстовый, то он обычно отображается браузером подчеркнутым синим шрифтом. При наведении курсора мыши на указатель курсор принимает форму руки, указывая, что это ссылка и можно произвести переход. Если указатель графический, внешне он не отличается от других рисунков, но при наведении курсора мыши на такой указатель, курсор также принимает форму руки.

УНИВЕРСАЛЬНЫЙ ИДЕНТИФИКАТОР РЕСУРСОВ URL

По своей сути URL – это адрес файла, к которому происходит переход. Указание адреса может быть относительным или абсолютным.

Когда в URL указывается полный путь к файлу, независимо от того, где этот файл находится (в Интернете или на жестком диске компьютера), то это абсолютное указание. Например: **http://www.mysite.ru/page.htm** – абсолютный адрес документа, находящегося в Интернете, **c:\web\doc1.htm** – абсолютный адрес документа на диске **c**.

Если в URL указывается не полный путь, а путь относительно адреса документа, в котором находится ссылка, то это относительное указание. Например, браузер отображает документ, абсолютный адрес которого **c:\web\doc1.htm**, в этом документе

имеется ссылка с адресом **pict/ris1.jpg**, то это означает, что на самом деле ссылка будет на документ по адресу **c:\web\pict\ris1.jpg**.

Когда надо сослаться на файл, находящийся на другом компьютере, тогда следует пользоваться абсолютным указанием, а если сослаться на файлы, находящиеся на том же компьютере, где и документ, содержащий ссылку, то лучше использовать относительное указание.

Правила записи ссылок

Для организации ссылки необходимо указать браузеру, что является указателем ссылки, а также определить адрес документа, на который происходит ссылка. Оба этих действия выполняются с помощью тега **<A>**.

Тег **<A>** имеет следующую спецификацию:

```
<A  
HREF="URL-адрес"  
NAME="имя ссылки"  
TARGET="объект для вывода"  
>
```

Атрибут **HREF** используется для задания адреса файла, к которому производится переход. Значением этого атрибута является текстовая строка, содержащая абсолютный или относительный URL-адрес.

Атрибут **NAME** предназначен для задания ссылке имени. Значением этого атрибута является короткая текстовая строка. Этот атрибут используется для ссылок внутри одного HTML-документа.

Атрибут **TARGET** позволяет определить, куда будет выводиться документ, на который происходит переход. Этот атрибут может иметь значение **_blank** – это означает, что документ будет выводиться в новом окне браузера.

Пример ссылки:

```
<A HREF="doc1.htm">Документ 1</A>
```

Браузер отобразит эту строку так:

Документ 1

При нажатии мышью на этой строке браузер загрузит и отобразит файл **doc1.htm**, т.е. “Документ 1” – это указатель ссылки, а “doc1.htm” – URL-адрес.

ВНУТРЕННИЕ ССЫЛКИ

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

Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Для этого в месте, куда потом будет производиться ссылка, надо поместить тег **<A>** с атрибутом **NAME**, определив этим атрибутом имя указателя. Например:

```
<A NAME="glava5"></A>
```

Обратите внимание, что в этом примере отсутствует содержимое тега **<A>**. Обычно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого в атрибуте **HREF** тега **<A>** помещается имя ссылки с префиксом **#**, говорящим о том, что это внутренняя ссылка. Например:

`Глава 5`

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

Можно совмещать внутренние ссылки со ссылками на другие документы. Например:

`Глава 5 Документа 1`

При нажатии на эту ссылку браузер откроет файл doc1.htm, найдет в этом файле указатель glava5 и выведет в окне просмотра соответствующую информацию.

ССЫЛКИ НА ДОКУМЕНТЫ РАЗЛИЧНЫХ ТИПОВ

Когда пользователь щелкает мышью на ссылке, указывающей на другой HTML-документ, то документ выводится непосредственно в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать дальше.

□ Браузер знает этот тип документа и умеет с ним обращаться. Например, если создать ссылку на графический файл формата GIF и щелкнуть мышью на ней, то браузер очистит окно просмотра и отобразит указанное изображение.

□ Браузер не распознает тип принятого документа и не знает, что с ним делать дальше. В этом случае он обратится к вспомогательным программам, имеющимся на компьютере пользователя. Если подходящая программа найдется, браузер запустит ее и передаст ей полученный документ для обработки. Например, если пользователь щелкнет на ссылке на видеофайл формата AVI, браузер загрузит файл, найдет программу для демонстрации AVI-файлов и запустит ее. Видеофайл будет показан в дополнительном небольшом окне.

Ссылки на ресурсы Интернета

Основное назначение HTML-документов – это глобальная компьютерная сеть Интернет. HTML-документ, размещенный в Интернете, становится Web-страницей. Чтобы обратиться к Web-странице, надо указать URL-адрес в такой форме: **http://sitename/docname**, где sitename – имя сайта, docname – имя документа. Например: <http://kotoz.newmail.ru/autor.htm>.

Можно на Web-странице разместить адрес электронной почты. Для этого URL-адрес указывается так: **mailto:address**, где address – это адрес почтового ящика. Например: <mailto:vasya@mail.ru>.

Также существуют специальные форматы URL-адреса для других ресурсов Интернета (FTP, TelNet, Newsgroup, Gopher, WAIS).

18. Вопросы для закрепления.

1. Что такое URL?
2. Что такое внутренняя ссылка и как она создается?
3. На какие файлы можно ссылаться в HTML-документах?

19. Домашнее задание: Создать HTML- документ с ссылками на три различных типа документов.

Лекция 9

Тема: Создание Web- узла с помощью мастера (редактор Front Page).

Цели: Ознакомить учащихся с основами создания web- страниц и web- узлов с помощью визуального редактора Front Page.

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Создание web- узла с помощью мастера
 - 2) Свойства страницы
 - 3) Проектирование и создание таблиц.
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Создание web-узла с помощью мастера

Для быстрого и удобного создания Web-сайта в FrontPage существуют мастера и шаблоны, которые позволяют автоматически создавать новые страницы с расширенными возможностями. В случае использования шаблонов новые сформатированные страницы образуются без вашего непосредственного участия. Создание же страницы с помощью мастера происходит в диалоговом режиме: *мастер* задает вопросы о создаваемой странице и строит ее на основании ваших ответов. Мастера и шаблоны находятся в главном меню **Файл**→**Новый**→**Сайт**. При открытии данного меню появляется следующий список (рис. 9.1):

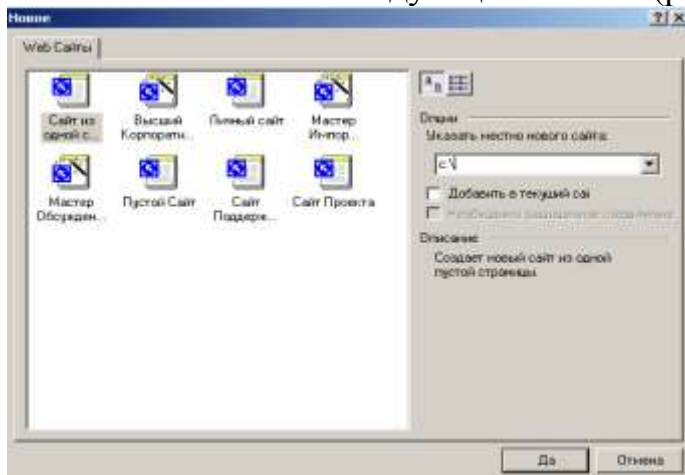


Рисунок 9.1

Сайт из одной страницы (одностраничный Web-сайт) - содержит только домашнюю страницу.

Высший корпоративный мастер - предоставляет информацию о компании (роль деятельности, вид выпускаемой продукции, предоставляемые услуги), а также сведения о том, как связаться с ней и как наладить обратную связь.

Личный сайт - содержит страницы с личной, профессиональной и контактной информацией.

Мастер импорта сайта - создается новый Web-сайт с содержимым, которое импортируется с другого сайта.

Мастер обсуждений в Веб - электронная доска объявлений, на которой одни пользователи оставляют вопросы, а другие — ответы на них.

Пустой сайт - Web-сайт без страниц (вы можете импортировать их с другого Web-сайта).

Сайт поддержки заказчика - содержит страницы и формы для поддержания полномасштабного диалога с клиентами.

Сайт проекта - предоставляет сведения о проекте: график выполнения, информация о тех, кто работает над проектом, время завершения и т.д.

Мастера и шаблоны очень удобны, но их использование приводит к созданию очень похожих друг на друга сайтов.

Создание структуры Web-сайта

Структура Web-сайта создается в режиме **Навигация**. Щелкните правой кнопкой на пиктограмме домашней страницы и в контекстном меню выберите команду **Новая страница**, чтобы создать новую пустую страницу. Пиктограмма новой страницы разместится под пиктограммой домашней страницы и между ними установлена связь (они соединены вертикальной линией). Это означает, что новая страница является дочерней по отношению к домашней странице.

Теперь давайте создадим ту структуру сайта, которую спроектировали ранее. Если мы хотим создать страничку более низкого уровня, то необходимо щелкнуть правой кнопкой на страничку выше, выбрать в контекстном меню **Новая страница** (рис. 9.2). Как только создается новая страница, ее следует тут же переименовать. Щелкните дважды на пиктограмме страницы (промежуток между щелчками должен быть таким, чтобы система не интерпретировала эти два нажатия как двойной щелчок) или в контекстном меню выберите пункт **Переименовать**. Имя должно быть набрано только *латиницей и только маленькими буквами*.

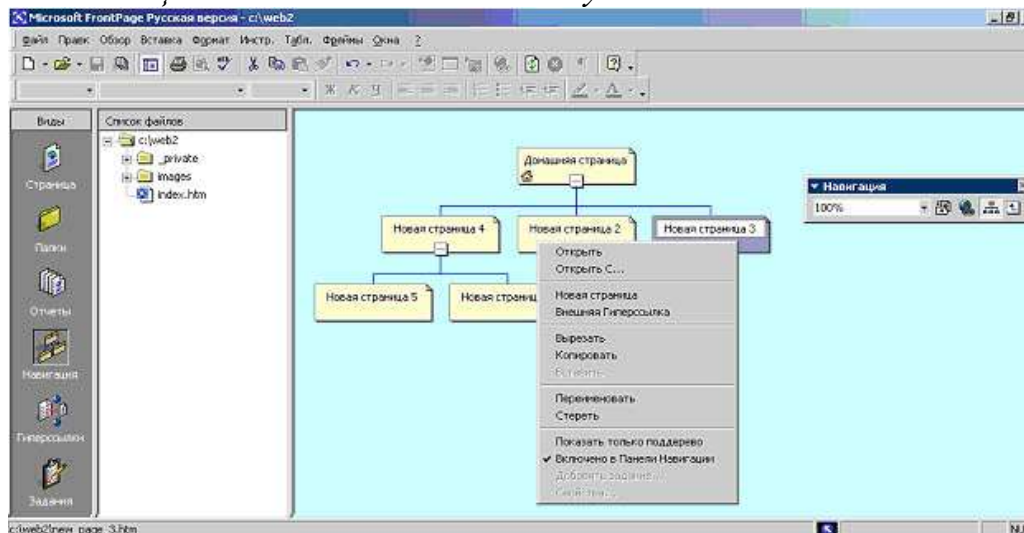


Рисунок 9.2

Так же можно изменять связи между страницами. Для этого выберете любую страницу самого нижнего уровня и, удерживая кнопку мыши, перетяните ее на одну ступеньку выше. Вы увидите, что связи изменятся. Верните страницу на прежнее место.

Внимание! Все это делается только в режиме просмотра **Навигация**.

Если у вас есть страницы третьего уровня и ниже, то лучше всего эти страницы поместить в отдельные папки, у которых имена будут такие же, как у файлов, с которых будут организованы ссылки на данные страницы.

В нашем случае создадим папку PROBA. Для этого перейдем в режим просмотра **Папки**. На левой панели **Список файлов** в любом месте нужно щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать опцию **Новая папка**. Созданную папку переименовываем в PROBA.

Затем перейдем обратно в режим просмотра **Навигация** и перенесем файлы нижних уровней одной из страниц в созданную папку. Затем переименуем папку. Присвоим ей имя родительской страницы. Родительской называется страница, которая связана с другой страницей и расположена выше.

Общие области

*Создавайте страницы сайта на основе одного макета. Макет всего сайта создается с помощью **общих областей**. Их размещают на верхнем, нижнем, левом и (реже) правом краях страницы. Они называются общими, потому что по умолчанию отображаются на всех страницах Web-сайта.*

Общие области часто включают навигационные панели. Обычно:

- Верхняя общая область содержит заголовок страницы
- Нижняя общая область содержит информацию об авторских правах, контактную информацию и др.
- Левая - содержит ссылки на каждую страницу

Примечание. Общие области редко размещают в правой части страницы Web-сайта, поскольку в этом случае пользователи могут их не увидеть.

Создание общих областей.

Активизируйте команду **Общие границы** меню **Формат**. Откроется окно **Совместные границы** (рис. 9.3.). Установим в нем нужные опции. Также можно вызвать это окно с помощью контекстного меню **Общие границы**.

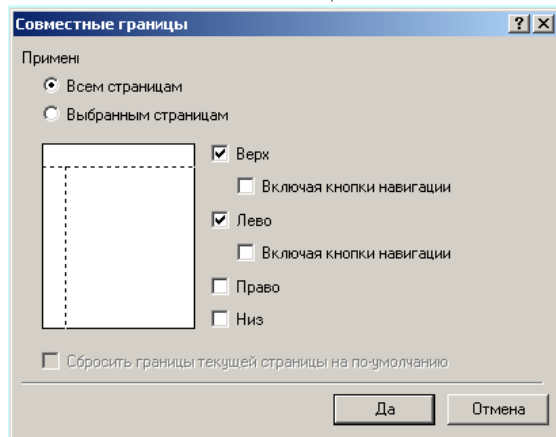


Рисунок 9.3

Редактирование содержимого общих областей.

Когда вы редактируете содержимое общей области на одной странице, выполненные вами изменения распространяются на все страницы, которые содержат данную область.

1. Щелкните левой кнопкой мыши на содержимое любой области.
2. Чтобы добавить текст в общую область, введите его обычным способом.
3. Когда вы введете и отредактируете содержимое общей области, щелкните за ее пределами левой кнопкой мыши.

Свойства страницы

Оформление каждой страницы осуществляется в режиме **Страница** и начинается с настройки **свойств страницы**.

После выполнения команды **Page Properties** в контекстном меню, которое вызывается нажатием правой кнопки мыши в окне страницы появляется одноименная панель со множеством закладок.

Основные свойства:

- Местонахождение – URL-адрес данной страницы;
- Название – имя страницы;
- основное место, конечный фрейм – окошки работы с фреймами;
- фоновый звук – местонахождение файла и время воспроизведения музыки;
- управляющие скрипты дизайнера – выбор платформы (рекомендуется оставить все как есть).

Фон:

- фоновый рисунок – задание фонового рисунка на страницу;
- водяной знак – при включении данного флажка фоновый рисунок не прокручивается вместе с текстом;
 - стиль ролловера – здесь можно изменить стиль гипертекста, который появляется после наведения на ссылку курсора;
 - цвета – можно задать цвет фона всей страницы, текста, гиперссылок;
 - получить фоновую информацию из другой страницы – если Вы хотите использовать один и тот же рисунок для нескольких или всех страниц сайта, то можно загрузить рисунок с первой страницей, а затем с помощью этого окошка, брать рисунок для других страниц (тем самым экономя время загрузки).

Для создания фона страницы иногда используют уже готовые изображения, текстуры, которые можно загрузить из Internet или взять из графической библиотеки.

Текстура обычно представляет собой небольшое графическое изображение (оптимальный размер 96 x 96 пикселей). При создании фона FrontPage “размножает” текстуру, создавая мозаику. Текстура должна быть создана так, чтобы при построении мозаики “стыки” между отдельными текстурами не были видны.

Поля:

В окне **Поля** можно изменить расстояние от верхнего левого угла браузера до информации, расположенной на страничке. Для этого необходимо активировать опции **Указать верхнее поле** и **Указать левое поле**. В окошках можно набрать нужные расстояния в пикселях. Что бы убрать поля, необходимо поставить нули в обеих опциях.

Язык:

- в окошке **Язык страны** выбираем «русский».
- в окошке **Сохранить документ как** выбираем “кириллица” для кодировки Win-1251 или “кириллица (KOI8-R)” для кодировки KOI. Некоторые сервера требуют убирать тэг **charset**, который прописывает этот пункт в HTML-коде. В этом случае просто укажите **нет**;
- окошко **Перезагрузить текущий документ как** предназначено для перекодировки странички (по умолчанию – **Автоматическая кодировка**).

Рабочая группа:

На этой закладке делаются настройки для создания сайта сразу несколькими разработчиками.

Проектирование и создание таблицы

Во FrontPage используются два средства, позволяющих разбивать страницу на области, содержащие текст и графические изображения - таблицы и кадры (фреймы). Следует заметить, что предпочтительнее использовать таблицы, чем фреймы.

Важнейшим средством разработки макета Web-страницы является **таблица**. Функции таблиц при создании Web-страниц разнообразны. В первую очередь следует выделить следующие из них:

- выравнивание данных путем ввода их в ячейки таблицы (при этом предоставляется возможность рисовать таблицу линиями различной толщины);
- представление текста в виде колонок;
- выравнивание в формах полей ввода и их названий;
- распределение текста и графики по разным колонкам;
- создание рамки вокруг текста или графических изображений;
- размещение графических изображений по обе стороны от текста (или со всех сторон) и наоборот;
- расположение текста по контуру графического изображения;
- создание цветного фона для текстового фрагмента или отдельного изображения;
- включение таблиц в другие таблицы.

В процессе создания таблицы в диалоговом окне **Вставить таблицу** устанавливаются значения таких свойств: количество строк и столбцов, ширина таблицы, толщина линий, разделяющих ячейки и ограничивающих таблицу, величина интервалов между содержимым и внутренним краем границ ячейки, выравнивание таблицы и т.д. В дальнейшем значения свойств таблицы можно изменить. В любой момент можно удалить как одну, так и несколько строк или колонок, объединить смежные ячейки, сформатировать особым образом содержимое одной ячейки или выделенного диапазона ячеек и т.д. При вводе данных размер ячейки увеличивается по вертикали и по горизонтали в пределах ограничений, установленных для таблицы и столбца.

Вы можете задать ширину таблицы как в пикселях (фиксированная ширина), так и в процентном соотношении относительно ширины окна браузера или кадра, в который включена страница. Если ширина отдельного столбца или всей таблицы задана в пикселях, то отображение этой таблицы в браузере будет зависеть от разрешения экрана, заданного на компьютере пользователя.

Выделите домашнюю страницу и перейдите в режим просмотра **Страница**.

Откройте меню **Таблица** и активизируйте команду **Вставка →Таблица**. В результате откроется диалоговое окно **Вставить Таблицу**. В этом окне отображены значения свойств таблицы, установленные по умолчанию. Элементы диалогового окна **Вставить таблицу** описаны ниже:

Строк	Количество строк в таблице
Колонок	Количество столбцов в таблице
Привязка	Способ выравнивания таблицы (по умолчанию задано выравнивание по левому краю)
Размер	Толщина разделительных и ограничительных линий в пикселях. Значение по умолчанию — 0. Если это значение не изменять, то в браузере рамка таблицы вообще не отображается, а во FrontPage обозначается пунктирной линией

Дополнение ячейки	Величина интервалов (в пикселях) между краями ячейки и ее содержимым (значение по умолчанию — 0)
Промежутки ячейки	Интервал (в пикселях) между смежными ячейками (значение по умолчанию — 0)
Указать ширину	Ширина таблицы. Это значение задается либо в пикселях, либо в процентах от ширины окна браузера или кадра, в который включена страница

Таблица 9.1

Переходить из ячейки в ячейку можно клавишей клавиатуры **[Tab]**, причем если установить курсор в последней ячейке и нажать клавишу **[Tab]**, то будет добавлена новая строка, а курсор переместится в первую ячейку новой строки.

В том случае, если сайт будет разрабатываться для монитора с разрешением 800x600 пикселей, самым оптимальным будет поставить ширину таблицы – 760 пикселей.

Свойства таблицы.

Значения, присваиваемые параметрам таблицы, определяют ее внешний вид.

Щелкните правой кнопкой мыши на таблице и в контекстном меню выберите команду **Свойства таблицы**, вследствие чего откроется одноименное диалоговое окно (рис.9.4).

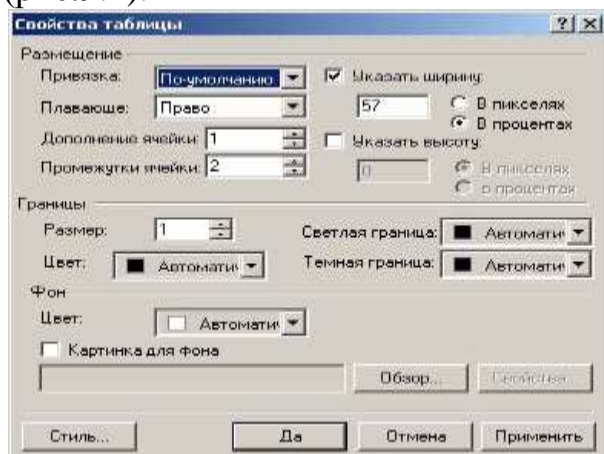


Рисунок 9.4

Для изменения свойств таблицы нужно выполнить щелчок правой кнопкой мыши на таблице и, в контекстном меню выбрать пункт **Свойства таблицы** и заполнить диалоговое окно:

- **привязка** - размещение таблицы относительно других элементов страницы,
- **плавающее** - обтекание таблицы,
- **указать ширину** – определяется ширина таблицы в пикселях или процентах,
- **указать высоту** – определяется высота таблицы в пикселях или процентах,
- **дополнение ячейки** - расстояние между границей ячейки и ее содержимым,
- **промежутки ячейки** - интервал между ячейками,
- **размер, цвет, светлая граница, темная граница** – толщина рамки таблицы и ее цветовое оформление,

- **цвет** – цвет фона,

Свойства ячейки.

Выполните щелчок правой кнопкой мыши на любой ячейке таблицы и, в контекстном меню, выберите команду **Свойства ячейки**. Откроется одноименное диалоговое окно. Содержащиеся в нем элементы описаны ниже:

Привязка по горизонтали	Выравнивание содержимого ячейки по горизонтали. Возможны следующие варианты: по левому краю, по центру, по правому краю . По умолчанию задано выравнивание по левому краю
Привязка по вертикали	Выравнивание содержимого ячейки по вертикали. Возможны следующие варианты: по верхнему краю, по центру, по опорной линии, по нижнему краю . По умолчанию задано выравнивание по центру
Заполнение	В этом поле указывается, на сколько строк вы хотите растянуть данную ячейку
Заполнение ячеек	В этом поле указывается, на сколько столбцов вы хотите растянуть данную ячейку
Ячейка заголовка	Форматирование ячейки как заголовка строки или столбца (тексту ячейки назначается полужирное начертание)
Без переноса	Запрет расстановки переносов в пределах ячейки в Web-браузере; в противном случае текст разбивается, если окно браузера слишком узкое для его полного отображения
Указать ширину	Определение ширины ячейки в пикселях или в процентах от общей ширины таблицы (только в том случае, если установлена опция ширина таблицы). Если указанная опция выключена, ширина ячейки автоматически меняется в зависимости от длины вводимого текста или длины размещаемого в ней объекта
Указать высоту	Определение высоты ячейки в пикселях или в процентах от общей высоты таблицы (только в том случае, если активна опция высота таблицы). Если указанная опция выключена, высота ячейки автоматически меняется таким образом, чтобы вводимый текст или размещаемый в ней объект поместился полностью
Границы	В этой области окна можно определить цвет для разделительных и ограничительных линий, которые могут быть как одноцветными, так и двухцветными. В первом случае используйте поле ввода граница , во втором — два любых поля из трех предложенных. В последнем случае рамка получается объемной
Фон	Создание фона для ячейки (как цветного однотонного, так и содержащего изображения)

Таблица 9.2

Внимание! Значения свойств можно изменить как для одной ячейки, так и для выделенного диапазона.

При изменении ширины ячейки следите за тем, чтобы общая ширина ячеек в строке не превысила 100%, иначе часть страницы будет выходить за область экрана и появится полоса прокрутки.

Щелкните на любой ячейке таблицы, откройте меню **Таблица**. Данное меню содержит четыре подменю. Команды **Таблица** описаны ниже:

Рисов. Таблица	Позволяет начертить таблицу подобно тому, как она рисуется карандашом на листе бумаги
Вставка/Таблица	При активизации данной команды открывается диалоговое окно Вставка Таблица. В нем можно задать параметры создаваемой таблицы, которая будет вставлена в той позиции, где находится курсор (если курсор расположен в ячейке другой таблицы, вторая таблица разместится в ней)
Вставка/Строки или колонки	В диалоговом окне Вставить строки или колонки определяется количество строк и столбцов, которые можно вставить выше, ниже, правее или левее выделенной области
Вставка/Ячейки	Вставляет новую ячейку слева от выделенной ячейки, которая смещается вправо
Вставка/Надпись	Предназначена для вставки пустой строки над таблицей. Данная строка связана с таблицей: при выделении или удалении таблицы заголовков также выделяется или удаляется. По умолчанию для заголовка таблицы установлено выравнивание по центру, но его можно выровнять по левому или по правому краю
Удал. ячейки	Служит для удаления выделенных ячеек
Слить клетки	Позволяет объединить несколько ячеек как в строке, так и в столбце
Разъед. клетки	В диалоговом окне можно задать разделение выделенных ячеек как на строки, так и на столбцы
Выбор/ Ячейка, Строка, Таблица	Первая команда служит для выделения текущей ячейки, вторая — текущей строки, третья — текущего столбца, а четвертая — всей таблицы
Распределить строки равномерно; Распределить колонки равномерно	Первая команда позволяет установить одинаковую ширину для выделенных строк, вторая — для выделенных столбцов
АвтоПос.	Уменьшает ширину каждого столбца до тех размеров, которые непосредственно занимает находящаяся в них информация. Активизация данной команды приводит к тем же последствиям, что и отключение опции Ширина в диалоговом окне Вставка Таблица , т.е. ширина ячеек автоматически будет меняться в зависимости от объема вводимого текста или ширины размещаемого в них объекта
Перевод/Текст в таблицу	Позволяет преобразовать выделенный текст в таблицу. При активизации этой команды открывается диалоговое окно для выбора разделительного символа. Для разделения содержимого столбцов следует использовать запятые, а для разделения строк — маркеры абзацев
Перевод/ Таблицу в текст	Позволяет преобразовать табличные данные в текст, при этом содержимое каждой ячейки образует отдельный

	абзац
Свойства/Таблица, Ячейка, Надпись	Эти три команды предоставляют доступ к окнам свойств заглавия таблицы Свойства надписи , выделенного диапазона Свойства ячейки и таблицы Свойства таблицы

Таблица 9.3

Теперь давайте перейдем на разрабатываемый Web-сайт и поработаем с таблицей на главной страничке, вставим нужные картинки и текст.

Когда мы находимся в режиме *Страница*, то доступны команды меню, отвечающие за редактирование содержимого Web-страницы. В этом режиме окно FrontPage похоже на окно текстового редактора Microsoft Word.

Изображения на Web-странице

Существует два основных способа, позволяющих вставлять на страницу графические изображения:

- с помощью диалогового окна **Свойства страницы** задается фон страницы.

- с помощью диалогового окна **Рисунок**, которое открывается через меню **Вставка** на страницу добавляется графическое изображение.

Параметры изображения устанавливаются в диалоговом окне **Свойства рисунка**. Окно открывается посредством одноименной команды контекстного меню. В этом окне можно набить текст, который выводится при просмотре страницы в браузере в случае отсутствия графического изображения на Web-сайте (этот текст отображается, если в браузере установлена опция загрузки страниц без изображения).

На вкладке **Основное** определяются следующие параметры:

- **обзор, изменить** – местоположение файла,
- **тип** – формат графического файла (“GIF”, ”JPEG”, ”PNG”),
- **прозрачный** – эффект прозрачности в GIF-изображении,
- **чередов. строк** – чересстрочное отображение при загрузке GIF-изображения,
- **качество** – степень качества JPG-изображения
- **прогресс. проходы** – число этапов, за которое отобразится JPG-изображение
- **низкое разр.** - изображение с низким разрешением, которое отображается при отсутствии оригинала или в процессе его загрузки

- **текст** - текст, который будет отображаться при отсутствии изображения или в процессе его загрузки,

- **адрес, обзор** – адрес гиперссылки,
- **конечный фрейм** – уточнение параметров гиперссылки при использовании фреймовой структуры.

▪ На вкладке **Внешний вид** определяются следующие параметры:

- **привязка** – позволяет организовать обтекание изображения другими объектами. Для отмены обтекания текстом какого-либо объекта:

Вставка→**Прерыватель**→**Очистить оба отступа**

- **толщина границ** – размер рамки вокруг графического изображения,
- **горизонт. расстояния, вертикал. Расстояния** - величина отступов от графического изображения,
- **ширина, высота** – размер графического изображения,

▪ **хранить коэффициент** – поддержка пропорций графического изображения.

Размещение текста на Web-странице

Front Page имеет интерфейс текстового процессора Word. Текст может быть введен с клавиатуры или скопирован из другого документа через буфер обмена и вставлен на страницу с помощью команды “**Правк→Встав**”.

Изменить вид шрифта выделенного фрагмента текста можно с помощью кнопок **панели инструментов** (рис. 9.5) или в диалоговом окне **Формат→Шрифт**



Рисунок 9.5

При редактировании выделенных абзацев можно использовать команды **панели инструментов** (рис.9.5)



Рисунок 9.5

и команды меню **Формат**:

▪ **Абзац** – выбирается привязка (justify – выравнивание справа и слева), сдвиг, межстрочное расстояние.

▪ **Маркеры и номера** – подпункты **простые маркеры** и **номера** позволяют выбрать тип маркера. Подпункт **маркеры изображения** позволяет заменить типовые маркеры на графические изображения. Для этого нужно активизировать **Указать рисунок→Обзор**, выбрать графический файл на диске с помощью кнопки

▪ **Границы и тени** – В подменю **границы** можно выбрать стиль границы, ее цвет и ширину, расстояние до текста.

▪ В подменю **тени** определяется цвет фона, текста. В качестве фона можно выбрать графическое изображение.

Для перехода на новую строку в текущем абзаце используется команда **Вставка→Прер→Да**.

В FrontPage форматирование выполняется в режиме WYSIWYG (что видите – то и получаете).

Завершив формирование страницы, сохраните ее. Каждая страница сайта сохраняется отдельно командой **Файл→Сохранить**.

В том случае, если на страницу вставлялось графическое изображение, то появляется окно **Сохранить внедренные файлы**. Необходимо: **Изменить папку** и выбрать папку **images**.

Замечание. Не все шрифты поддерживают браузеры, и не все шрифты могут быть установлены на компьютере пользователя.

3. Вопросы для закрепления.

- 1) Для чего предназначен визуальный редактор Front Page?
 - 2) Как создается структура web – страницы.
 - 3) Как установить свойства web- страницы?
 - 4) Назовите способы редактирования текста
 - 5) Назовите способы создания и редактирования таблиц в редакторе
- 4. Домашнее задание:** Подготовить материалы для создания будущего сайта(текст, графику)

Лекция 10

Тема: Основные принципы работы с DW

Цели: *Ознакомить учащихся с основными возможностями редактора Dreamweaver; научить работать с основными окнами редактора и настраивать свойства сайта*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Запуск редактора, знакомство с окнами DW
 - 2) Определение и настройка сайта
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Dreamweaver MX 2004 – увлекательнейшая программа для создания Web-страниц. Ее возможности шире по сравнению с обычными функциями подобных редакторов. Dreamweaver позволяет создавать объекты и страницы с использованием языка HTML 4 со всеми его формами, фреймами и прочим, а также DHTML, поддерживает каскадные таблицы стилей, слои, объекты и возможность вставки интерактивных элементов JavaScript и ActiveX, апплетов действия JavaScript, а также добавление анимации Macromedia Flash и Shockwave

Разработчику Web-проекта предлагаются гибкий и мощный инструментарий управления сайтом, включая, встроенный полноценный FTP-клиент, визуальные карты сайтов и контроль над ссылками. Возможности Dreamweaver MX 2004 значительно облегчают процесс создания страниц и управления сайтом.

Запуск редактора

Для запуска редактора Dreamweaver MX 2004 необходимо выполнить следующую последовательность действий:

Пуск – Программы – Macromedia – Macromedia Dreamweaver MX 2004.

Выбор интерфейса

При первом запуске **Dreamweaver MX 2004** предлагается два варианта вида рабочей области редактора:

- Designer (Дизайнерский);
- Coder (Кодировщик).

Чем они отличаются — понятно из их названий. Это окно появляется, как правило, только при первой загрузке редактора. В дальнейшем настройки можно будет изменить командой меню Edit • Preferences (Редактировать • Предпочтения). В диалоговом окне Preferences (Предпочтения) в списке Category (Категория) выберите пункт General (Общие) и нажмите кнопку Change Workspace (Смена рабочей области). После этого откроется окно Workspace Setup (Установка рабочей области).

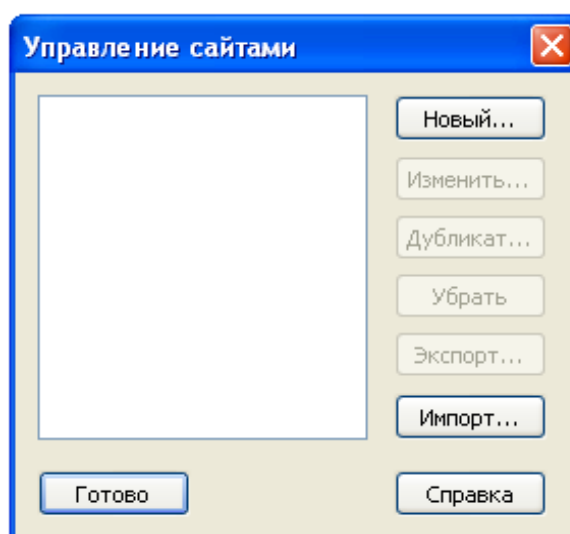
Сразу следует выбрать дизайнерский вид рабочей области редактора. Так будет проще работать. После того как выбор сделан, загрузка редактора продолжится. По окончании загрузки на экране появится рабочая область редактора. В ее центре размещается так называемое стартовое окно редактора.



Определение и настройка сайта

Определение (или регистрацию) нового сайта в Dreamweaver MX рекомендуется проводить еще на начальном этапе построения веб-страниц, входящих в него. Во-первых, так будет удобнее потом, это позволит избежать путаницы при сортировке веб-страниц. Во-вторых, некоторые очень полезные функции редактора будут доступны только после определения сайта. И наконец, на этапе опубликования сайта в Интернете не возникнет проблем с адресацией ссылок и демонстрацией рисунков. А еще при использовании на разных веб-страницах одних и тех же рисунков (например, логотипа фирмы или значка счетчика посещений) понадобится не множество копий таких рисунков для каждой страницы, а всего лишь один рисунок.

Итак, для начала определения нового сайта выполняем команду меню Site • Manage Sites (Сайт • Управление сайтами). Эта команда меню открывает диалоговое окно Manage Sites (Управление сайтами). Диалоговое окно, которое называется менеджером сайтов, отображает все уже имеющиеся проекты (сайты), а также дает возможность определить новый сайт (рис. 3.1). Для этого в колонке кнопок у правой границы окна щелкаем по самой верхней кнопке New (Новый).

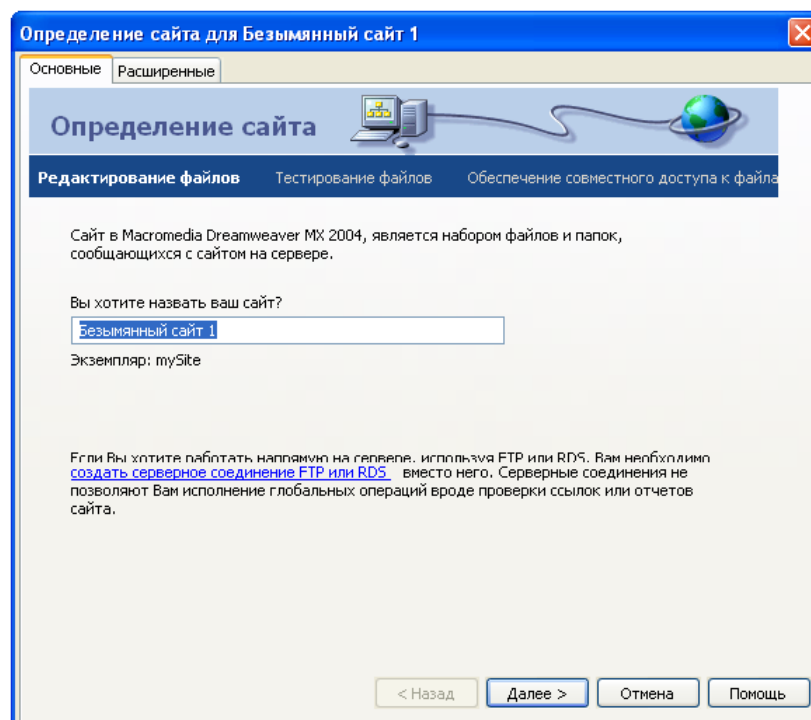


В появившемся списке следует выбрать Site (Сайт). Dreamweaver сразу же обозначит имя нового сайта как Unnamed Site (Безымянный сайт). Откроется новое диалоговое окно Site Definition (Определение сайта). Именно здесь и производится определение сайта. Определение сайта можно производить двумя способами —

автоматически и вручную. Этим способам соответствуют две вкладки окна — Basic (Основной) и Advanced (Расширенный). Определение сайта по первой вкладке — Basic — является наиболее простым. В пошаговом режиме вам требуется отвечать на вопросы или вводить требуемые данные. Здесь определяются лишь самые основные, необходимые настройки сайта. Понятно, что такой режим будет удобен начинающим.

Второй режим определения сайта — Advanced — является, как следует из названия, расширенным. Он больше подходит профессионалам, так как содержит более тонкие настройки сайта. Здесь пользователь уже сам выбирает, какие ему делать установки и в какой последовательности. Однако в любой момент времени вы можете перейти из одного режима в другой.

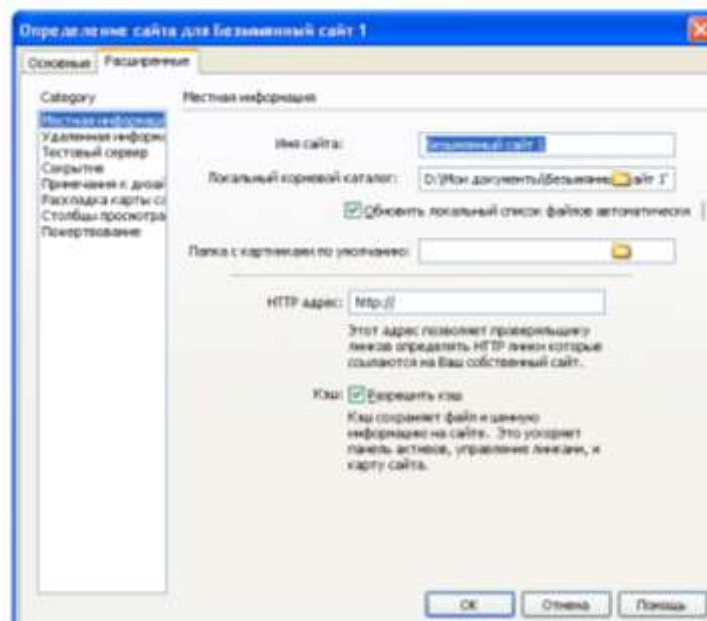
Теперь рассмотрим, какие **установки** необходимо сделать **при определении сайта**. Конечно, **во-первых, необходимо указать имя сайта**. В режиме определения сайта Basic определение имени сайта является **первым шагом**. Для **первого шага на вкладке Basic имеется всего одно поле ввода — это поле ввода имени сайта**. Следует заметить, что не стоит давать сайту слишком длинное имя, а писать лучше **латинскими буквами**. Ведь это имя — лишь условное имя той папки, в которую будут записываться все файлы, используемые в данном проекте, и никто, кроме вас, этого имени знать не будет. Для примера мы назовем сайт Site 1. Теперь нажимаем кнопку Next (Далее) в правом нижнем углу окна и переходим к следующему шагу. **На втором шаге предлагается определить, будем ли мы использовать сервисные технологии или нет**. А пока отказываемся от их использования и переходим далее.



Третий шаг является одним из важнейших после определения имени сайта. Здесь мы выбираем, будем ли создавать сайт на своем компьютере (Edit local copies on my machine, then upload to server when ready), или на сервере по локальной сети (Edit directly on server using local network). Как и рекомендует Dreamweaver, мы выбираем первый вариант. **В следующем поле указываем папку (каталог)**, в которой будут храниться все документы нашего сайта. Удобнее, когда имя сайта совпадает с именем папки. Если такой папки еще нет, создаем ее тут же. Следует особенно подчеркнуть важность этого шага в ситуациях, когда планируется создание многостраничного сайта или вставка на веб-страницу рисунков и гиперссылок. Определение корневой

папки сайта необходимо для создания относительных ссылок (то есть ссылок, не привязанных к конкретному компьютеру), чтобы эти ссылки также работали и на удаленном сервере.

Вновь нажимаем кнопку Next для перехода к четвертому шагу. **На четвертом шаге** указываем, каким **способом будем производить загрузку файлов нашего сайта на сервер**. Мы выбрали самый типичный для этого случая протокол FTP, указали адрес провайдера, который предоставляет нам место для сайта, а также регистрационное имя (логин) и пароль для выхода в Интернет. Впрочем, этот шаг важен только в том случае, если предполагается загрузка сайта на сервер посредством Dreamweaver MX 2004, а выполнить его можно и непосредственно перед самой загрузкой. Здесь же, щелкнув на кнопке Test Connection (Тест соединения), можно проверить работоспособность указанных нами данных. На заключительном этапе Dreamweaver MX 2004 подытожит все введенные данные и покажет их вам. На любом из шагов вы всегда можете вернуться назад и что-либо исправить. Режим определения сайта *Advanced* является более профессиональным и позволяет делать специализированные установки. Этот режим несколько отличается от только что рассмотренного. У левой границы окна расположен список Category. Перечисленные в нем категории по сути являются названиями шагов в режиме Basic (основные). Выбираем нужную категорию и производим необходимые установки. По умолчанию открыта категория Local Info (Расширенные). Здесь указывается имя сайта (Site Name), локальная корневая папка (Local root folder). Кроме того, можно включить автоматическое обновление сайта на сервере.



Следующая категория — Remote Info (Удаленная информация) — задает параметры копии сайта на удаленном сервере. Здесь очень важно определить режим загрузки файлов сайта на сервер (Access). От этого будут зависеть все остальные настройки данной категории. Если выбран вариант загрузки по FTP, то все последующие настройки будут связаны именно с этим протоколом. Так, далее в поле FTP host вводим адрес FTP-сервера, например ftp://ftp.narod.ru. Этот адрес можно узнать у провайдера, предоставляющего вам место на сервере для сайта. Ниже вводятся имя корневой директории сайта — Host directory, имя пользователя и пароль, под которыми осуществляется вход в сеть. А рядом расположена кнопка Test (Тест), нажав на которую вы сможете тут же проверить работоспособность введенных данных. Если FTP-сервер, с помощью которого производится копирование файлов

сайта на сервер, поддерживает только пассивный протокол FTP, то следует установить флажок Use passive FTP (Использовать пассивный FTP). Если используется прокси-сервер, то следует установить флажок Use firewall. При использовании защищенной версии протокола SFTP установите флажок Use Secure FTP (SFTP).

Все, что мы рассматривали до сих пор, это первоначальные настройки сайта, после которых можно приступить непосредственно к его построению. Нажимаем кнопку ОК. Dreamweaver сам создаст папку сайта, а имя сайта появится в списке в окне Manage Sites. Теперь папку сайта и все хранящиеся в ней документы можно будет увидеть на панели File.

Dreamweaver — программа с многодокументным интерфейсом или просто многодокументная программа. Это значит, что вы можете открыть в одном и том же окне программы сразу несколько документов. В этом случае окна, содержащие открытые документы, открываются внутри большого окна самой программы. К многодокументным приложениям также относятся Microsoft Word и Adobe Photoshop.

В отличие от них, программы с однодокументным интерфейсом (однодокументные программы) могут открыть только один документ; чтобы открыть второй, нужно запустить вторую копию программы. Примерами однодокументных приложений являются, в частности, текстовый редактор Microsoft WordPad и графический редактор Microsoft Paint, поставляемые в составе Windows.

Главное окно служит "вместилищем" для превеликого множества других окон, содержащих как открытые Web-страницы, так и различные инструменты, предназначенные для работы с ними. Также в главном окне находится строка главного меню, с помощью которого вы сможете получить доступ ко всем возможностям Dreamweaver.

Окно документа Dreamweaver служит для отображения открытой Web-страницы. Как и любая другая многодокументная программа, Dreamweaver может открыть в главном окне сколько угодно окон документов. Вы можете перемещать, свертывать и разворачивать эти окна и изменять их размеры, в общем, проделывать с ним те же манипуляции, что и с любым другим окном Windows. Единственное исключение: вы не можете "вытащить" ни одно из этих окон за пределы главного окна программы (его еще называют родительским окном).

3. Вопросы для закрепления.

- 1) Что такое Dreamweaver?**
- 2) В каком окне настраиваются свойства web-сайта .**
- 3) Назовите основные настройки сайта**

4. Домашнее задание: Подготовить доклад на тему: «Назначение и возможности визуальных редакторов, их сравнительный анализ»

Лекция 11

Тема: Работа над web-сайтом

Цели: *Ознакомить учащихся с основами создания web – страниц в редакторе DW; настройка свойств web – страницы.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Создание новой web – страницы
 - 2) Настройка свойств веб-страницы
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Создание новой веб-страницы

Теперь мы переходим к созданию нашей первой веб-страницы. На самом деле в Dreamweaver MX новый документ можно создать несколькими способами — «с нуля» или на основе шаблонов. Создать пустую веб-страницу в Dreamweaver MX File • New (Файл • Новый). Эта команда, как и комбинация клавиш Ctrl+N, открывает диалоговое окно New Document (Новый документ). Диалоговое окно New Document по своему содержанию аналогично стартовому окну. На вкладке General (Общие) выбираем из списка Category (Категория) категорию создаваемого документа — Basic page (Основная страница). Чуть правее откроется новый список Basic page. Именно в этом списке выбираем тип основной страницы. Для создания простой HTML-страницы следует выбрать тип HTML.

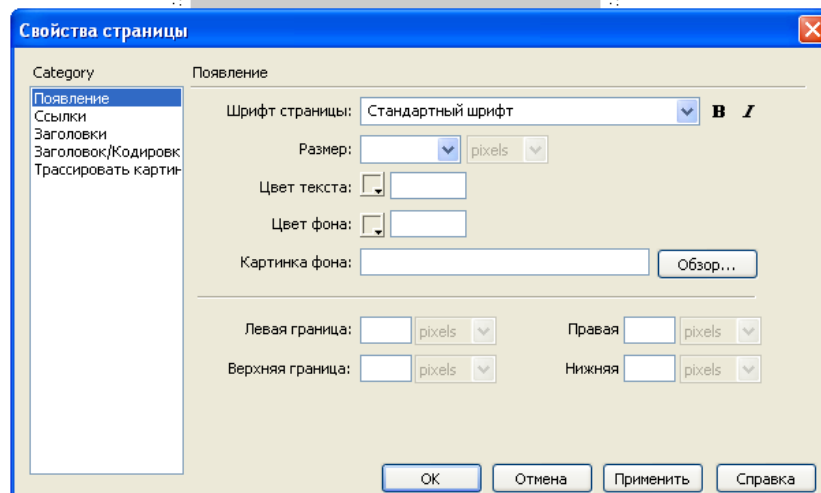
Нажимаем кнопку Create (Создать) — и страница будет создана. По умолчанию новая страница будет названа Untitled-1. Это имя документа будет отображаться на «закладке» у верхней границы документа. Одновременно с созданием новой страницы Dreamweaver MX 2004 напишет необходимый начальный HTML-код.

Создание на основе шаблона: В диалоговом окне New Document из списка Category выбираем пункт Page Designs (Дизайн страницы). В открывшемся правее одноименном списке мы увидим коллекцию шаблонов типичных веб-страниц. Еще правее расположено окно Preview (Предварительный просмотр), в котором можно увидеть миниатюру выбранного шаблона. Нажимаем кнопку Create — и веб-страница готова! Вы увидите в рабочей области веб-страницу, созданную на основе выбранного шаблона, причем эта веб-страница будет отображаться в дизайнерском режиме (Design). Вам остается только вставить свои картинки и текст, а потом сохранить получившийся результат.

Настройка свойств веб-страницы

Итак, мы начали оформлять нашу, пока еще пустую, веб-страницу. В начале выполним команду меню Modify • Page Properties (Изменить • Свойства страницы). По этой команде откроется диалоговое окно Page Properties (Свойства страницы)

Именно в этом окне мы немного изменим тот HTML-код, который Dreamweaver MX 2004 самостоятельно написал при создании страниц. Мы введем параметры в соответствующие поля диалогового окна Page Properties.



В правой части окна расположен список Category — здесь выбираем нужную категорию параметров веб-страницы. Это самая основная категория при настройке веб-страницы. Именно здесь мы указываем параметры, которые определяют внешний вид нашей веб-страницы.

В первом раскрывающемся списке Page font (Шрифт страницы) мы выберем шрифт, которым будем писать основной текст на нашей странице. В этом списке отображаются все шрифты, установленные на данном (вашем) компьютере.

Но не стоит выбирать экзотические шрифты — ведь у пользователей, которые будут посещать вашу страничку в сети Интернет, могут быть не установлены такие шрифты, и тогда они не разберут тот текст, который им адресован. Поэтому необходимо выбрать наиболее распространенную группу шрифтов — Arial, Helvetica, sans-serif или Times New Roman. Здесь же можно указать стиль написания текста — полужирный или курсивный (кнопки B — Bold и I — Italic).

В следующем ниже раскрывающемся списке можно ввести или выбрать из предлагаемого списка размер шрифта (Size) и единицы измерения этого размера. Мы выбираем наиболее распространенный для веб-страниц размер шрифта — 10 пикселей.

Еще ниже выбираем цвет шрифта, которым будет окрашен основной текст. Перед тем как его выбрать, следует определиться с общей цветовой гаммой страницы. Надо также учесть цвета ссылок, если вы предполагаете разместить их на странице. Цвет текста не должен сильно контрастировать с цветом фона. А цвет ссылки не должен сильно отличаться от цвета основного текста, хотя ссылка и должна быть хорошо заметна на странице. Определившись с этим, в поле ввода Text color (Цвет текста) вводим код цвета либо выбираем подходящий образец цвета на раскрывшейся палитре. А в следующей ниже строке делаем аналогичные установки, но уже для цвета фона в поле ввода Background color (Цвет фона). На самом деле для оформления фона совсем не обязательно использовать однотонную раскраску. Фон веб-страницы можно «заполнить» специально подготовленной или подобранной картинкой. В качестве такой картинки вполне можно использовать уменьшенный логотип компании. Чтобы он не казался слишком пестрым, в графическом редакторе его можно немного высветлить или размыть.

Для заполнения фона такой картинкой в поле Background image (Фоновый рисунок) указываем путь к фоновому рисунку или его имя. Следует заметить, что фоновый рисунок имеет преимущественное значение перед фоновым цветом.

Поэтому если указать оба параметра, то на веб-странице можно будет наблюдать только фоновый рисунок.

После разделительной черты следуют поля ввода для установки четырех параметров, определяющих величину полей между веб-страницей и границами окна браузера. Дело в том, что по умолчанию (то есть когда эти поля не заполнены) веб-страница в окне браузера обычно отображается с некоторыми отступами сверху и по бокам. Чтобы веб-страница заполняла все пространство окна браузера, надо ввести в следующие поля нулевые значения. Итак, в нижней части окна Page Properties находятся следующие поля ввода значений:

- Left margin (Левое поле) — определяет отступ от левой границы;
- Right margin (Правое поле) — определяет отступ от правой границы;
- Top margin (Верхнее поле) — определяет отступ сверху;
- Bottom margin (Нижнее поле) — определяет отступ снизу.

Здесь же можно установить единицы измерения отступов. Возникает резонный вопрос: зачем надо вводить две пары значений, задающих по существу одно и то же? Дело в том, что параметры Left margin и Top margin воспринимаются браузером Internet Explorer, но игнорируются Netscape Navigator. И наоборот — параметры Right margin и Bottom margin отлично работают в Netscape Navigator, но не работают в Internet Explorer. Поэтому чтобы веб-страница одинаково хорошо отображалась во всех браузерах, следует ввести все четыре параметра.

Следующая категория — Links (Ссылки) — предусматривает оформление ссылок (если они есть на веб-странице). В начале работы с веб-страницей эта категория необязательна. К ней можно будет обратиться тогда, когда будут уже расставлены нужные ссылки и создан общий дизайн страницы. Мы коснемся этой категории несколько позже.

Среди других категорий обратим внимание на Title/Encoding (Заголовок/Кодировка). На самом деле это одна из важнейших категорий. С ее помощью мы будем определять кодировку веб-страницы и ее заголовок.

В самое верхнее поле ввода Title (Заголовок) вводим заголовок нашей веб-страницы, который будет отображаться на самой верхней панели окна браузера, обычно на синем фоне. Этот заголовок отражает суть нашей страницы — это как название книги. Поэтому он должен быть лаконичным. Хорошо, когда он совпадает с предполагаемым названием сайта. Заголовок можно написать и русскими буквами. Мы назовем наш первый проект коротко и ясно — Мой первый сайт.

Следующий ниже раскрывающийся список Encoding (Кодировка) задает кодировку страницы. Это и есть наиболее важный момент. Дело в том, что каждый символ, вводимый с клавиатуры и отображаемый на экране, имеет уникальный номер, который обычно называют кодом символа. Набор кодов вместе с их описаниями составляет так называемую кодировку. Кодировок существует очень много. Некоторые из них универсальные, некоторые поддерживают только определенные языки. Кроме того, разные операционные системы используют различные кодировки. Для русского языка наиболее распространенными являются кодировки КОИ-8 и Windows-1251. Для русскоязычных сайтов последняя кодировка наиболее

распространена. Именно ее мы и выбираем в раскрывающемся списке Encoding. Правда, здесь она значится как Кириллица (Windows).

Когда кодировка указана, нажимаем находящуюся рядом кнопку Reload (Перезагрузить). А по окончании всех установок свойств страницы нажимаем кнопку ОК и переходим к дальнейшему редактированию веб-страницы.

3. Вопросы для закрепления.

- 1) Назовите способы создания web страниц »?
- 2) Какие основные настройки страницы необходимо выполнить перед созданием web – страниц?

Лекция 12

Тема: Использование графических изображений

Цели: Ознакомить учащихся со способами добавления графических объектов на веб-страницу, заменитель графики – его назначение, с созданием графических ссылок.

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Вставка графики на веб- страницу
 - 2) Заменитель графики
 - 3) Создание графических ссылок
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

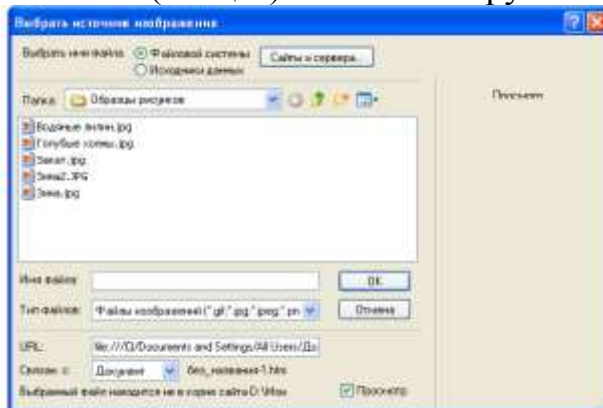
Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Вставка графики на веб-страницу

Вставку графического изображения на веб-страницу следует производить после точного подбора подходящего графического изображения, установки нужного размера, оптимизации и сохранения его в подходящем формате. Затем следует подобрать на веб-странице подходящее место для вставки этого изображения.

Для **вставки графического изображения** выполняем команду меню Insert • Image (Вставить • Изображение). Также можно использовать кнопку Image (Изображение) на вкладке Common (Общее) панели инструментов Insert (Вставка)



В поле ввода URL отображается относительный путь к файлу. Если веб-страница, на которую мы вставляем изображение, еще не сохранена, путь будет абсолютным. А после щелчка на кнопке ОК Dreamweaver MX 2004 предупредит вас, что данный файл будет записан с абсолютным путем. Раскрывающийся список Relative to (Относительно) позволяет выбрать, относительно чего определяется путь (документа или корневого каталога сайта).

В правой части диалогового окна располагается панель Image Preview (Предварительный просмотр изображения), на которой не только осуществляется предварительный просмотр выбранного изображения, но и отображаются его характеристики: размер, формат, объем и предполагаемое время загрузки. Впрочем, эту панель можно и закрыть. Для этого следует убрать флажок Preview image

(Предварительный просмотр изображения) в нижней части диалогового окна. Так как мы планируем вставить фотографию автора сайта, то не удивительно, что файл сохранен в формате JPEG. В заключение щелкаем на кнопке ОК. Изображение будет вставлено на место текстового курсора на веб-страницу. Если файл, содержащий изображение, предварительно не был помещен в корневую папку сайта, Dreamweaver MX 2004 предупредит вас об этом, а заодно поинтересуется, не надо ли переместить этот файл в папку сайта. Конечно, надо. Иначе на опубликованной веб-странице этот рисунок может не отобразиться.

Заменитель графики

Бывает так, что ко времени верстки веб-страницы подходящее графическое изображение еще не готово. А вставлять на веб-страницу неформатированное, неоптимизированное изображение не хочется. Для этих целей Dreamweaver MX 2004 припас так называемый заменитель графики.

Заменитель графики — это, по сути, заглушка для временного заполнения места на веб-странице, пока не будет найдена (или подготовлена) подходящая картинка. Для установки заменителя графики выполним команду меню Insert • Image Objects • Image Placeholder (Вставка • Объекты изображения • Поле изображения). Кроме того, можно на панели инструментов Insert (Вставка) открыть вкладку Common (Общее), на которой найти кнопку Image (Изображение). В раскрывшемся списке выбираем пункт Image Placeholder (Поле изображения). В любом случае откроется окно Image Placeholder (Поле изображения), показанное на рисунке.



В диалоговом окне Image Placeholder (Заменитель изображения) имеется всего четыре поля ввода. В самое верхнее поле Name (Имя) вводим лаконичный текст, который будет отображаться внутри заменителя графики. Так как размеры заменителя графики небольшие, то в поле Name (Имя) следует вводить одно, максимум два слова. Причем имя должно быть указано только латинскими буквами и не начинаться с цифры. Впрочем, это поле можно вообще оставить пустым. Например рисунок или фотография еще не готовы или не совсем готова, заменитель графики используется для того чтобы заменить данный рисунок. Поэтому и в поле ввода Name (Имя) вводим: name. В следующую ниже строку вводим размеры заменителя изображения. Эти размеры по умолчанию указаны: 32 x 32 пиксела. Согласимся с ними (если точно не уверены, какие размеры нам нужны). Позже, когда заменитель графики будет вставлен на веб-страницу, эти размеры можно будет подобрать опытным путем в соответствующих полях на панели Properties (Свойства). Параметр Color (Цвет) задает цвет заливки внутренней области заменителя изображения. По умолчанию установлен серый цвет. Таковым его можно и оставить. В поле ввода Alternate text (Альтернативный текст) вводим альтернативное описание, которое будет передано и вставлено на место заменителя изображения.

Это весьма полезный параметр. Возможно, что пользователь в своем браузере отключил отображение графики или графика загружается слишком медленно. В этом случае на месте графики и отобразится альтернативный текст. Данное поле можно заполнять и русскими буквами. Все, теперь щелкаем на кнопке ОК. Dreamweaver MX 2004 вставит заменитель графики на место текстового курсора. В результате

заменитель графики будет представлять собой прямоугольник заданных размеров и цвета, с помещенными внутрь именем и размером. Все дальнейшие операции, как с изображением, так и с его заменителем, будут схожими. В основном они выполняются на панели Properties (Свойства).

Заменитель графики — вспомогательный инструмент. Он будет виден только в Dreamweaver MX 2004.

Графические ссылки

В принципе, создание ссылки из графического изображения мало чем отличается от создания обычной, например текстовой, ссылки. Для создания такой ссылки нужно иметь как минимум два объекта. Во-первых, заранее создаем и сохраняем веб-страницу, на которую будет ссылка. Во-вторых, подбираем подходящее изображение, которое собираемся использовать в качестве графической ссылки. Предварительно это изображение следует отформатировать, подобрать нужные размеры и сохранить его с нужным именем, в нужном формате и в соответствующей папке сайта. К подборке изображения следует относиться очень серьезно.

Как правило, изображение, используемое в качестве ссылки, имеет маленькие размеры. Скажем, мы планируем показывать на веб-странице галерею этикеток одного знаменитого петербургского пивоваренного завода. Чтобы подробно представить информацию о продукции этого пивзавода, требуется представить довольно много этикеток. Причем все эти этикетки должны быть отсканированы с большим разрешением и иметь достаточно большие размеры. При таких условиях веб-страница будет загружаться очень долго (учитывая характеристики наших линий связи и модемов посетителей веб-страницы). Все это негативно действует на популярность нашей веб-страницы. Но есть элементарный выход! Мы не станем размещать на нашей веб-странице сразу большие и качественные изображения. Тем более что все они могут просто не поместиться на данной веб-странице. Лучше мы подготовим уменьшенные копии (возможно даже и с худшим разрешением) изображений этикеток и разместим их последовательно на нашей веб-странице. Для этого используем команду меню Insert • Image (Вставка • Изображение). В диалоговом окне Select Image Source выбираем имя файла, содержащего соответствующую миниатюру этикетки. Так же поступаем и с другими миниатюрами этикеток. Под ними разместим предложение щелкнуть мышью по той или иной картинке для получения более подробной информации на соответствующем рисунке .

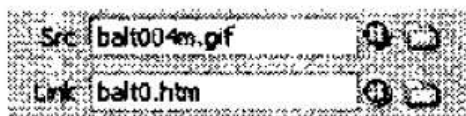
Этикетки из моей коллекции



Щелкни на картинке для просмотра

Фрагмент веб-страницы, на которой размещены миниатюрные изображения-ссылки

Сохраним созданную веб-страницу с миниатюрами этикеток. Теперь создаем отдельные веб-страницы с изложением подробной информации о каждом сорте пива, этикетка которого представлена на начальной странице. Здесь же приведем и крупное изображение такой же этикетки с лучшим разрешением. Сохраняем все созданные веб-страницы. А потом вновь переходим к начальной веб-странице с миниатюрами этикеток. Выделяем первое маленькое изображение. Переходим на панель Properties (Свойства). Здесь в поле ввода Src будет отображаться адрес файла, содержащего это изображение (или имя файла с изображением — если он расположен в корневой папке сайта). А ниже него располагается поле ввода Link (Ссылка), в которое надо ввести адрес (или имя) файла, содержащего веб-страницу, открывающуюся по ссылке.



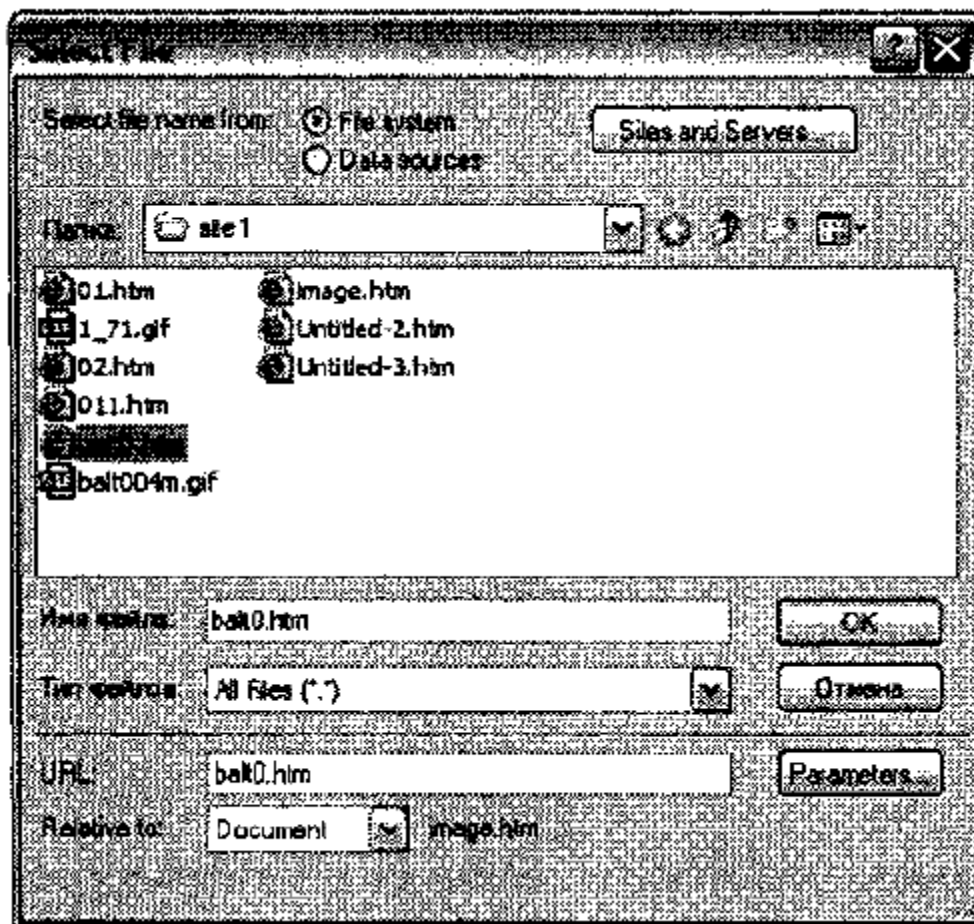
Впрочем, если щелкнуть по пиктограмме папки справа от поля ввода Link (Ссылка), то откроется диалоговое окно Select File (Выбор файла), в котором можно выбрать файл, содержащий веб-страницу с подробным описанием сорта пива и увеличенной картинкой. Этот способ подходит в том случае, если мы точно не помним имя нужного файла.

Вот и все, цель достигнута. Рисунок после прикрепления к нему функций ссылки никак не изменится. Но при наведении на него указателя мыши последний изменит свой вид на «указательный палец». Это, как мы помним, означает, что в этом месте расположена ссылка.

Стоит заметить, что если в свойствах изображения параметр Border (Граница) имеет не равное нулю значение, то рамка, рисуемая вокруг изображения-ссылки, будет иметь цвет, установленный в общих свойствах веб-страницы для гиперссылок. Однако граница рисунка-ссылки используется очень редко.

Активная картинка

При наведении указателя мыши на изображение-ссылку само изображение не изменяется. Указатель же мыши меняет свой вид на пиктограмму «указательный палец». Это очень хорошо, но не слишком примечательно. Особенно плохо заметен этот эффект при создании маленьких ссылок или ссылок-кнопок. Часто дизайнерами используется принцип, когда при наведении указателя мыши меняется вид самого объекта-ссылки. Такой объект называется активным, а ссылка — активной ссылкой. Для создания такой ссылки, конечно, можно использовать соответствующие скрипты (сценарии). Но Dreamweaver MX 2004 позволяет сделать то же самое гораздо быстрее и проще.

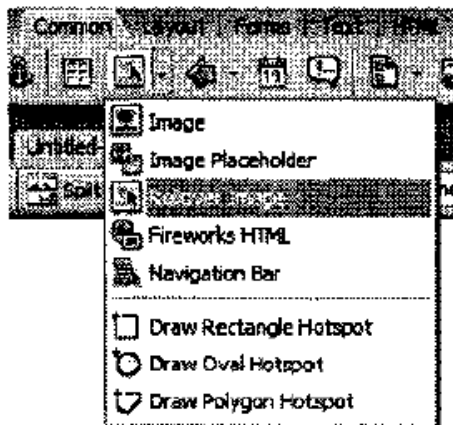


Для этого нам потребуется два похожих изображения. Первое — исходное, второе — появляющееся на месте исходного после наведения курсора (так называемое перекрывающее изображение). Как правило, второе изображение отличается от первого только оттенком либо несущественными изменениями оформления. Мы будем создавать кнопки меню сайта. В графическом редакторе (например, в редакторе Adobe Photoshop) создаем изображение первой кнопки. Это будет небольшой прямоугольник, имеющий эффект выпуклости. На прямоугольнике-кнопке пишем слово **НОВОСТИ** — это будет кнопка для перехода на страницу новостей сайта. Сохраняем созданную кнопку с именем `knorkal.gif`. Приступаем к созданию второй части кнопки — она будет заменять первую при наведении указателя мыши. Для второй кнопки используем прямоугольник с аналогичными свойствами, только увеличим размер текста и добавим затемнение по краям прямоугольника. Этот рисунок сохраним в файле `knorka2.gif`. Предварительная подготовка изображений закончена. Таким образом, мы имеем два изображения одной кнопки — до и после наведения указателя мыши.



Теперь переходим к Dreamweaver MX 2004. Открываем документ, содержащий нужную веб-страницу. Здесь выбираем место для вставки кнопки. Выполняем команду меню **Insert • Image Objects • Rollover Image** (Вставка • Графический объект • Изменяющийся рисунок). Впрочем, можно на панели **Insert** (Вставка) открыть вкладку **Common** (Общее), на которой, в свою очередь, щелкнуть на уже знакомой нам кнопке

Image (Рисунок). В раскрывающемся списке следует выбрать пункт Rollover Image (Изменяющийся рисунок).



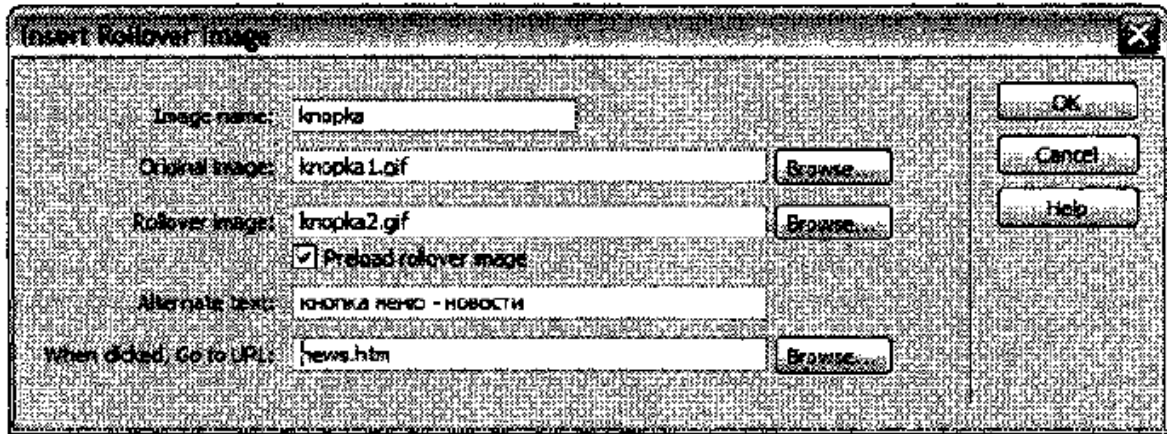
В результате описанных выше действий откроется диалоговое окно Insert Rollover Image (Вставка изменяющегося рисунка). Именно в этом окне мы и будем создавать нашу кнопку (активное изображение).

В самое верхнее поле ввода Image name (Имя рисунка) вводим уникальное имя нашей кнопки — кнопка. По умолчанию Dreamweaver MX 2004 сам вставляет имя в это поле (Image 1). Заметьте, это имя может содержать только латинские буквы и начинаться должно только с буквы. Ниже следуют два поля ввода, в которых указываем имя первого (Original image) и второго (Rollover image) изображения нашей кнопки или путь к ним. С помощью кнопок Browse (Просмотр) можно найти нужные рисунки, а не вводить их имена вручную.

В строке Preload rollover image (Предварительная загрузка изменяющегося рисунка) по умолчанию включен флажок. Лучше так его и оставить. Дело в том, что этот флажок вставляет в HTML-код веб-страницы специальную программу-сценарий. При загрузке веб-страницы в браузер эта программа-сценарий выполнит предварительную подзагрузку второго изображения. Поэтому при наведении на кнопку указателя мыши нам не придется ждать изменения картинки — оно выполнится сразу же.

В поле ввода Alternate text (Альтернативный текст) вводим так называемый текст-подсказку. При наведении указателя мыши на изображение этот текст появится в возникшем поле желтого цвета, под указателем. В данное поле текст можно вводить и русскими буквами. Позднее текст-подсказку можно будет изменить в поле Alt панели Properties (Свойства) изображения.

В последнее поле ввода When clicked, Go to URL (По щелчку перейти по адресу) вводим URL-адрес веб-страницы, на которую надо переходить после щелчка на активной ссылке.



Закончив ввод всех значений, нажимаем на кнопке ОК. Dreamweaver MX 2004 самостоятельно создаст активную картинку-кнопку и разместит ее на веб-странице. Вот и все. Проверить действие кнопки можно будет после загрузки веб-страницы в браузер.

3. Вопросы для закрепления.

- 1) Какими способами можно добавить графические изображения в web- страницу?
- 2) Что такое заменитель графики, назовите его назначение?.

Лекция 13

Тема: Работа с таблицами

Цели: *Ознакомить учащихся с основными способами создания и редактирования таблиц и дополнительных элементов для разметки документа*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Элементы разметки страницы
 - 2) Таблицы
 - 3) Табличный дизайн страницы
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Разметка веб-страницы

Чтобы все элементы на веб-странице смотрелись эстетично, пространство веб-страницы использовалось максимально, но при этом посетителю на ней было комфортно, чтобы вся веб-страница была красива, надо поработать над правильной и оптимальной разметкой веб-страницы. Ясно, что перед началом работы в Dreamweaver MX 2004 необходимо создать план-макет веб-страницы — что называется, «на бумаге».

Вспомогательные инструменты разметки веб-страницы

В Dreamweaver MX 2004 заложено много средств для того, чтобы сделать вашу работу по созданию красивого и оптимального сайта быстрой и эффективной. Такими вспомогательными инструментами являются среди прочих так называемая модульная сетка и линейка. Основное их назначение — это выравнивание различных объектов на веб-странице.

Сетка

Чтобы включить модульную сетку в Dreamweaver MX 2004, надо выполнить команду меню View • Grid • Show Grid (Вид • Сетка • Показать сетку). Повтор этой же команды приведет к скрытию сетки. В результате на рабочей области веб-страницы сетка. Эта сетка будет видна только в режиме редактирования Dreamweaver MX 2004. По умолчанию линии сетки расположены с интервалом в 50 пикселей. Чтобы изменить интервал следования линий сетки, выполняем команду меню View • Grid • Grid Setting (Вид • Сетка • Настройка сетки). Эта команда открывает диалоговое окно Grid Setting (Настройка сетки). Флажок **Show grid** (Показать сетку) показывает сетку, а флажок **Snap to grid** (Привязать к сетке) включает привязку элементов веб-страницы к сетке. Таким образом, действия этих флажков аналогичны соответствующим командам меню. В поле ввода Spacing (Расстояние) устанавливаем значения промежутков между соседними линиями сетки. Здесь же указываем и единицу измерения этих расстояний. Последний переключатель **Display** (Отображение)

включает один из двух возможных режимов отображения сетки на экране: линиями (Lines) или точками (Dots).

Командой меню View • Grid • Snap to Grid (Вид • Сетка • Привязать к сетке) включается так называемая привязка к сетке. В этом случае все помещаемые на веб-страницу объекты будут привязываться («прилипать») к той или иной линии сетки. Это несколько упрощает взаимное размещение и выравнивание объектов.

Линейки

Вспомогательный инструмент линейки (Rulers) присутствует практически везде и работает во всех приложениях одинаково. Основное назначение таких линеек — быстрое и точное определение текущих координат того или иного объекта на веб-странице. Как и сетка, линейки существенно упрощают и ускоряют работу по созданию четко упорядоченной, а если нужно, то и симметричной веб-страницы.

Для включения режима отображения линеек выполняем команду меню View • Rulers • Show (Вид • Линейки • Показать). В этом же меню можно выбрать и единицы измерения на линейках — пиксели (Pixels), дюймы (Inches) или сантиметры (Centimeters).

В результате выполнения этой команды меню у верхней и левой границ рабочей области редактора появятся вспомогательные линейки. На линейках имеются бегунки, которые точно отслеживают движения указателя мыши в пределах рабочей области редактора. Как и сетка, линейки отображаются исключительно в режиме редактирования.

Начальную точку обеих линеек можно совместить (или перенести на новое место). Для этого щелкаем на маленьком белом квадратике у пересечения линеек и, не отпуская левой кнопки мыши, перетаскиваем его в нужное место. А чтобы вернуть начальную точку в первоначальное место, выполним команду меню View • Rulers • Reset Origin (Вид • Линейки • Вернуть в исходное). После этого начала отсчета обеих линеек вернуться в исходное состояние — верхний левый угол.

Дополнительная вспомогательная разметка

Кроме вспомогательных сетки и линеек есть еще несколько видов иной вспомогательной разметки. Вспомогательные инструменты существенно облегчают работу по созданию веб-страницы и доступны только в Dreamweaver MX 2004. Они абсолютно не влияют на внешний вид веб-страницы и ее HTML-код.

Для того чтобы включить (или выключить) какую-либо вспомогательную разметку, необходимо выполнить команду меню View • Visual Aids (Вид • Вспомогательная разметка). По этой команде меню раскроется список возможных вариантов вспомогательной разметки. Уже включенные разметки помечены галочкой. Пункт Show All (Показать все) автоматически включает все представленные здесь варианты вспомогательной разметки. При этом сам он изменяется на пункт Hide All (Спрятать все). Щелчок по такому пункту приводит к автоматическому выключению сразу всех вспомогательных разметок.

Предусмотрены следующие вспомогательные разметки.

- Table Widths включает отображение размеров ячеек таблицы.
- Table Border отвечает за отображение границ таблицы.

Таблицы на веб-странице, строящейся в Dreamweaver MX 2004, вместе со слоями являются наиболее популярными средствами вспомогательной разметки.

- Layer Borders включает отображение границ слоев.

- Frame Borders отображает границы фреймов.
- Image Maps показывает разметку графических карт.
- Invisible Elements отображает знаки невидимых элементов страницы: комментарии, именные якоря, сценарии Java Script и т. д.

Все эти специальные элементы так или иначе облегчают работу по построению стильного, хорошо отформатированного и правильно размеченного сайта

Таблицы

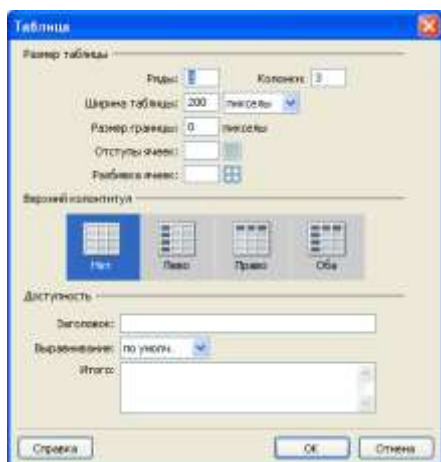
Таблицы в HTML уже давно не просто оболочка для специального представления и хранения данных.. Таблицы — еще и очень простой и эффективный способ форматирования целой веб-страницы. Можно создавать таблицы различной сложности, редактировать их свойства.

Итак, все таблицы могут быть разделены на два больших класса Во-первых, это обычные таблицы для удобного систематизированного представления каких-либо данных. Во-вторых, это таблицы для простого и удобного форматирования содержимого веб-страницы.

Создание таблиц

Выполним команду меню Insert • Table (Вставить • Таблицу). Кроме того, можно нажать и на кнопку Table (Таблица) вкладки Common (Общее) на панели инструментов Insert (Вставка). Все это приводит к открытию диалогового окна Table (Таблица). В этом диалоговом окне мы будем задавать начальные параметры нашей таблицы. Группа параметров Table size (Размеры таблицы) является основной. В полях ввода Rows (Строки) и Columns (Колонки) задаем соответственно количество строк и столбцов (колонок) будущей таблицы. Это начальные данные. При заполнении таблицы на веб-странице можно будет изменить эти значения — добавить или удалить столбцы или строки.

В поле ввода Table width (Ширина таблицы) задаем ширину таблицы. Здесь же в раскрывающемся списке можно выбрать одну из двух систем измерения этой ширины — проценты (percent) или пиксели (pixels). Затем в поле ввода Border thickness (Толщина границы) задаем толщину границ таблицы в пикселах. По умолчанию установлено единичное значение — оно и является наиболее подходящим. Попробуйте увеличить его — ячейки таблицы станут больше напоминать кнопки или рамки. Для того чтобы вообще убрать границы, надо ввести в это поле 0 (ноль). Этот способ применяется при использовании таблицы в качестве вспомогательной разметки.



В поле ввода Cell padding задаем расстояние между содержимым и границами отдельной ячейки. А в поле ввода Cell spacing задаем расстояние между отдельными

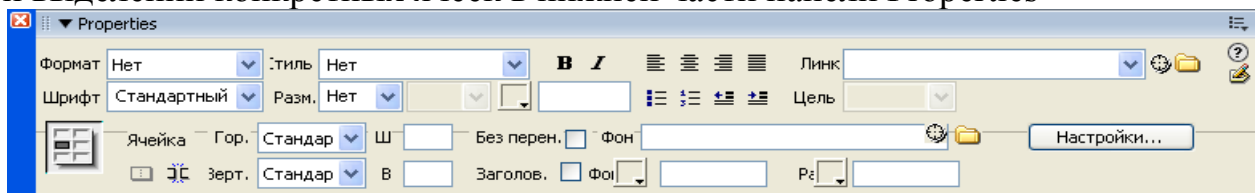
ячейками таблицы. Оба этих параметра задаются в пикселах. В следующей ниже группе Header (Заголовок) выбираем один из четырех возможных вариантов задания заголовка таблицы:

- None — нет заголовка;
- Top — в виде первой строки;
- Left — в виде первого столбца;
- Both — в объединенном виде (первый столбец и первая строка).

При задании такого заголовка таблицы Dreamweaver MX 2004 будет самостоятельно выравнивать по центру и выделять полужирным шрифтом введенные в ячейки данные. В следующей группе параметров таблицы полезными являются только два первых. Так, в поле ввода Caption (Заголовок) вводим текст, который на веб-странице будет отображаться как заголовок, над таблицей. В раскрывающемся списке Align caption (Выравнивание заголовка) выбираем один из пяти возможных вариантов выравнивания заголовка. В последнем, самом широком поле ввода Summary (Резюме) можно было бы указать подводный итог текста. Но пока ни один браузер этого поля не понимает и не отображает. После того как все параметры указаны, нажимаем кнопку О К и получаем соответствующую таблицу на веб-странице

Изменение размеров ячеек

Размеры как всей таблицы, так и отдельных ее ячеек устанавливаются в одних и тех же полях W (ширина) и H (высота) панели Properties. Параметры для таблицы будут отображаться при выделении всей таблицы в верхней части панели, а для ячеек — при выделении конкретных ячеек в нижней части панели Properties

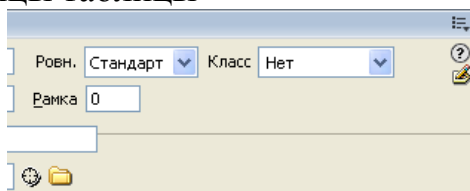


Самый простой и верный способ изменения размеров ячеек таблицы состоит в ручном растягивании границ таблицы. В Dreamweaver MX 2004 предусмотрен специальный режим отображения границ таблицы. Он включается командой меню View • Table Mode • Expanded Tables Mode (Вид • Табличный режим • Расширенный табличный режим) или аналогичной командой контекстного меню таблицы. Команда меню View • Visual Aids • Table Widths (Вид • Вспомогательная разметка • Ширина таблицы) включает отображение текущих размеров как всей таблицы, так и отдельных столбцов. В результате применения этой команды под таблицей светло-зеленым цветом будут постоянно отображаться текущие размеры (ширина) таблицы.

Размеры ячеек таблицы могут измеряться в пикселах или процентах — кому как удобнее. В любой момент можно изменить эти единицы измерения. Размеры в процентах рассчитываются от имеющегося свободного места.

Граница таблицы

Толщина прямоугольной рамки таблицы задается значением на панели Properties в поле ввода Border (Граница или Рамка), данное значение задает толщину внешней границы таблицы



Форматирование таблицы

Считается хорошим тоном, если вся таблица создана в одном стиле. Но при ручном задании размеров, цвета границ и фона ячеек, свойств текста и т. д. легко сбиться и тем самым испортить всю таблицу. Dreamweaver MX 2004 на этот случай приберет очень полезный инструмент, называемый быстрым форматированием таблиц.

На новой веб-странице мы создаем обычную таблицу командой меню Insert • Table. В диалоговом окне Table задаем только необходимое количество строки столбцов будущей таблицы и вставляем таблицу на веб-страницу.

Не снимая выделения с таблицы, выполняем команду меню Command • Format Table (Команда • Форматировать таблицу). Эта команда меню открывает диалоговое окно Format Table (Формат таблицы), представленное. Именно здесь мы и задаем общий формат для всей таблицы.



Также Dreamweaver позволяет сортировать данные, экспортировать импортировать табличные данные

Табличный дизайн веб-страницы

Один из распространенных способов применения таблиц в качестве основы структуры веб-страницы — это так называемый табличный дизайн. При табличном дизайне в основе веб-страницы находится сложная таблица, а вся информация помещается в соответствующие ячейки. Такой способ позволяет разместить на одной веб-странице больше информации, систематизировать ее и сделать удобной для пользователей. Обычно именно по этому принципу строятся веб-страницы поисковых или информационных сайтов. Рассмотрим на примере создание веб-страницы с такой табличной структурой. Для начала создадим пустую веб-страницу с помощью команды меню File • New (Файл • Новый). Затем вставим на эту веб-страницу простую таблицу. Выполним команду меню Insert • Table. В диалоговом окне Table зададим параметры нашей таблицы. Стоит вновь напомнить, что предварительно следует составить макет будущей веб-страницы на бумаге. Итак, в нашей таблице будет точно 3 колонки (Columns) и порядка 5 строк (Rows). Если с количеством строк точно определиться не получилось, не беда — в случае чего мы в любой момент можем добавить или удалить необходимые строки. После того как табличная заготовка дизайна веб-страницы готова, можно начинать заполнять ее начинкой. В ходе этой работы структурную таблицу можно будет подогнать по размерам изображения или наоборот.

При использовании табличного дизайна веб-страницы зачастую приходится применять вложенные таблицы. В этом случае в одну из ячеек таблицы можно вставить другую таблицу. Однако не стоит слишком увлекаться — это затрудняет дальнейшую работу.

Описанный выше табличный дизайн веб-страницы считается довольно сложным применением таблиц. Есть способы и попроще — например, использование таблицы в качестве рамки для текста или изображения.

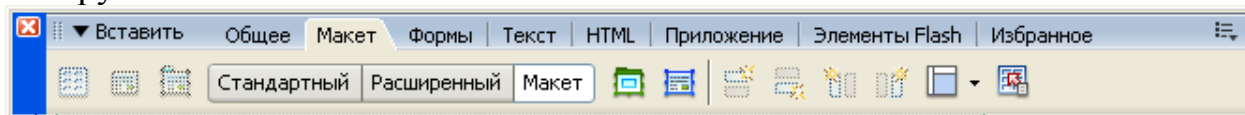
Табличные рамки

Тот факт, что в ячейку таблицы можно поместить что угодно, широко используется веб-мастерами. Например, с помощью обычной таблицы можно создать интересную рамку для текста или изображения. Для этого на веб-странице командой меню Insert • Table создаем простую таблицу, которая будет содержать всего одну ячейку. Все делается точно так же, как и при создании обычной таблицы. Только количество колонок (Columns) и строк (Rows) задаем равными единице. При указании параметров в диалоговом окне Table особое внимание следует уделить размеру границ таблицы (Border). Для нашего примера это значение мы установили равным 15 пикселей. После того как одноячеечная таблица готова, можно приступить к вводу в нее текста.

Использование таблиц разметки

Мы уже касались вопросов использования таблиц в интересах дизайна (или структурирования) веб-страницы. Однако Dreamweaver MX 2004 предусмотрел еще интересную возможность использования таблиц для удобства макетирования веб-страницы.

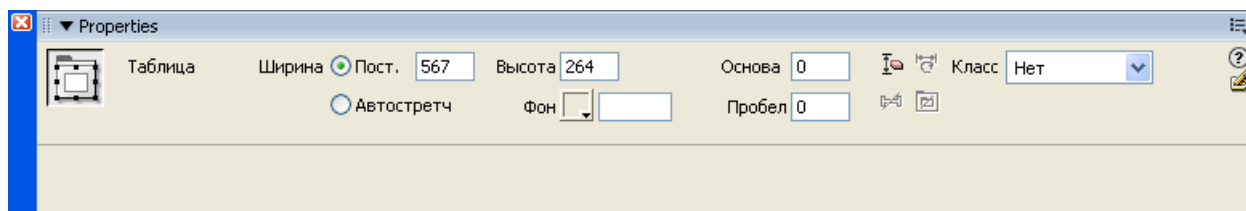
Для переключения в режим табличной разметки выполним команду меню View • Table Mode • Layout Mode (Вид • Табличный режим • Режим разметки) или нажмем кнопку Layout (Разметка), которая находится на вкладке Layout (Макет) панели инструментов



Здесь же имеется еще две кнопки. Первая кнопка — **Standard** (Стандартный) — выключает режим табличной разметки и переводит веб-страницу в обычный режим редактирования. Кнопка **Extended** (Растянутый) осуществляет переход в режим отображения растянутых (увеличенных) таблиц. Для переключения в режим табличной разметки выполним команду меню View • Table Mode • Layout Mode (Вид • Режим таблицы • Режим макета) или нажмем кнопку Layout (Разметка), которая находится на вкладке Layout панели инструментов. Здесь же имеется еще две кнопки. Первая кнопка — **Standard** (Стандартный) — выключает режим табличной разметки и переводит веб-страницу в обычный режим редактирования. Кнопка **Extended** (Растянутый) осуществляет переход в режим отображения растянутых (увеличенных) таблиц.

Параметры таблицы разметки

Ранее были рассмотрены свойства отдельных ячеек таблицы разметки. Однако можно задать и свойства самой таблицы разметки. Панель Properties, определяющая свойства таблицы разметки, немного отличается от панели Properties для обычных таблиц



Ширина самой таблицы разметки, как и ширина ячейки, может быть фиксированной (Fixed) или растягивающейся (Autostretch). Выбор осуществляется переключателем Width. В общем, все так же, как и при работе с ячейками. Но стоит заметить, что таблицы разметки с фиксированной (Fixed) шириной не могут содержать растягивающиеся ячейки. Стандартный селектор цвета Bg задает цвет фона таблицы разметки.

Поле ввода CellPad задает расстояние между границей ячейки и ее содержимым.

Поле ввода CellSpace задает расстояние между границами соседних ячеек. По умолчанию оба значения равны нулю.

Чуть правее этих полей ввода на панели Properties располагаются четыре кнопки, позволяющие выполнять типовые операции над ячейками таблицы.

- Кнопка Clear Row Heights (Очистить высоту строки) позволяет обнулить заданные значения высоты ячеек таблицы разметки. После нажатия этой кнопки каждая ячейка будет иметь такую высоту, чтобы только вместить все свое содержимое.

- Кнопка Make Cell Widths Consistent (Задать ширину ячейки по содержимому) позволяет автоматически задать значения ширины ячеек таблицы, чтобы они полностью соответствовали ширине содержимого. Эта кнопка применима в тех случаях, когда ширина ячейки таблицы разметки задана фиксировано, а ее содержимое превышает эти размеры. Нажатие на данную кнопку позволяет «подогнать» ширину ячейки под ширину содержимого.

- Кнопка Remove All Spacer Images (Удалить все изображения-распорки) приводит к удалению всех принятых изображений-распорок для всех ячеек таблицы разметки.

- Кнопка Remove Nesting (Удалить вложенность) приводит к удалению оболочки вложенной таблицы. В этом случае все ячейки этой вложенной таблицы становятся ячейками «родительской» таблицы. При этом содержимое ячеек не теряется.

3. Вопросы для закрепления.

- 1) Что такое «линейка»?
- 2) Что такое «сетка»?
- 3) Назовите другие способы разметки страницы
- 4) Назовите способы создания и редактирования таблиц в редакторе DW.

4. Домашнее задание: Подготовить две страницы с различными типами разметки страницы в редакторе DW.

Лекция 14

Тема: Создание фреймов

Цели: *Ознакомить учащихся с основными фреймовой структуры; дать понятие фрейма, его назначение и способы создания*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Задание фреймовой структуры веб-страницы
 - 2) Выбор и редактирование фрейма
 - 3) Заполнение фреймов
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

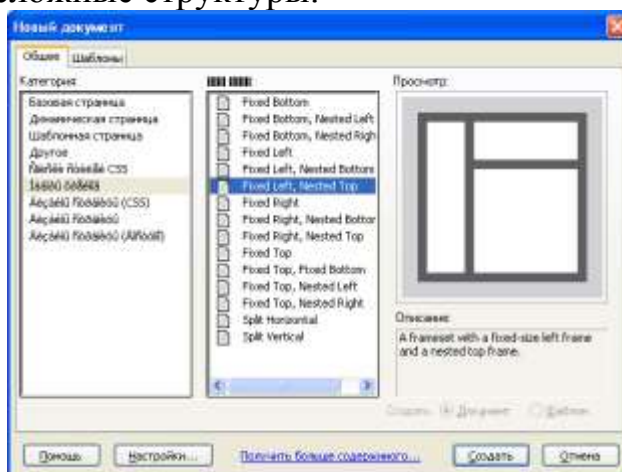
Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Задание фреймовой структуры веб-страницы

Фреймовая структура сайта должна быть задана в самом начале его проектирования. После выполнения команды меню File> New открывается диалоговое окно New Document. В этом окне переходим на вкладку General и в ней выбираем пункт Framesets (Набор фреймов). В появившемся чуть правее окне Framesets (Набор фреймов) подбираем подходящий шаблон фреймовой структуры веб-страницы (набор фреймов). Символический вид выделенной структуры будет отображаться в области Preview (Предварительный просмотр) диалогового окна New Document.

Здесь представлены лишь пятнадцать наиболее распространенных наборов фреймов. Можно, конечно, придумать и другие варианты, но опыт показывает, что не стоит создавать слишком сложные структуры.



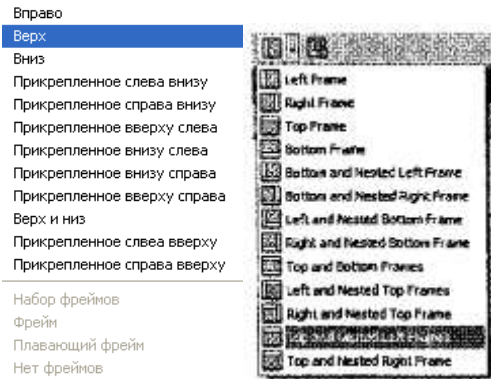
Нажимаем кнопку Create (Создать) и получаем соответствующую заготовку сайта. Здесь мы описали задание набора фреймов сайта на тот момент, пока у нас еще нет самой веб-страницы. Но Dreamweaver MX 2004 дает возможность задать набор фреймов и тогда, когда основная веб-страница уже создана. В этом случае мы переходим на вкладку Layout панели инструментов, где нажимаем кнопку Frames (Фреймы). При нажатии на эту кнопку откроется целый список возможных наборов

фреймов полностью повторяющий описанный выше список, открывающийся в диалоговом окне New Document .

В этом окне следует выбрать один из возможных наборов фреймов сайта. Миниатюры пиктограмм отображаются перед названием соответствующей структуры.

Ниже приведены краткие описания наборов фреймов, вызываемых с помощью кнопки Frames вкладки Layout панели инструментов.

Естественно, предложенные варианты не являются исчерпывающими. Здесь приведены лишь основные из них, наиболее часто используемые. На их базе можно строить свои новые наборы фреймов, в том числе и как комбинации перечисленных.



Выбор и редактирование фрейма

Для того чтобы задать свойства определенного фрейма или заполнить его содержимым, предварительно фрейм следует выделить (сделать текущим). Сделать это в Dreamweaver MX 2004 можно (как, впрочем, и все остальное) несколькими способами. Так, для выделения фрейма можно, удерживая нажатой клавишу Alt, просто щелкнуть левой кнопкой мыши на этом фрейме. Но удобнее пользоваться панелью **Frames** (Фреймы). Для ее открытия выполним команду меню **Window • Frames** (Окно • Фреймы). Панель **Frames** появится в колонке панелей инструментов у правой границы рабочей области Dreamweaver MX 2004. На этой панели схематично будут отображаться все имеющиеся в данном наборе фреймы. Выбранный фрейм будет помечен черной рамкой. Наборы фреймов имеют общую толстую светло-серую рамку.



Для выбора нужного фрейма из текущего набора фреймов на панели **Frames** достаточно просто щелкнуть левой кнопкой мыши по изображению нужного фрейма. Щелчок по границам фрейма приводит к выделению сразу целого набора фреймов.

Редактирование свойств набора фреймов

Если необходимый фрейм выделен, на панели Properties отображаются его свойства. В центральной части панели Properties размещаются поля ввода основных общих свойств данного набора фреймов. В раскрывающемся списке Borders (Границы) задаем наличие или отсутствие границ у набора фреймов. Здесь доступны всего три варианта:

- Yes (Да) — границы есть;
- No (Нет) — границ нет;
- Default (По умолчанию) — установлено по умолчанию.

Ниже расположено поле ввода Border width (Ширина границы), в котором задаем толщину границы (если она есть) в пикселах. Если граница есть, но толщина ее имеет нулевое значение, то граница не будет видна. Также рекомендуется задавать нулевое значение и в случае отсутствия границы. Еще правее находится поле ввода Border color (Цвет границы) и раскрывающаяся палитра, которые служат для задания цвета границы набора фреймов. По умолчанию обычно используется светло-серый цвет границ. В нижней части панели Properties расположены поля ввода высоты (или ширины) набора фреймов, а так же единицы ее измерения. В поле ввода Value (Значение) как раз и вводится само значение высоты (или ширины) набора фреймов. Единицы измерения указываются в раскрывающемся списке Units (Единицы). Так, высота (или ширина) набора фреймов может указываться:

- в пикселах (Pixels);
- процентах (Percent);
- без указания конкретного размера, в этом случае набор фреймов будет занимать все оставшееся от других наборов фреймов место (Relative). При выборе последнего пункта высота (или ширина) набора фреймов, указанная в поле ввода Value, уже не имеет значения. К аналогичному результату приводит и ввод в поле Value символа * («звездочка»). В правой части панели Properties размещено схематическое изображение наборов фреймов.

Редактирование свойств фрейма

Панель Properties, определяет основные свойства набора фреймов в целом. Кроме того, можно задавать и свойства конкретного фрейма из этого набора. При выделении какого-то фрейма из текущего набора фреймов на панели Properties отражаются его свойства

Заполнение фреймов

Заполнить фреймы необходимой информацией. Как уже говорилось, в каждый фрейм загружается отдельная веб-страница. Поэтому предварительно надо создать эти самые веб-страницы для загрузки в различные фреймы. Если такие веб-страницы уже созданы, то после создания наборов фреймов останется указать, в какой фрейм какую веб-страницу загружать. Эту информацию указываем в поле ввода Src на панели Properties отдельного фрейма. Однако если заготовок веб-страниц не было создано до построения набора фреймов, то эти самые веб-страницы можно создать и теперь.

3. Вопросы для закрепления.

- 1) Какие типы фреймов вы знаете?
- 2) Назовите назначение фреймов
- 3) Назовите основные свойства фреймов
- 4) Каким образом сохраняются наборы фреймов

4. Домашнее задание: Подготовить два набора фреймов в редакторе DW.

Лекция 15

Тема: Создание и использование форм

Цели: *Ознакомить учащихся с основными понятиями элементами форм. Способы их размещения и управления ими.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Назначение форм
 - 2) Создание новой формы
 - 3) Элементы формы
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

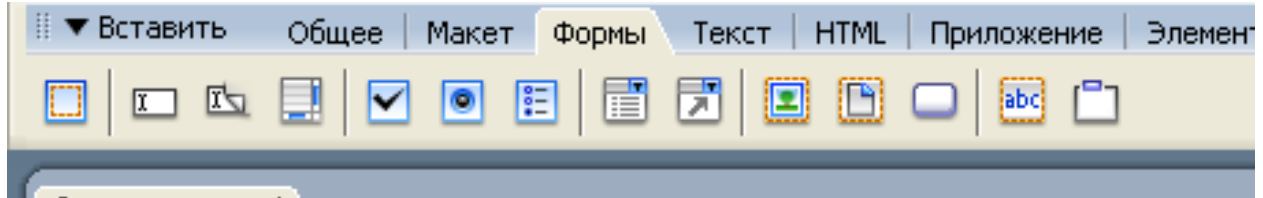
2. Изложение нового материала.

Последним базовым элементом дизайна веб-страниц являются формы (или вебформы). Фактически формы играют не столько роль украшения веб-страницы, сколько серьезного посредника для обратной связи между веб-сайтом и посетителем. Формы содержат множество самых различных элементов, которые помогут даже самому изощренному дизайнеру. Здесь имеются разнообразные кнопки, переключатели, поля ввода, раскрывающиеся списки и т. д. — в общем, все элементы интерфейса, к которому все уже давно привыкли (их можно встретить в различных диалоговых окнах практически всех приложений Windows). Однако все это только внешние украшения. Расположив на своей веб-странице эти самые кнопки переключатели или поля, вы только создадите их образ (изображение). Но чтобы заставить их работать, придется написать так называемую серверную программу. Именно она и подскажет, что делать при нажатии на ту или иную кнопку после ввода информации в текстовое поле или при выборе определенного пункта меню, раскрывающегося списка. И здесь вам может существенно помочь Dreamweaver MX 2004. Вставляя на веб-страницу форму или отдельный ее элемент, Dreamweaver MX 2004 уже делает заготовку кода соответствующей серверной программы. О том, как создавать сами серверные программы, мы поговорим несколько позже. А пока — собственно о самих формах. Чаще всего формы на веб-страницах используются для создания обратной связи с посетителем. Наиболее популярными являются анкетные формы. В этом случае посетителю сайта предлагается ответить на некоторые вопросы. При нажатии на кнопку (например, «Отправить») все введенные им данные отправляются по электронной почте на ваш адрес. Затем эти данные можно использовать, скажем, для статистических подсчетов. Кроме того, очень часто встречаются так называемые формы голосования. В этом случае достаточно установить переключатель в одну из позиций, соответствующих предлагаемым в ответ на вопрос предложениям. Такие формы наиболее популярны на новостных или музыкальных (хит-парады) сайтах. Создаются также формы, с помощью которых, не покидая веб-страницы, можно отправить сообщение хозяину сайта по электронной

почте. Область применения форм практически не знает пределов, хватило бы фантазии!

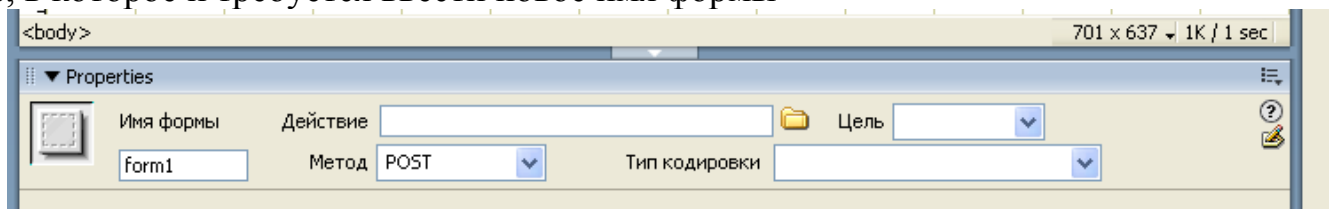
Создание новой формы

Dreamweaver MX 2004 предлагает на выбор несколько способов создания формы (или ее элементов). Какой из них вам больше понравится, такой и используйте. Наверное, самый простой способ — использовать вкладку Forms (Формы) панели инструментов Insert. Все элементы на этой панели разделены на шесть логических групп



Вкладка Forms панели инструментов Insert

В первой группе панели Forms расположен всего один элемент Form (Форма). Щелчок по нему левой кнопкой мыши приводит к созданию новой формы. Первоначально форма является пустой (не содержит никаких элементов). Также новую форму можно создать, выполнив команду меню Insert • Form • Form (Вставить • Форма • Форма). Кстати, в этом же меню расположены и другие команды, позволяющие вставлять различные элементы формы. При создании новой формы на веб-страницу будет помещена узкая полоска с красной пунктирной границей. Собственно, это и есть наша новая пустая форма. Одновременно с этим в HTML-код веб-страницы будет вставлен парный тег `<form>` с заготовкой некоторых параметров. В зависимости от поставленной цели новую форму можно разместить и внутри других, уже имеющихся, элементов веб-страницы, например внутри таблицы. Точно так же внутри формы можно разместить различные элементы веб-страницы, даже те, которые не являются элементами формы (таблицы, картинки, текст и т. д.). Единственное ограничение — в форму нельзя вставить другую форму. Новая форма сразу же станет активной (то есть готовой к дальнейшему редактированию). В связи с этим изменится и вид панели Properties. На ней можно настроить свойства самой формы. В поле ввода Form name (Имя формы) задается имя данной формы. По умолчанию Dreamweaver MX 2004 вставит туда стандартное имя form 1. Имя можно таким и оставить. Для изменения имени формы щелкните правой кнопкой мыши точно по пунктирной границе формы. Это приведет к открытию контекстного меню формы. Здесь следует выбрать первый пункт Name. Откроется небольшое диалоговое окно Change Attribute (Изменение атрибута). Оно содержит единственное поле ввода Name, в которое и требуется ввести новое имя формы



Панель Properties новой формы

Вернемся к панели Properties нашей новой формы. В верхней строке центральной части панели располагается поле ввода Action (Действие). Здесь требуется указать URL-адрес серверной программы, которая будет обрабатывать данные, введенные в эту форму. В раскрывающемся списке Target выбираем место, куда будет загружаться веб-страница, созданная серверной программой на основе

данных, введенных в форму. В раскрывающемся списке Method (Метод) выбираем метод пересылки введенных в форму данных. Здесь есть три возможных варианта: • Default — метод не определен. В этом случае работает метод, устанавливаемый по умолчанию (Get); • Get — установлен по умолчанию. В этом случае перед пересылаемыми данными ставится знак вопроса. Затем данные пересылаются на сервер, на котором находится программа обработки этих данных. Учтите, что все данные, пересылаемые таким методом, должны быть введены только латинскими буквами. А максимальный объем этих данных ограничивается максимальной длиной URL-адреса (8192 символа); • Post — в этом случае данные, введенные в форму, пересылаются в теле запроса, который происходит по указанному URL-адресу обработчика. В отличие от второго метода, при этом практически нет ограничений по объему или типу символов данных. В раскрывающемся списке Enctype выбираем метод кодирования данных, полученных формой. Здесь всего два варианта: • Application/x-www-form-urlencoded — используется для кодировки данных при методе пересылки Post; • Multipart/form-data — используется в случаях, если в форме имеется поле ввода File Field (Поле ввода), в которое посетитель может ввести имя файла. При особой необходимости здесь можно указать и свой собственный метод. Но это уже не для начинающих. После создания самой формы можно приступить к ее наполнению, так как пустая форма не имеет смысла.

Элементы формы

Как уже упоминалось, внутри формы можно размещать самые разные элементы. Единственное ограничение — внутри одной формы нельзя поместить другую форму. Однако без особого труда в форму можно поместить типичные элементы веб-страницы, а также специальные элементы управления. Вставка в форму типичных элементов веб-страницы Для того чтобы вставить в форму типичные элементы веб-страницы, не требуется придумывать что-либо новое. Устанавливаем текстовый курсор в нужное место формы. А затем выполняем привычные нам действия, порядок которых диктуется тем, какой элемент мы желаем вставить в форму. Действуйте так, как будто вы вставляете этот элемент не в форму, а на обычную веб-страницу. Так, если вы хотите вставить в форму обычный текст, то просто наберите его на клавиатуре. Для вставки графического изображения придется выполнить команду меню Insert • Image. Пользуясь стандартными инструментами, можно изменять свойства вставляемых элементов. Однако учтите, что размеры формы будут меняться в зависимости от размеров вставляемых в нее элементов. Если со вставкой типичных элементов веб-страницы все ясно, то перейдем к подробному рассмотрению вставки в форму различных элементов формы. Таких элементов всего тринадцать. Ничего сложного во вставке этих элементов нет. Их можно вставлять либо прямо в форму, либо помещать внутрь уже вставленных в форму типичных элементов веб-страницы (например, в таблицы). Все элементы формы расположены на вкладке Forms панели инструментов. Они разделены на пять логических групп. По какому принципу их рассортировали, догадаться не трудно. Для вставки в форму необходимого элемента достаточно просто щелкнуть по его значку на вкладке левой кнопкой мыши или перетащить его изображение со вкладки в нужное место формы. Кроме того, можно вставить нужный элемент, выполнив соответствующую команду меню Insert • Form. Для удаления ненужного элемента формы просто выделяем его и нажимаем клавишу Del. Таким же способом можно удалить и ненужную форму. Если вы попытаетесь вставить элемент формы вне границ самой формы, то Dreamweaver MX 2004 это сразу

заметит. Откроется окно предупреждения, в котором Dreamweaver MX 2004 спросит: Add form tag? (Добавить тег формы?). Есть два варианта ответа. Если вы нажмете кнопку Yes (Да), то перед тем как вставить элемент, Dreamweaver MX 2004 создаст новую форму и уже в нее поместит сам элемент. В противном случае элемент будет помещен непосредственно на веб-страницу. Теперь, когда общие правила работы с элементами формы ясны, рассмотрим все эти элементы несколько подробнее.

Вставка в форму текстового поля

Первым слева на вкладке Forms панели инструментов располагается группа элементов полей. В эту группу входит три таких элемента. Первый из них — TextField (Текстовое поле). Для вставки в форму текстового поля можно использовать и команду меню Insert • Form • Text Field (Вставить • Форма • Текстовое поле). Как известно, обычное поле ввода служит для ввода в него какого-либо текста. После вставки в форму этого элемента появится небольшой горизонтальный прямоугольник. Он сразу будет выделен пунктирной линией. В панель Properties этого элемента можно сразу внести изменения параметров. В поле ввода TextField (Текстовое поле) вводится имя данного текстового поля. Поле ввода Char width (Ширина по символам) служит для ввода значения ширины поля ввода, то есть определяет, сколько введенных символов можно будет увидеть в этом поле. Не следует задавать слишком большое значение. Для этого существуют другие элементы формы. Поле ввода Max Chars (Максимальное число символов) задает максимальное число символов, которое можно ввести в данное поле ввода. В случае если значение Max Chars будет больше, чем значение Char width, то содержимое поля ввода можно будет прокручивать по горизонтали. Группа переключателей Type, расположенная в верхней строке панели Properties, служит для выбора типа поля ввода. Возможно одно из трех положений: • Single line (Одна линия) — создается поле ввода в одну строку; • Multi line (Многострочное поле ввода) — позволяет вводить целый абзац; • Password (Поле ввода пароля) — все вводимые символы отображаются в виде «звездочек» (*). Если выбрать второе положение (Multi line), то поле ввода сразу увеличится в размерах и приобретет полосы прокрутки (пока недоступные). Вместе с тем на панели Properties несколько изменятся поля ввода параметров. Так, вместо поля ввода Max Chars появится поле ввода Num Lines, задающее число строк в создаваемом многострочном поле ввода. Станет доступным раскрывающийся список Wrap, значения которого определяют способ переноса слов в вводимом тексте. В нем четыре варианта: • Off — отключает перенос строк; • Virtual — перенос строк будет выполняться путем вставки так называемых «мягких переводов строк». При отправке для обработки на сервер они удаляются; • Physical — заменяет все «мягкие переводы строк» на «жесткие» перед отправкой на сервер; • Default — соответствует значению по умолчанию (Off). Увеличится и размер поля ввода Init val. Кстати, здесь вводится начальное значение, которое будет отображаться в поле ввода. Аналогичного результата можно добиться и при выборе на вкладке Forms панели инструментов в той же группе элемента Textarea (Текстовая область). Если переключатель Type установлен в положение Password (Пароль), то введенные в поле ввода Init val символы в самом поле ввода в форме будут отображаться в виде точек. В этой же группе элементов формы, чуть правее, расположен элемент HiddenField (Скрытое поле). Он не отображается на самой веб-странице. В поле ввода Value панели Properties данного элемента вводится начальное значение этого поля, которое не может быть изменено посетителем веб-страницы.

Свойства этого поля ввода очень удобно использовать при обработке разного рода анкет или заявок виртуального магазина.

Переключатели и флажки

Следующая группа элементов вкладки Forms панели инструментов объединяет переключатели и флажки. В этой группе три элемента — Checkbox (Флажок), Radio Button (Переключатель) и Radio Group (Группа переключателей). Элемент Checkbox устанавливается в форме флажок. Этот элемент используется в основном в анкетных формах. Флажок может быть установлен или нет по желанию посетителя веб-страницы. В отличие от других элементов этой группы, флажки могут работать и по одиночке, независимо друг от друга. Для установки элемента Checkbox можно выполнить команду меню Insert • Form • Check Box (Вставить • Форма • Флажок). В том и другом случае в форму вставляется маленький квадратик. Если флажок на веб-странице установлен, в этом маленьком квадратике появляется «галочка». Панель Properties свойств элемента Check Box совсем проста. В поле ввода CheckBox name (Имя флажка) вводится имя элемента — впрочем, его можно оставить и без изменений. Далее следует поле ввода Checked value (Значение флажка), предназначенное для ввода значения элемента, которое он принимает при установке данного флажка посетителем веб-страницы. Именно это значение и будет обрабатываться серверной программой. Переключатель Initial state (Начальное положение) позволяет выбрать начальное состояние данного флажка: положение Checked — установлен, положение Unchecked — не установлен. Элементы Radio Button в отличие от флажков обычно используют группами. Ведь для переключателя должно быть как минимум два возможных положения. Для вставки в форму переключателя надо выбрать второй элемент Radio Button первой группы на вкладке Forms панели инструментов либо выполнить команду меню Insert • Form • Radio Button (Вставить • Форма • Переключатель). Допустим, мы решили на веб-странице разместить небольшой опрос для выяснения возрастной статистики посетителей сайта. Для этого создаем новую форму. Устанавливаем текстовый курсор в первой строке формы и вводим текст заголовка анкеты. Затем устанавливаем текстовый курсор на новую строчку. Выполняем команду меню Insert • Form • Radio Button и вставляем в форму первый переключатель. Чуть правее от него вводим текст данного положения переключателя. Повторяем эту последовательность действий еще несколько раз. Вот и все, форма готова. Останется только подключить к ней соответствующую серверную программу, которая и обработает принятые формой данные. Внешний вид панели Properties для элемента Radio Button мало чем отличается от описанной раньше аналогичной панели для элемента CheckBox. Здесь также имеется переключатель Initial state, определяющий, будет ли флажок установлен по умолчанию. Последний элемент в группе, Radio Group, служит для создания сразу целой группы переключателей, каждый из которых аналогичен описанному выше. При выполнении команды меню Insert • Form • Radio Group (Вставить • Форма • Группа переключателей) или при нажатии на пиктограмму Radio Group (Группа переключателей) на вкладке Forms откроется диалоговое окно Radio Group (Группа переключателей), в котором задаются свойства создаваемой группы переключателей. Поле ввода Name служит для задания имени идентификатора элемента. Dreamweaver MX 2004 самостоятельно вписывает сюда стандартное имя группы переключателей. Главное, не перепутать его с именем отдельного положения переключателя. Большую часть окна Radio Group занимает поле-таблица Radio buttons (Кнопки переключателя).

Именно здесь в колонке Label указывается название, соответствующее очередному положению переключателя, а в колонке Value — значение, которое будет направлено для обработки серверной программе. По умолчанию в этой таблице всего два положения переключателя с абсолютно одинаковыми свойствами. Но можно добавить еще несколько положений в эту группу. Для этого служит кнопка со знаком +, находящаяся над таблицей в левом углу. Рядом с ней располагается кнопка со знаком -, которая, наоборот, удаляет лишние (ошибочные) положения переключателя. Кнопки со стрелками служат для изменения очередности следования положений переключателей в группе. В нижней части диалогового окна Radio Group расположен переключатель Lay out using (Использовать расположение), определяющий оформление группы переключателей в форме. Здесь всего два варианта: • Line breaks (
 tags) — в этом случае переход на другую строку для размещения следующего положения переключателя осуществляется путем вставки стандартного тега
 (перевод строки); • Table — оформление группы переключателей выполняется на основе таблицы. В этом случае каждое положение переключателя занимает свою собственную ячейку таблицы. В верхней части формы располагается группа переключателей, оформленная с использованием тега
.

Списки и меню

Следующая группа элементов формы содержит два элемента общей группы — раскрывающиеся списки. List/Menu (Список/Меню) — первый элемент этой группы. Создается командой меню Insert • Form • List/Menu (Вставить • Форма • Список/Меню). В результате в форму вставляется незаполненный раскрывающийся список (меню) или просто список. Все настройки выполняются, естественно, на панели Properties данного элемента формы. В поле ввода List/Menu указывается индивидуальное имя элемента. Переключатель Type как раз и позволяет выбрать вид нашего элемента. В нем два варианта: Menu (Меню) и List (Список). На словах довольно трудно объяснить, чем они отличаются друг от друга. Поэтому, забегаю немного вперед, приведем пример, где в одной форме рядом расположены и меню и список. Все остальные свойства у этих элементов одинаковые, различаются только значения параметра Type. Сверху, как вы уже поняли, размещен пример, соответствующий типу Menu. Видным остается только один его пункт. Какой именно это будет пункт (возможна и пустая строка), указываем на панели Properties в списке Initially selected (Выделенный элемент). Здесь надо просто щелкнуть левой кнопкой мыши по соответствующему пункту. На веб-странице, загруженной в браузер, после щелчка на значке со стрелкой раскроется весь список (или меню, как его здесь называют). Вновь выбранный пункт будет отображаться и при закрытии этого меню. Снизу расположен типичный вариант списка **List** В случае когда на панели Properties переключатель Type установлен в положение **List**, на самой панели станут доступны еще два параметра. В поле ввода **Height** указывается число строк списка, которые будут видны одновременно. В поле ввода **Height** было установлено значение 4, в то время как всего в этом списке было шесть пунктов. Флажок Selection: Allow multiple (Выделение: разрешить многократное) дает возможность выбирать в списке сразу несколько (можно и все) пунктов. Однако при таком выборе придется удерживать нажатой клавишу Ctrl. Итак, мы подробно рассмотрели все нюансы настройки списков или меню. Однако о том, как эти списки наполнить, мы еще не говорили. Для этих целей на панели Properties имеется кнопка List Values (Значение списка). Она открывает одноименное диалоговое окно List Values. Большую часть этого окна

занимает таблица, состоящая из двух колонок. В левой колонке Item Label содержатся названия пунктов меню. В правой — Value — значения, которые пересылаются серверной программе. Следующий элемент в этой группе элементов вкладки Forms называется Jump Menu («Выскакивающее» меню). В отличие от описанных ранее списков или меню, в данном случае создается раскрывающееся меню (раскрывающийся список), состоящее из гиперссылок. Каждая такая гиперссылка создает переход на другую веб-страницу. При выполнении команды меню Insert • Form • Jump Menu (Вставить • Форма • «Выскакивающее» меню) откроется диалоговое окно Insert Jump Menu (Вставка «выскакивающего» меню). Основная часть данного диалогового окна разделена на две части. В верхней части, в поле ввода Menu items (Пункты меню), отображаются все названия пунктов создаваемого меню так, как они будут отображаться на веб-странице. Органы управления (кнопки, расположенные над этим полем ввода) вам уже знакомы. В поле ввода Text, которое располагается чуть ниже поля ввода Menu items, вводим название очередного пункта меню. После нажатия клавиши Enter название из поля ввода Text переместится в верхний список. Последнее поле ввода в верхней части этого окна — When selected, go to URL (Если выбран, перейти по URL-адресу). Оно предназначено для ввода URL-адреса гиперссылки, по которому будет осуществлен переход при выборе соответствующего пункта меню. В раскрывающемся списке Open URLs in (Открыть URL-адрес в) указывается место, в котором будет загружаться веб-страница, вызываемая по гиперссылке. По умолчанию в этом списке имеется только одно значение — Main window (Главное окно). В этом случае загрузка новой веб-страницы будет осуществляться в основном окне браузера. Если же сайт основан на многофреймовой структуре, то в этом раскрывающемся списке будут перечислены названия всех фреймов сайта. В поле ввода Menu name (Имя меню) требуется ввести оригинальное название идентификатор создаваемого меню. И наконец, в нижней части описываемого диалогового окна располагается группа флажков Options. В ней всего два флажка. При установке флажка Insert go button after menu (Вставить кнопку GO после меню) Dreamweaver MX 2004 правее меню разместит кнопку GO (Перейти). В этом случае переход по гиперссылке будет осуществлен только после нажатия этой кнопки. Если флажка нет, то нет и кнопки на веб-странице. В этом случае переход по гиперссылке меню будет выполнен сразу после выбора нужного пункта меню. Установка флажка Select first item after URL change (Выбран первый пункт) автоматически делает первую строку в меню гиперссылок выбранной при переходе на веб-страницу, содержащую такое меню. В заключение следует заметить, что панель Properties данного элемента ничем не отличается от аналогичной панели других элементов типа списков или меню. Поэтому для того, чтобы изменить состав меню, на панели Properties надо будет нажать кнопку List Values (Список значений). При этом откроется одноименное окно, в котором и следует изменить список меню. Как это делается, вы уже знаете.

Кнопки

Переходим к следующей группе элементов формы вкладки Forms панели инструментов. Она включает всего три элемента. Все элементы, входящие в эту группу, можно объединить под общим понятием — кнопка. Кнопка является одним из наиболее популярных элементов формы. Остановимся на элементах этой группы подробнее. ImageField (Графическое поле). Несмотря на свое название, создает не что иное, как графическую кнопку. Иными словами, вместо обычной серой кнопки с

надписью в форму будет вставлено графическое изображение. После выполнения команды меню Insert • Form • ImageField (Вставить • Форма • Графическое поле) откроется диалоговое окно Select Image Source, в котором предлагается выбрать имя файла, содержащего графическое изображение. После этого на веб-страницу будет вставлена соответствующая форма с выбранной картинкой.

В поле ввода ImageField (Графическое поле) указываем оригинальное имя-идентификатор создаваемой кнопки. В полях ввода W и H указываются соответственно ширина и высота графической кнопки. В поле ввода Src задаем имя файла, содержащего графическое изображение, используемое в качестве кнопки в данной форме. В поле ввода Alt указываем так называемый альтернативный текст, который отобразится вместо графического изображения, если последнее не загрузится на веб-страницу. Раскрывающийся список Align позволяет выбрать один из вариантов выравнивания графического изображения относительно окружающего его содержимого веб-страницы. Кнопка Edit image (Редактировать изображение) загружает программу-редактор, позволяющую сразу же подправить графическое изображение, используемое в качестве кнопки. Следующий элемент этой группы — File Field (Файловое поле) — создает элемент формы, служащий для ввода (или выбора) имени файла. А файл с указанным именем будет отправлен от посетителя сайта на сервер. Этот элемент формы иногда еще называют селектором файлов или полем ввода файлов. И то и другое в принципе отвечает сути элемента. Однако если в форме вы используете такой элемент, то необходимо выбрать метод кодирования multipart/form-data, а метод пересылки — Post. Сам элемент состоит из двух частей — собственно поля ввода и кнопки Browse. При нажатии на эту кнопку открывается стандартное диалоговое окно Windows Select File. Для вставки его в форму также можно использовать команду меню Insert • Form • File Field (Вставить • Форма • Файловое поле).

3. Вопросы для закрепления.

- 6) Что такое форма и ее назначение?
- 7) Какие элементы формы вы знаете и их назначение?
- 8) Свойства элементов формы

4. Домашнее задание: Подготовить форму регистрации участника на сайте.

Лекция 16

Тема: Использование шаблонов

Цели: *Ознакомить учащихся с основными понятиями шаблона. С основными способами создания, сохранения и редактирования шаблонов.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Шаблон
 - 2) Способы создания шаблонов
 - 3) Создание веб-страницы на основе шаблона
3. Закрепление изученного.
4. Итог урока.

Ход урока.

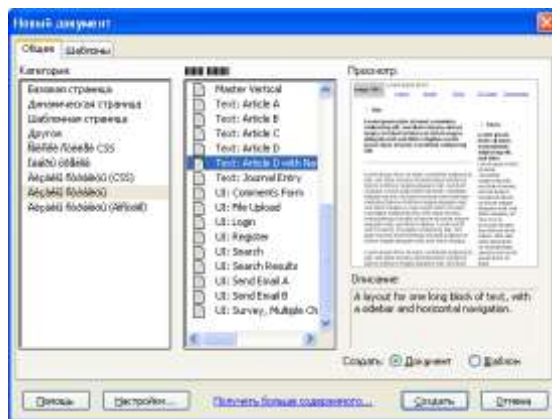
1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Шаблон — это заготовка веб-страницы, содержащая общие для всех веб-страниц данного сайта элементы: стиль дизайна, заголовков, навигационное меню, подписи и таблицу разметки. Сайт может содержать и одну веб-страницу, и больше. Если сайт содержит не сколько веб-страниц, считается хорошим тоном, чтобы все они были выполнены в едином дизайнерском стиле: имели бы одинаково оформленные заголовки, подобранную цветовую гамму, одинаковые навигационное меню и т. д. Изменяться будет только конкретное содержимое отдельной веб-страницы — в соответствии с ее назначением. Когда веб-страниц в сайте достаточно много, то каждый раз заново выполнять «с нуля» все дизайнерское оформление каждой веб-страницы, а потом еще и наполнять их индивидуальными данными — довольно трудоемкое занятие. Вот в этом случае и надо использовать шаблоны веб-страниц. Если вы однажды создадите такой шаблон, то вам останется только заполнять страницы соответствующей информацией или вносить в них несущественные изменения. К тому же и стиль всего сайта будет неизменным. Когда новая веб-страница создается на основе такого шаблона, то вносить данные можно только в специально отведенные для этого области. А общие элементы шаблона изменять нельзя. Поэтому во многих веб-студиях шаблоны выполняются художниками, а заполнение информацией производится уже другими специалистами. Для создания веб-страницы на основе такого шаблона выполняем команду меню File • New. открывшемся диалоговом окне New Document на вкладке General выбираем список Category. В этом списке выбираем пункт Page Design (Дизайн страницы).

В расположенном чуть левее списке Page Design (Дизайн страницы) и будет представлен список возможных шаблонов оформления веб-страницы. Здесь же можно их предварительно просмотреть в окне Preview (Свойства).



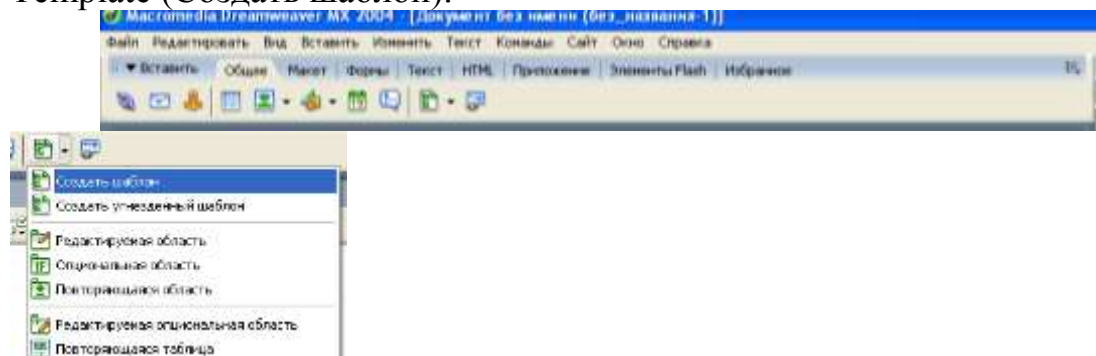
Создание нового шаблона

Создавать свой собственный шаблон можно двумя способами. Во-первых, можно создать первую веб-страницу сайта, а затем преобразовать ее в шаблон, после чего удалить все индивидуальные данные этой веб-страницы. Во-вторых, можно создать шаблон на чистом месте, сохранить его, а уже затем наполнять соответствующими данными.

1 способ. Создание шаблона на основе существующей веб-страницы

Итак, мы уже создали первую веб-страницу нашего сайта. Эта веб-страница содержит конкретную информацию: картинки, текст, навигационное меню, счетчики, баннеры, заголовок и подписи, а также табличную разметку, определяющую, по существу, весь дизайн веб-страницы. Мы долго трудились над созданием этой веб-страницы, точно вымеряли все интервалы, подбирали цветовую гамму, расположение различных элементов дизайна и т. д. Поэтому именно на основе этой веб-страницы мы и будем создавать шаблон для остальных веб-страниц нашего сайта.

затем сохраняем ее как шаблон. Для этого выполняем команду меню File • Save as Template (Файл • Сохранить как шаблон). Кроме того, можно на панели инструментов Insert (Вставить) открыть вкладку Common (Общее), на которой в раскрывающемся списке Templates (Шаблоны) выбрать самый первый пункт Make Template (Создать шаблон).



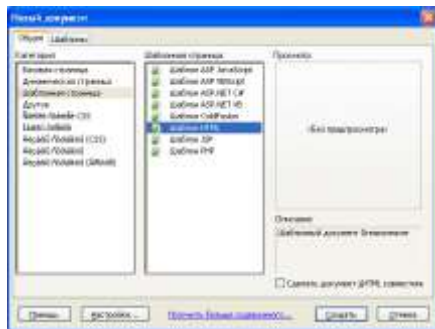
Оба способа приводят к открытию диалогового окна Save As Template (Сохранить как шаблон), представленному на рис. 10.3. Это диалоговое окно содержит один раскрывающийся список и одно поле ввода. В раскрывающемся списке Site (Сайт) выбираем один из имеющихся сайтов, для которого данный шаблон и создается. Дело в том, что шаблон является строго привязанным к конкретному сайту. В расположенном ниже списке Existing templates (Существующие шаблоны) перечислены все уже имеющиеся в выбранном сайте шаблоны. Если таковых еще нет (как и в нашем случае), то в этом списке всего один пункт — No Templates (Нет шаблонов). Если же в списке Existing templates все же имеются шаблоны, то новый шаблон можно перезаписать в один из них. И наконец, в нижней части окна расположено поле ввода Save as (Сохранить как). В это поле вводим имя нашего

шаблона. Если все параметры указаны, проверяем их и нажимаем кнопку Save (Сохранить).

Если на веб-странице, на основе которой мы создаем шаблон, имеются ссылки, Dreamweaver MX 2004 спросит, нужно ли их сохранить. Конечно, нужно. Этот шаблон будет сохранен в файле с расширением dwt. Теперь созданный шаблон будет отображаться и на вкладке Assets (Активы) панели Files (Файлы). Если эта панель не видна, то ее надо включить командой меню Window • Assets (Окно • Активы). У правой границы данной панели располагается столбец возможных активов. Мы выберем предпоследний актив — Templates (Шаблоны). Теперь в центральной части вкладки Assets представлена основная информация о шаблонах

2 способ Создание нового шаблона

Создание первым способом начинается с команды меню File • New. Как всегда, в этом случае откроется диалоговое окно New Document. На вкладке General (Общее) выбираем список Category (Категории). В нем находим и выделяем категорию Template page (Шаблонная страница). Среди восьми возможных, выбираем шестой вариант — HTML template (Шаблон HTML).



И сохраняем документ (Save as)

Изменяемые области шаблона

Участки шаблона веб-страницы, которые можно редактировать (изменять их содержимое или добавлять новое), принято называть изменяемыми областями шаблона. Соответственно, те участки шаблона, которые будут являться общими для всех веб-страниц данного сайта, не должны подвергаться никаким изменениям. Поэтому их принято называть неизменяемыми областями шаблона веб-страницы. Достаточно специально задать только изменяемые области шаблона.

Для создания изменяемой области выполняем команду меню Insert • Template Objects • Editable Region (Вставить • Шаблоны объектов • Изменяемая область). По этой команде открывается диалоговое окно New Editable Region (Новая изменяемая область).

Создание веб-страницы на основе шаблона

Когда шаблон окончательно подготовлен (на самом деле, наш шаблон еще не готов, но об оставшихся моментах мы поговорим чуть позднее) и сохранен, можно приступить к созданию веб-страниц сайта на его основе. Для создания новой веб-страницы на основе шаблона выполняем команду меню File • New. Эта команда открывает уже хорошо знакомое нам диалоговое окно New Document. Только на этот раз мы откроем вкладку Templates. При этом изменится и название самого окна, теперь это — New from Templates (Новый из шаблонов). Диалоговое окно разделено, как обычно, на три области. В крайней левой области размещен список Templates for (Шаблоны для). В этом списке перечислены все имеющиеся у нас сайты. Выбираем из них наш последний сайт — piterpivo. Вместе с этим чуть левее от первого списка

появится новый список Site «piterpivo», содержащий все имеющиеся шаблоны выбранного сайта. Мы выбираем первый шаблон shablon01. В последней области диалогового окна тут же откроется окошко предварительного просмотра внешнего вида выбранного шаблона Preview.

Под окном предварительного просмотра внешнего вида шаблона расположен флажок Update page when template changes (Обновить страницу при изменении шаблона). Он установлен по умолчанию, и в этом случае вы сможете изменять на веб-странице только информацию, размещенную в изменяемых областях. Но если понадобится для данной веб-страницы внести изменения в сам шаблон, этот флажок необходимо снять. Остается только нажать кнопку Create в правом нижнем углу диалогового окна New from Template. Созданная Dreamweaver MX 2004 веб-страница на первый взгляд ничем не будет отличаться от самого шаблона или обычной веб-страницы. Та же таблица разметки, те же изображения, та же изменяемая область и т. д. Однако при перемещении указателя мыши по пространству веб-страницы мы замечаем, что указатель изменяет свой вид в разных областях этой веб-страницы. А потом выясняется, что внести изменения в данную страницу мы можем только в некоторые ее области. На остальных участках указатель мыши принимает вид перечеркнутого кружочка. При редактировании такой веб-страницы в режиме отображения HTML-кода HTML-код не изменяемых областей окажется недоступным и будет отображаться светло-серым цветом.

Если в ходе создания веб-страницы на основе шаблона возникла необходимость внести изменения в сам шаблон, то его можно открыть командой меню Modify • Templates • Open Attached Template (Изменить • Шаблоны • Открыть прикрепленный шаблон). В результате шаблон будет открыт в отдельном документе. После внесения необходимых изменений не забудьте его сохранить.

В случае когда в веб-страницу, созданную на основе шаблона, требуется внести серьезные изменения, в том числе затрагивающие не только изменяемые области, но и многие другие, можно «открепить» данную веб-страницу от шаблона. Для этого выполняем команду меню Modify • Templates • Detach from Template (Изменить • Шаблоны • Отделить от шаблона). В этом случае предоставляется возможность полностью редактировать всю веб-страницу. При этом сам шаблон не изменяется.

3. Вопросы для закрепления.

- 1) Что такое «шаблон»?
- 2) Назначение шаблона.
- 3) Назовите два способа создания шаблонов»

4. Домашнее задание: Подготовить шаблон сайта.

Лекция 17

Тема: Свободно позиционируемые элементы

Цели: *Ознакомить учащихся с основными понятиями свободно позиционируемых элементов, установки их свойств, назначение и размещение элементов на web – странице.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Возможности свободно- позиционируемых элементов
 - 2) Параметры управления свободно- позиционируемых элементов
 - 3) Способы создания
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Свободно позиционируемые элементы — совсем недавнее нововведение в HTML. Фактически они появились в 1997 году, одновременно с таблицами стилей. И это не случайно: для создания свободно позиционируемых элементов используются особые атрибуты стилей, задающие координаты и размеры таких элементов.

Было сказано, что свободно позиционируемые элементы появились совсем недавно, но в мире компьютерных вообще и интернет-технологий в частности время летит очень быстро, и пять лет истории какого-либо нововведения — это целая эпоха. Однако до сих пор свободно позиционируемые элементы не снискали особой популярности у Web-дизайнеров. И автор не знает, в чем причина: консерватизм Web-дизайнеров или неприспособленность этих элементов к применению в Web-дизайне.

Однако потенциал свободно позиционируемых элементов очень велик. В самом деле, они позволяют полностью освободиться от "потока" текста, расположить фрагменты содержимого страницы так, как нужно Web-дизайнеру, без оглядки на ограничения таблиц и фреймов. При этом фрагменты могут располагаться друг относительно друга как угодно и даже перекрывать друг друга, чего никак не удастся сделать, используя таблицы. И все это великолепие достигается столь компактным HTML-кодом, что свободно позиционируемые элементы оказываются вне конкуренции.

У свободно позиционируемых элементов есть еще одно неоспоримое преимущество: они предоставляют Web-дизайнеру полнейший контроль над Web-страницей. Используя специально написанные сценарии, разработчик может заставить отдельные элементы страниц двигаться, создавая впечатляющие анимационные эффекты. (Вы, наверно, часто встречали на некоторых страницах анимированные курсоры мыши. Это как раз свободно позиционируемые элементы, "наученные" двигаться за мышью.) Идя дальше, программист (именно так, не Web-дизайнер, а Web-программист!) может создать на странице строку меню с выпадающими

подменю и, в конце концов, превратить Web-страничку в подобие окна Windows-приложения.

Какими же параметрами свободно позиционируемого элемента можно управлять? А вот какими:

- координатами его левого верхнего угла;
- геометрическими размерами;
- слоем (уровнем), или z-индексом;
- видимостью или невидимостью;
- поведением в случае, если содержимое этого элемента выйдет за его размеры;

- некоторыми другими, о которых будет рассказано ниже.
- z-индексе нужно рассказать подробнее. Предположим, что все созданные нами на странице свободно позиционируемые элементы "сложены" в своеобразную "стопку". И не просто "сложены", а еще и пронумерованы в -порядке "снизу" "вверх". При этом, как вы поняли, элементы с большими номерами перекрывают элементы с меньшими номерами. Так вот этот номер в воображаемой "стопке" и есть z-индекс.

Внимание!

Свободно позиционируемый элемент в любом случае перекрывает обычное содержимое страницы, лежащее в "потоке" текста.

Как видите, вы можете задать для свободного элемента довольно много параметров. Вы даже можете сделать его видимым или невидимым, что часто используется в анимации и при создании различного рода спецэффектов. Кроме того, это пригодится, если вы будете программировать для своей страницы интерфейс, аналогичный интерфейсу Windows-приложения.

Отдельно хотелось бы остановиться на одной особенности свободных элементов. Как вы поняли, они могут содержать внутри себя некое содержимое, т. е. могут являться родителями для других элементов. Так вот: можно задать поведение такого свободного элемента в случае, если его содержимое перестанет в нем помещаться. Вы можете задать, появится ли в элементе-родителе полоса прокрутки или не помещающееся в нем содержимое будет "отрезано". Таким образом, вы можете создавать своеобразные "документы в документе" без использования фреймов.

Как создается свободно позиционируемый элемент

Итак, мы выяснили, что такое свободно позиционируемый элемент. Осталось разобраться, как он создается.

Рассмотрим небольшую Web-страничку, содержащую фрагмент текста. Ее HTML-код приведен ниже.

```
<HTML>
<HEAD>
<TITLE>Пример WEB-страники</TITLE>
</HEAD>
<BODY>
```

```
<P>Это текст, который будет показан в окне Web-обозревателя. Это текст,
который будет показан в окне Web-обозревателя. Это текст, который будет показан в
окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя.
Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет
показан в окне Web-обозревателя.</P> </BODY> </HTML>
```

Как видите, ничего сложного в нем нет. Эта страничка специально была максимально упрощена, чтобы никакие "навороты" не отвлекали вас от главного.

Что же мы увидим, если откроем данную страничку в Web-обозревателе⁰ Ничего особенного: просто абзац с повторяющимся текстом. Вы можете проверить, если не верите автору на слово.

Этот текст — типичный пример фиксированного элемента страницы, находящегося в "поток" текста и зависящего от "соседей". В данном случае, правда, никаких "соседей" нет. Однако Web-обозреватель сам решает, где и как расположить этот абзац, основываясь на свободном пространстве, которое он может ему отвести. Мы не можем поместить этот абзац там, где хочется, и дать ему нужные размеры.

Теперь преобразуем его в свободно позиционируемый элемент. Для этого просто допишем кое-какой код (в листинге он выделен полужирным шрифтом).

```
<HTML>
<HEAD>
<TITLE>Пример WEB-страницы</TITLE>
<STYLE>
#para (position: absolute; left: 50; top: 50; width: 200; height: 100;
background-color: #00FF00;}
</STYLE>
</HEAD>
<BODY>
<DIV ID="para">
```

Это текст, который будет показан в окне

Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя.

```
</DIV>
</BODY>
</HTML>
```

Сохраните этот код в файле и откройте его в Web-обозревателе.

Вот мы и сделали первый свободно позиционируемый элемент. И для этого нам потребовалось внести в исходный HTML-код совсем небольшие изменения. Давайте рассмотрим их подробнее.

Прежде всего мы для создания абзаца использовали тег <DIV> вместо тега <p>. Тег <DIV> применяется для создания любого элемента страницы, фиксированного или свободно позиционируемого, содержащего внутри себя любое, простое или сложное, содержимое. В частности, его можно использовать для создания обычного текстового абзаца, что только что и было сделано. А элемент страницы, созданный с помощью тега <p>, т. е. обычный текстовый абзац, не может быть позиционирован свободно.

Затем мы дали нашему элементу уникальное имя. Назвали его para и сделали это с помощью атрибута ID, поддерживаемого практически всеми "видимыми" тегами. С помощью данного имени мы в дальнейшем зададим для этого абзаца стиль.

Такой принцип создания свободно позиционируемых элементов использовался старыми версиями Dreamweaver. Две последние версии — 4.0 и MX — применяют для

этого внутренние стили, что, на взгляд автора, порождает более компактный HTML-код, но, возможно, менее наглядно.

Рассмотрим таблицу стилей. В ней мы использовали множество новых атрибутов, которые сейчас и рассмотрим.

Этот атрибут делает элемент страницы свободно позиционируемым:

```
position: absolute;
```

Запомните его — он обязательно должен присутствовать в определении стиля любого свободного элемента. Если у вас что-то не работает, прежде всего, проверьте, задали ли вы для данного атрибута соответствующее значение.

У обычного фиксированного элемента названный атрибут установлен в значение `static`. Это же его значение по умолчанию.

Эти атрибуты задают, соответственно, горизонтальную и вертикальную координаты левого верхнего угла свободного элемента:

```
left: 50; top: 50;
```

В данный момент они заданы в пикселах, хотя вы можете использовать любую из поддерживаемых CSS единиц измерений. Имейте в виду, что данные атрибуты имеют силу только для свободно позиционируемых элементов, т. е. тех, у которых атрибут `position` установлен в значение `absolute`.

Примечание

Запомните, что координаты свободно позиционируемого элемента отсчитываются относительно родителя, а не относительно окна Web-обозревателя. Поскольку в нашем случае родителем является сама-страница, это несущественно, но вообще об этом забывать не следует.

А эти атрибуты задают, соответственно, ширину и высоту свободно позиционируемого элемента:

```
width: 200; height: 100;
```

Они также даны в пикселах, хотя могут быть заданы в любой другой из поддерживаемых CSS единиц измерения. И также имеют силу только у свободно позиционируемых элементов.

А теперь взгляните еще раз на рис. 11.1. Вы видите, что высота нашего свободного элемента заметно больше ста пикселей (сравните ее с шириной). Дело в том, что по умолчанию свободный элемент растягивается по вертикали, если его содержимое в нем не помещается. А у нас как раз такой случай. (Как уже говорилось, вы можете задать другое поведение, но об этом позже.)

```
background-color: #00FF00;
```

Здесь специально задан зеленый фон для свободно позиционируемого элемента, чтобы он был заметнее.

А теперь держитесь крепче! Добавлением всего одной строки в таблицу стилей мы превратим свободно позиционируемый элемент в небольшую "страничку в странице" (добавленный текст выделен полужирным шрифтом).

```
#para {position: absolute; left: 50; top: 50; width: 200; height: 100;
background-color: #00FF00; overflow : scroll }
```

Сохраните новый файл и откройте его в Web-обозревателе.

Давайте еще раз взглянем на добавленную нами строку:

```
overflow: scroll
```

`overflow` — атрибут, как раз и задающий поведение свободно позиционируемого элемента, когда его содержимое в нем не помещается. Значение по умолчанию — `auto`

— заставляет элемент растягиваться по вертикали, что мы и наблюдали в предыдущем случае. А значение `scroll` заставляет элемент отобразить полосы прокрутки.

Теперь посмотрим, как это выглядит в Dreamweaver MX.

```
<HTML> <HEAD>  
<TITLE>Пример WEB-страницы</TITLE>  
<SCRIPT>
```

```
<!-- Сценарий, необходимый для нормальной работы этой Web-страницы в  
старых версиях Navigator --> </SCRIPT> </HEAD> <BODY>
```

```
<DIV ID="para" STYLE="position: absolute; left: 50; top: 50; width: 200; height:  
100; background-color: #00FF00; overflow: scroll">
```

Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя. Это текст, который будет показан в окне Web-обозревателя.

```
</DIV>  
</BODY> </HTML>
```

Этот код дает тот же самый результат, что и предыдущий. Просто выглядит более компактным, особенно если выбросить код сценария, "отвечающего" за совместимость с Navigator.

Превратить фиксированный элемент страницы в свободный не составляет особого труда. Нужно всего лишь задать соответствующие стили и заменить теги `<p>` на `<DIV>`. А в большинстве случаев вам даже не нужно будет более-менее серьезно переделывать дизайн ваших страниц, если, конечно, они не основаны на фреймах или таблицах.

3. Вопросы для закрепления.

- 1) Дайте понятие свободно позиционируемого элемента
- 2) Как создается свободно позиционируемый элемент
- 3) Какой тег применяется для создания любого элемента страницы, фиксированного или свободно позиционируемого.

Лекция 18

Тема: Использование Web-анимации

Цели: *Ознакомить учащихся с основным назначением web – анимации, способы создания web анимации с помощью возможностей редактора DW.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Назначение анимации
 - 2) Создание анимации
 - 3) Размещение анимации на web – страницы
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Зачем нужна анимация

Как правило, анимация на Web-страницах преследует три цели:

- оживить страницы;
- привлечь к чему-либо внимание;
- показать что-либо в учебных целях.

поэтому можно выделить три цели применения анимации:

- развлечение;
- реклама;
- образование.

Все остальные цели, на взгляд автора, можно свести к трем вышеприведенным. А о трех вышеперечисленных мы поговорим подробнее.

Развлечения в Интернете — достаточно молодая отрасль Web-строительства. Изначально Интернет был создан как сеть для ученых, которым нужно было обмениваться текстовыми документами (сначала даже без графики) и связывать их в некое подобие структуры. Потом в Сеть пришел обыватель, и Web-дизайнеры ринулись угождать его вкусам. (Не будем спорить, насколько они возвышенны или низменны. Личное мнение автора: обыватель слишком разнолик, чтобы свести его к одному-единственному ярлыку.) В Интернете появились аудио и видео, на Web-страницы пришли сложная графика и анимация. Сейчас все эти "навороты" используются так часто, что ими уже мало кого можно удивить.

Умелое и умеренное использование анимации значительно оживит ваши страницы. Важно только понять, какого эффекта вы хотите достичь, и сделать все для того, чтобы он был достигнут. Не переусердствуйте — применяйте анимацию только там, где она действительно нужна. И уж, не дай бог, ваша страница будет рябить, как экран телевизора с отключенной антенной, — в этом нет ничего хорошего. Никогда не забывайте принцип "содержимое превыше всего, все остальное — украшения", не позволяйте украшениям заслонить содержимое.

Теперь реклама. Она появилась в Сети вместе с развлечениями, а значит, вместе с обывателем. Она уже здорово надоела, эта интернет-реклама, едва ли не больше, чем реклама телевизионная. Но отдадим должное рекламе (и интернетовской, и телевизионной) -- благодаря ей получили возможность существовать очень многие популярные некоммерческие проекты. Если реклама вдруг исчезнет, эти проекты пропадут сразу же вслед за ней.

Традиционно для рекламных целей в Сети используются так называемые баннеры — небольшие графические изображения жестко стандартизированных форматов. Почти все баннеры создаются в формате "анимированный GIF", т. е. уже используют возможности анимации по привлечению внимания потенциального клиента. Рекламную анимацию, основанную на Web-сценариях, похоже, никто еще не применял, а если и использовал, то очень мало. Так что перед вами обширное непаханое поле деятельности, которое вполне может принести неплохие плоды.

Анимация, основанная на свободно позиционируемых элементах и Web-сценариях, значительно компактнее любых видеофайлов, будь то анимированный GIF, видеофайлы форматов AVI или Apple QuickTime. Однако ани-мированные GIF-файлы поддерживаются абсолютно всеми Web-обозревателями, даже самыми старыми, поэтому и применяются так широко. Web-сценарии же будут работать только на достаточно новых Web-обозревателях (которыми, надо сказать, сейчас пользуется подавляющее большинство интернетчиков).

Так или иначе, но попробовать Web-сценарии в рекламе стоит. И, кажется, кое-кто уже пробует.

От рекламы плавно перейдем к образованию. Программы — учебные пособия, часто используемые в образовании, как правило, пишутся на компилируемых языках программирования и представляют собой обычные MS-DOS- или Windows-приложения. Учебные пособия, сделанные на основе "живых" Web-страниц, встречаются пока еще довольно редко, хотя это направление весьма перспективно. Такие учебные пособия можно очень быстро создавать и модифицировать; а по сравнению с обычными программами они исключительно компактны (ну сколько места могут занимать несложная Web-страница и пара изображений?). Вдобавок такие пособия прямо-таки просятся в Интернет, на Web-сайты, а значит, можно без особых проблем организовать модное ныне дистанционное обучение.

Правда, данные учебные пособия имеют один огромный недостаток. Любой достаточно опытный интернетчик без труда сможет просмотреть исходный код самой страницы и ее Web-сценариев и, при желании, позаимствовать оттуда парочку ваших находок. Поэтому, если вы собираетесь применять в учебных пособиях какие-либо ноу-хау, лучше реализуйте их в обычной, откомпилированной программе, Взломать которую гораздо труднее.

Создание анимации

Для работы понадобится панель **Timelines**, в которой отображаются все анимации, созданные на Web-странице. Чтобы вывести эту панель на экран, включите пункт-выключатель **Timelines** в подменю **Others** меню **Window** или нажмите комбинацию клавиш <Alt>+<F9>. Сама панель **Timelines** показана на рис. 18.1.

Как видите, панель **Timelines** находится в еще одном доке, занимающем нижнюю часть главного окна Dreamweaver. Это значит, что мы можем скрыть панель **Timelines** (и любые другие панели, которые вы поместите в док), если она нам не нужна, щелкнув по кнопке скрытия дока, а потом вернуть ее на экран.

Большую часть этой панели занимает временная шкала, но проградуированная не в секундах, а в кадрах анимации — так удобнее. Значения времени (в кадрах) написаны на серой временной шкале, расположенной выше. Под этой шкалой отображаются все доступные каналы анимации, и занятые, и незанятые. В верхней же части панели находятся несколько элементов управления, предназначенных для задания некоторых параметров дорожек и самой анимации.

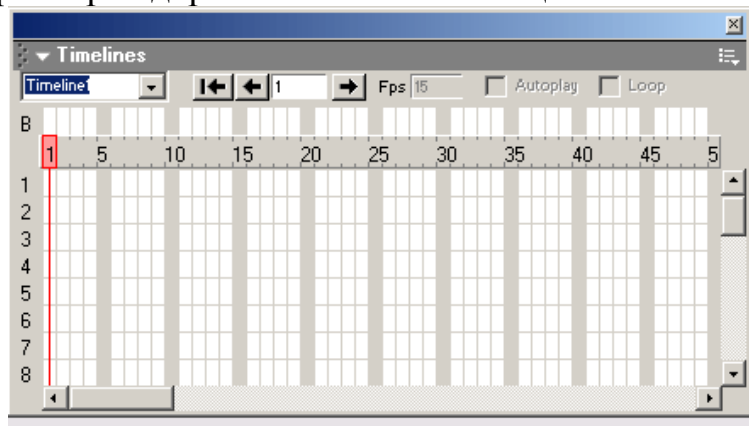


Рис. 18.1.

"Но ведь Dreamweaver позволяет работать одновременно с несколькими временными шкалами, — скажете вы. — Где же остальные шкалы?" Чтобы увидеть их, нужно выбрать соответствующий пункт комбинированного списка анимаций, показанного на рис. 18.2.

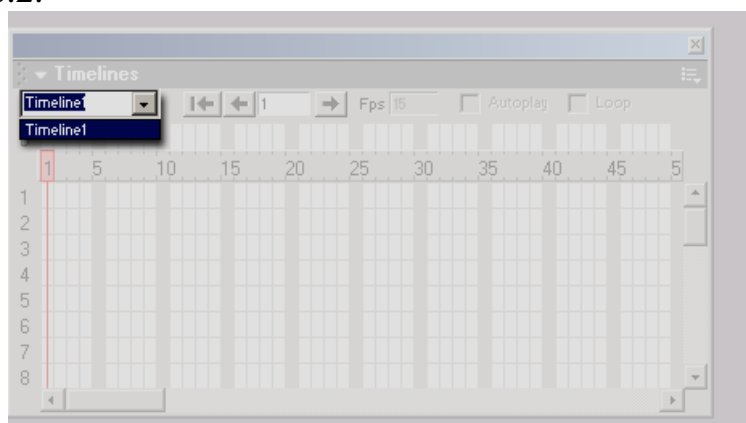


Рис. 18.2. Комбинированный список анимаций

Работой с несколькими разными анимациями мы займемся потом. А сейчас давайте создадим одну из них, самую простую.

Чтобы анимировать какой-либо из доступных свободно позиционируемых элементов, нужно поместить его на шкалу времени, создав дорожку анимации. Для этого выделим требуемый свободный элемент, щелкнем по нему правой кнопкой мыши и выберем в контекстном меню пункт **Add to Timeline**. Вы также можете выбрать пункт **Add Object** контекстного меню временной линии или пункт **Add Object to Timeline** подменю **Timeline** меню **Modify**. И, наконец, вы можете просто нажать комбинацию клавиш <Ctrl>+<Alt>+<Shift>+<T>.

После добавления в анимацию свободного элемента Dreamweaver выводит предупреждение, показанное на рис. 18.3. Этим он хочет сказать, что может управлять в процессе анимации следующими параметрами свободного элемента:

- горизонтальной и вертикальной координатами левого верхнего угла (атрибуты LEFT и TOP);
- шириной и высотой (атрибуты WIDTH и HEIGHT), причем работать это будет только в Internet Explorer;

- порядком перекрытия, иначе говоря, z-индексом (атрибут Z-INDEX);
- видимостью (атрибут VISIBILITY).

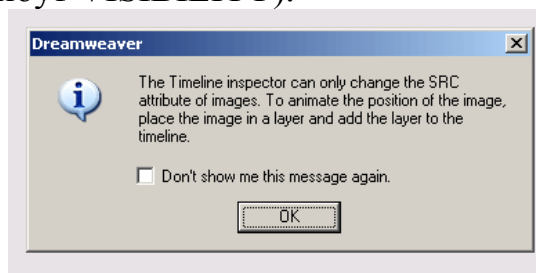


Рис. 18.3. Предупреждение, выводимое Dreamweaver после добавления свободного элемента в анимацию

Это значит, что вы можете управлять местоположением, размерами и видимостью свободного элемента. Как видите, возможностей, предлагаемых Dreamweaver для анимации, даже больше, чем мы думали.

Теперь закройте данное предупреждение нажатием кнопки **ОК**. Если вы не желаете больше его видеть, перед закрытием включите флажок **Don't show me this message again**.

Наконец, после всех треволнений, мы увидим в списке панели **Timelines** новую дорожку (рис. 18.4). Она отображается в виде светло-синей полосы, на которой написано имя свободного элемента, которому она принадлежит. По обеим сторонам данной полосы можно заметить светлые кружки. Это ключевые точки; их пока всего две: начало и конец траектории. Первая ключевая точка — начало — находится на первом кадре; это значит, что анимация для данного элемента начнется с первого кадра. Вторая ключевая точка — конец траектории — находится на пятнадцатом кадре; там наш анимированный элемент перестанет двигаться.

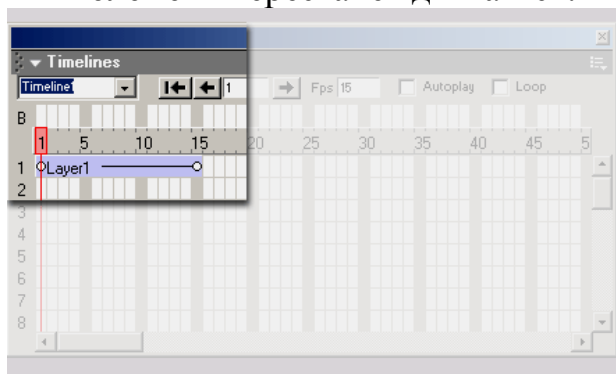


Рис. 18.4. Новая дорожка анимации

Вы можете выбрать как ключевые точки (в данном случае они выделяются темно-синим цветом), так и саму дорожку (в этом случае она вся становится темно-синей), просто щелкая по ним мышью. Более того, вы можете щелкнуть по любому месту на дорожке анимации, выделив таким образом какой-либо ее кадр. В этом случае на временной шкале появится маркер выделенного кадра, показанный на рис. 18.5.

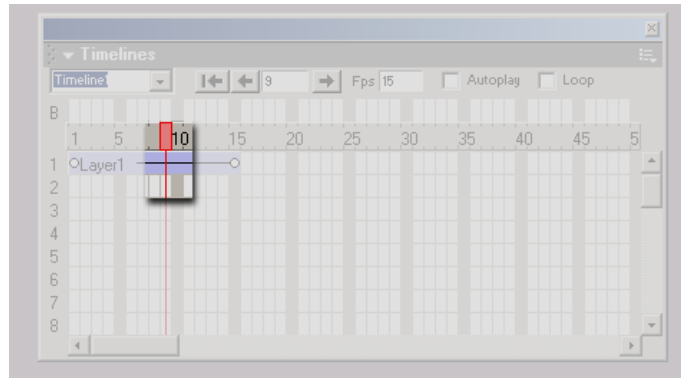


Рис. 18.5. Маркер выделенного кадра

Если вас не устраивает местоположение дорожки анимации, вы можете легко его изменить. Для этого "ухватите" мышью дорожку (не ключевые точки!) и перетащите ее вдоль шкалы, пока она не займет нужную позицию. Например, вы можете заставить анимацию начинаться с десятого кадра, а заканчиваться — двадцать пятым. Также вы можете изменять длину дорожки, а значит, продолжительность анимации, перетаскивая на необходимую позицию теперь уже ключевые точки. И, разумеется, вы можете удалить ненужную дорожку, выделив ее и нажав клавишу или выбрав пункт **Remove Object** контекстного меню или подменю **Timeline** меню **Modify**.

Ну что ж, давайте проверим свежесозданную анимацию. Нажимаем клавишу <F12>, чтобы загрузить страницу default2.htm в Web-обозревателе, и... наблюдаем совершенно неподвижное изображение. В чем же дело?

А дело в том, что начало и конец траектории движения нашего анимированного заголовка совпадают, т. е. никакой траектории нет, заголовок все время стоит в одной точке! Мы забыли задать траекторию!!!

Ну, ничего. Это не страшно. Сейчас мы все исправим.

Траектория в Dreamweaver задается следующим образом. Каждая ключевая точка представляет собой как бы "снимок" того состояния, в котором будет пребывать анимированный элемент. Выбрав ключевую точку, вы задаете нужные параметры свободно позиционируемого элемента (координаты, размеры и видимость), а Dreamweaver их запоминает, создавая тем самым данный "снимок". Исходя из этих снимков, он сам вычисляет все промежуточные состояния анимированного элемента, в которых он будет находиться между ключевыми точками. Поэтому, чтобы создать анимацию любой сложности, вам достаточно будет задать необходимые параметры нужных элементов страницы только в ключевых точках анимации. Dreamweaver сам сделает все остальное.

Предположим, наш заголовок будет "выплывать" из нижнего правого угла страницы и перемещаться прямо на свое законное место. Поскольку траектория его движения очень проста (прямая линия), для ее задания мы нуждаемся всего в двух ключевых точках. В первой ключевой точке заголовок находится в нижнем правом углу страницы, а во второй — в ее верхней части, там, где он и должен быть. В этих точках мы и должны задать параметры нашего заголовка, точнее, всего два параметра — горизонтальную и вертикальную координаты. (Наш заголовок во время движения не меняет ни размеры, ни видимость.)

Выделим первую ключевую точку, находящуюся в начале дорожки, щелкнув по ней мышью. Далее "захватим" заголовок (свободный элемент Header) мышью и переместим его в правый нижний угол страницы, в начало его траектории. Теперь

выделим вторую ключевую точку. Здесь нам менять ничего не нужно, т. к. заголовок уже стоит на своем месте. Вот, собственно, и все.

На рис. 18.6 показано то, что у нас получится после всех этих манипуляций. Хорошо видна тонкая серая линия траектории, отображаемая в окне документа, если в окне документа выделен анимированный элемент Header. Причем анимированный элемент будет находиться в том месте траектории, которое мы выделим на дорожке анимации. Так, если выделить вторую ключевую точку, он перескочит на свое законное место в верхней части страницы. А если выделить какую-либо промежуточную точку между первой и второй ключевыми точками, он займет соответствующее этой точке место.

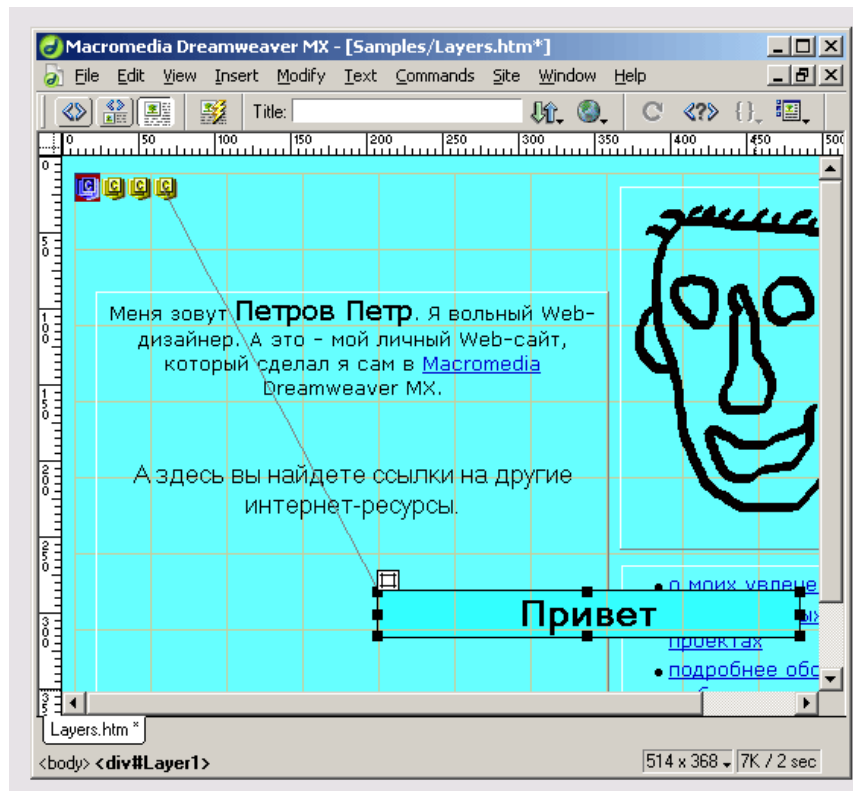


Рис. 18.6. Готовая траектория анимированного элемента (выделена первая ключевая точка)

Вот теперь можно и проверить наше творение. Загрузите получившуюся страницу в Web-обозревателе и посмотрите, что получится. А получится... все то же самое — опять-таки неподвижный заголовок. Что же мы на этот раз забыли?

Чтобы анимация запустилась сразу после загрузки страницы Web-обозревателем, Dreamweaver должен создать небольшой сценарий, который, собственно, ее и запускает. Но, по умолчанию, он этого не делает. Он предполагает, что вы хотите, чтобы анимация проигрывалась в ответ на действие пользователя, например щелчок по изображению. Но нам-то нужно, чтобы заголовок начинал свой путь сразу же, как только страница будет загружена. Для этого нам придется сделать соответствующие установки.

Сделать их очень просто. Вернемся в панель **Timelines** и включим флажок **Autoplay**, расположенный в верхней части этой панели. Dreamweaver, по своему обыкновению, выдаст очередное предупреждение, что сейчас в код страницы будет добавлен соответствующий сценарий. Избавьтесь от этого предупреждения, нажав кнопку **OK**; если не хотите больше его видеть, можете включить перед этим флажок **Don't show me this message again**. Вот теперь все на самом деле готово. Загрузите страницу в Web-обозреватель и убедитесь в этом.

Анимированный заголовок бодро пробежит заданную нами траекторию и замрет прямо над своей тенью. Конечно, плохо, что тень существует независимо от него, но мы это вскоре исправим. Не совсем хорошо также и то, что заголовок перемещается под остальными свободно позиционируемыми элементами, но вы можете исправить это прямо сейчас, просто изменив его z-индекс. Сохраним полученную страницу.

Все-таки нехорошо, что заголовок и его тень (свободный элемент Headersshadow) существуют отдельно. Давайте сделаем так, чтобы тень, пока заголовок движется к ней, была невидима на экране и появлялась лишь в самый последний момент. Для этого нам необходимо будет добавить свободный элемент, создающий эту тень, в анимацию, создав еще одну анимационную дорожку.

Выберите свободный элемент Headersshadow и добавьте его в анимацию. Как это сделать, вы уже знаете. Получившийся результат можно увидеть на рис. 18.7.

Проследите, чтобы обе дорожки находились точно друг под другом. Если вторая дорожка получилась смещенной относительно первой, измените ее местоположение и (или) размер.

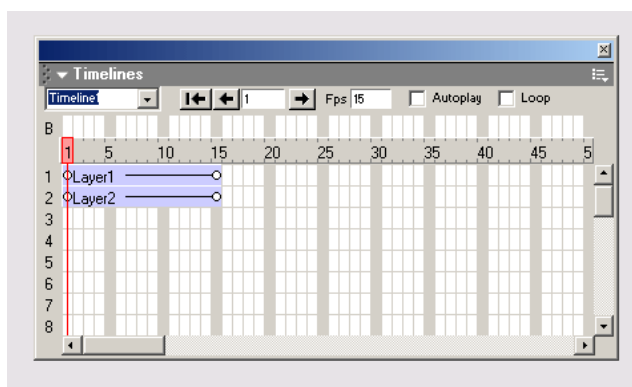


Рис. 18.7. Две дорожки анимации

Для тени мы не будем создавать траекторию движения, т. к. она не движется, а стоит на месте. Она должна просто появляться, когда заголовок "дойдет до точки". Поэтому никуда ее не перемещайте. А просто измените параметр видимости.

Итак, в начале траектории движения заголовка (первая ключевая точка) тень должна быть невидимой. Поэтому выделите первую ключевую точку второй дорожки и задайте в редакторе свойств значение **hidden** для параметра **Vis**. Теперь тень останется невидимой до тех пор, пока мы не изменим значение параметра видимости во второй ключевой точке (конец траектории движения заголовка). Так как по умолчанию параметр **Vis** имеет значение **inherit** ("наследование" видимости у родителя), а родитель (сама Web-страница) у нас видимый, специально задавать это значение во второй ключевой точке нам не нужно. На всякий случай проверьте, что на конце временной линии тень видима (параметр **Vis** должен иметь значение **inherit** или **visible**).

Сохраните результат вашей работы и откройте его в Web-обозревателе. Теперь тень будет скромно "прятаться", пока не "подъедет" ее "хозяин".

Следующим действием наш заголовок в своем пути "наверх" обходит всю Web-страницу. Это будет выглядеть эффектнее.

Сначала слегка удлиним траекторию, скажем, до 50 кадров. "Захватим" мышью вторую ключевую точку дорожки заголовка и переместим ее до соответствующего деления временной шкалы. И не забудем сделать то же самое с дорожкой элемента-тени!

Вот теперь можно и попетлять.

Как вы помните, ключевые точки, собственно, и задают траекторию движения анимированного элемента. Добавим также, что траектория его движения должна задаваться минимальным количеством ключевых точек. Для прямой, как мы уже убедились, вполне хватает двух точек, для изломанной линии — трех, а для окружности их понадобится, как минимум, десятка два, чтобы движение анимированного элемента выглядело более-менее плавным. В общем, Dreamweaver, приняв от вас набор ключевых точек, должен построить такую траекторию движения анимированного элемента, которая вам необходима. И количество таких точек, как указывалось, должно быть минимальным.

Исходя из этого правила, построим новую траекторию движения нашего анимированного заголовка. Если вы хотите, чтобы он двигался по криволинейной траектории, проставьте на дорожке анимации достаточно большое количество ключевых точек. Проще всего сначала нарисовать нужную траекторию на бумаге, расставить ключевые точки, а уже потом реализовывать это в Dreamweaver.

Но, предположим, что все это мы уже сделали. Теперь нам необходимо реализовать траекторию, что называется, в коде, поместив несколько ключевых точек на дорожку анимации. Как это сделать?

Прежде всего установите маркер выделенного кадра (см. рис. 18.5) на то деление временной шкалы, где будет находиться первая из вновь создаваемых ключевых точек. Для этого щелкните мышью по дорожке напротив нужного деления. Далее щелкните на этом месте правой кнопкой мыши и выберите в появившемся контекстном меню пункт **Add Keyframe**. Вы также можете выбрать одноименный пункт в подменю **Timeline** меню **Modify** или просто нажать клавишу <F6>. На дорожке в этом месте появится новая ключевая точка (рис. 18.8).

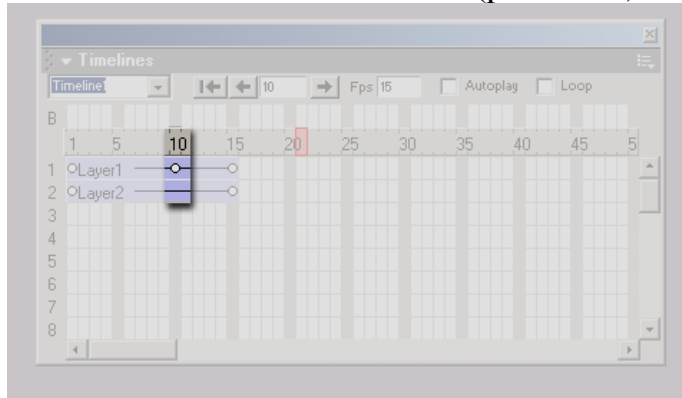


Рис. 18.8. Ключевая точка, помещенная на дорожку анимации (в данный момент выделена)

Теперь проверим, выделена ли наша новая ключевая точка, и, если нет, выделим ее. После этого активизируем окно документа и переместим заголовок в то место на траектории, где он должен находиться в настоящий момент. Тонкая серая линия, обозначающая траекторию движения заголовка, тотчас изогнется, следуя за заголовком (рис. 18.9).

Остальные ключевые точки траектории помещаются на дорожку анимации точно таким же образом. Поместите их и задайте для них новое положение анимированного заголовка. Изобретите траекторию посложнее, чтобы заставить Dreamweaver работать по-настоящему. Если вы поставили ключевую точку не в том месте, в котором хотели, выделите ее и выберите пункт **Remove Keyframe** контекстного меню. (Также вы можете выбрать одноименный пункт в подменю **Timeline** меню **Modify**.) После этого протестируйте страницу с новой анимацией в

Web-обозревателе. Вы увидите, что заголовок следует точь-в-точь по траектории, которую вы для него задали.

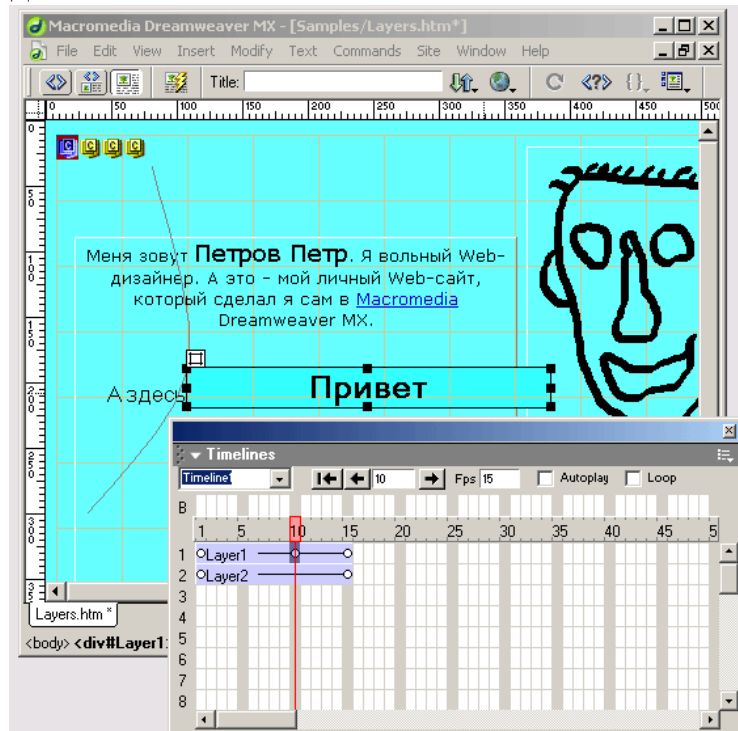


Рис. 18.9. Положение анимированного заголовка в новой ключевой точке

Здесь есть один интересный момент. Вы можете удивиться, что мы зачем-то создавали на дорожке анимации новые ключевые точки вместо того, чтобы поставить маркер на нужный кадр и просто переместить на необходимое место анимированный заголовок. Однако в этом случае мы переместили бы на другое место всю траекторию движения нашего заголовка, вместе с начальной, конечной и всеми промежуточными ключевыми точками. Попробуйте так сделать — и вы сами все поймете.

Создание траектории вручную, путем расстановки на дорожке анимации ключевых точек и задания новых параметров анимированного элемента в этих ключевых точках годится только для относительно простых случаев. В самом деле, создайте-ка таким образом что-либо более сложное, чем простая дуга! К тому же, даже опытный Web-дизайнер далеко не с первого раза сделает именно ту траекторию, какая ему нужна. Как раз для таких случаев Dreamweaver предоставляет замечательную возможность просто нарисовать нужную траекторию в окне документа. При этом он сам сформирует дорожку анимации и расставит на ней требуемое количество ключевых точек.

Давайте испытаем эту возможность. Только сначала выполним несколько подготовительных действий. Прежде всего, выделим последнюю ключевую точку (конец) траектории и запомним или запишем конечные координаты заголовка. Это нужно для того, чтобы впоследствии, когда мы создадим новую анимацию, точно его позиционировать. После этого удалим дорожку анимации заголовка, но оставим дорожку анимации его тени. Для этого выделим необходимую дорожку и выберем пункт **Remove Object** в контекстном меню. Теперь осталось только поставить заголовок в начало его предполагаемой траектории — и можно приниматься за дело.

Выделим заголовок и щелкнем по нему правой кнопкой мыши. В появившемся контекстном меню выберем пункт **Record Path**. Вы также можете выбрать пункт **Record Path of Layer** контекстного меню панели **Timelines**; этот пункт становится доступным, если в окне документа выделен какой-нибудь свободный элемент. Кроме

того, в подменю **Timeline** меню **Modify** также доступен пункт **Record Path of Layer**. Как видите, Dreamweaver предоставляет вам несколько возможностей сделать свой выбор.

Что произойдет после этого? Ничего. Кроме того, что при перемещении заголовка за ним будет тянуться тонкая серая линия — Dreamweaver начнет отслеживать и записывать его траекторию. А как только вы отпустите кнопку мыши, устанавливая заголовок в конец его траектории, он создаст новую дорожку анимации. После этого вам останется только выставить координаты заголовка в его конечной точке, воспользовавшись значениями, записанными (запомненными) ранее, и "растянуть" либо "сузить" дорожку анимации тени, чтобы она была равна дорожке своего "хозяина", иначе тень будет появляться раньше или позже, чем заголовок закончит свой извилистый путь.

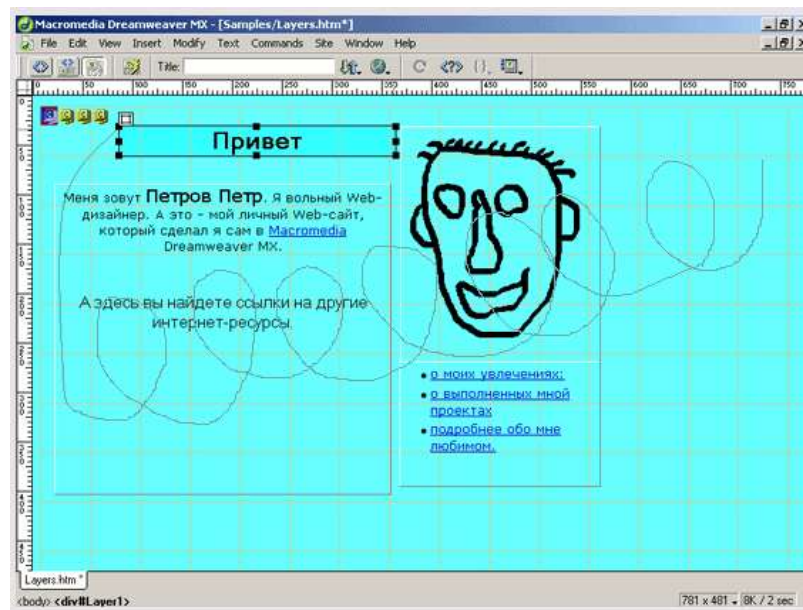


Рис. 18.10. Сложная траектория

Полученный результат можно увидеть на рис. 18.10. Сохраните страницу default2.htm, откройте ее в Web-обозревателе и посмотрите на результат ваших трудов.

Ну, вот и все. Мы познакомились с основными возможностями, предлагаемыми Dreamweaver для создания анимации. Более сложные анимации, включающие в себя множество элементов, согласованно движущихся по разным траекториям, создаются аналогичным образом. Только тогда вам предстоит несколько больше работы по заданию траекторий, выставлению параметров в ключевых точках анимации и согласованию движения анимированных элементов.

3. Вопросы для закрепления.

- 1) Что такое web- анимация?
- 2) Назначение web – анимации .
- 3) Назовите способы создания анимации

4. Домашнее задание: Подготовить доклад на тему: «Использование web-анимации в Интернет. Программы для создания web – анимации»

Лекция 19

Тема: Способы размещения сайтов в сети Интернет

Цели: *Ознакомить учащихся с основными понятиями и способами размещения и продвижения сайта в сети Интернет.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
 - 1) Основные понятия
 - 2) Домены различных уровней
 - 3) WEB-сервера, назначение, принципы организации
 - 4) Сравнение сайтов регистраторов
3. Закрепление изученного.
4. Итог урока.

Ход урока.

1. Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

2. Изложение нового материала.

Базовые понятия и термины

Web-сервер — компьютер и программное обеспечение на нем, распределяющее информацию по запросам в Интернете.

Домен — часть электронного адреса, состоящего обычно из нескольких полей. Является средством идентификации области ресурсов в Интернете. Домены бывают первого «ru», «com», «net», «org», второго «firma.ru» и третьего «forum.fir-ma.ru» уровня.

DNS (Domain Name System) — система адресации доменов, обеспечивающая преобразование имени компьютера в числовой IP-адрес сети Интернет.

MySQL — SQL-сервер баз данных. Обеспечивает размещение, хранение и выборку данных через функции доступа для приложений web-сайта. Обладает высокой производительностью, но не обладает большинством присущих для дорогих серверов баз данных возможностей.

PHP, PERL — языки программирования, широко используемые для создания интерактивных динамических приложений web-сайтов. Выполняют большой спектр задач, таких как обработка текстов, работа с базами данных, работа с сетевыми протоколами и т. д.

Скрипт — файл сценария (коммуникационная программа), написанный на PERL, PHP C++, ASP и т. п.

Трафик — суммарный объем передаваемых входящих и исходящих данных между сервером (почтовым сервером, web-сайтом, FTP-сервером) и пользователем.

Web-сервер - это выделенный компьютер с установленным программным обеспечением. Как известно, хорошие серверы стоят дорого, порядка нескольких тысяч, а порой и десятков тысяч долларов. Кроме всего прочего, сервер нуждается в поддержке системным администратором. Такие специалисты являются, как правило, высокооплачиваемыми.

Некоторые компании, у которых есть свой IP-адрес в Интернете, размещают сервер на обычном персональном компьютере, к примеру, используя для этого программу Microsoft Internet Information Server (IIS). В этом случае вы будете платить провайдеру за поддержку **доменного имени** и за трафик. Виртуальный сервер — это услуга, которую сегодня предоставляют многие хостинг-провайдеры. **Сервер называется виртуальным** в том случае, если управление его контентом (содержимым) осуществляется удаленно, а как он реализован физически провайдером, который его предоставляет, не имеет значения. Фактически хостинг-провайдеры сдают в аренду часть дискового пространства серверов своим клиентам. В этом случае провайдер обязуется поддерживать круглосуточную и бесперебойную работу Вашего виртуального сервера. Клиент лишь должен заниматься обновлением своего сайта и вовремя платить арендную плату.

Чтобы практически разместить виртуальный сервер в Интернете и он стал рабочим, минимально необходимо проделать следующее.

1. Первым делом придумайте доменное имя для компании (это имя впоследствии должно стать известным и в Интернете). **Доменное имя**, как правило, имеет такой формат записи: **www.имя_фирмы.имя_зоны**. От выбора доменного имени зависит будущее представительства в Сети. Если настоящее (уже существующее) имя компании хорошо известно на рынке, то лучше использовать именно его. Если же по каким-то причинам это имя уже кто-то занял, регистрируйте сокращенное название компании. Замечательно, если имя сайта будет совпадать со словами, известными каждому. Например, **www.anekdot.ru**, **www.rabota.ru**, **www.service_law.ru** и т. п. Если круг интересов компании относится исключительно к России, то домен следует регистрировать в зоне ru. Если же к Европе или странам зарубежья, тогда в зоне com, net, org или появившихся за последнее время новых зонах info и biz. Регистрация доменного имени стоит \$15-30 в год. Некоторые крупные компании регистрируют сразу несколько доменных имен, например «figma.ru» и «firma.com». А многие компании одновременно с регистрацией домена обеспечивают и хостинг для Вашего сайта.

2. После выбора домена надо **выбрать хостинг-провайдера**, у которого вы будете физически размещать свой виртуальный сервер и регистрировать доменное имя. В настоящее время таких компаний очень много, и они предоставляют далеко не одинаковые возможности своим клиентам.

Существует как платный хостинг, так и бесплатный. Следует отметить, что бесплатный хостинг целесообразно использовать исключительно если:

- Вы собираетесь сделать свою персональную страничку в Интернете с краткой информацией о себе, своих увлечениях и т. д.
- Ваш сайт не носит коммерческий характер.

Домены второго и третьего уровня

Конечно, если сайт вашей фирмы все же является зарегистрированным у провайдера бесплатного хостинга, то ничего криминального в этом тоже нет. Многие сотни фирм имеют именно такие сайты. Однако сам факт бесплатного размещения вашего сайта может негативно повлиять на имидж вашей компании, так как большинство солидных фирм являются обладателями **доменов второго уровня**. Если же сайт расположен на бесплатной хостинг-площадке, то его адрес в Интернете будет иметь примерно следующий вид: **http://www.xxx.yyy.ru**, где xxx — это название вашей компании, а ууу — это имя хостинг-провайдера, который и предоставляет

бесплатный хостинг. Представьте себе, что хостинг-компания «УУУ» обанкротилась или закрылась. Соответственно, ваш сайт тоже закрывается, так как название представительства вашей фирмы (которое называется в этом случае **доменом третьего уровня**) привязано к имени хостинг-провайдера (домену второго уровня). А домен всегда зависит от домена предыдущего уровня. Бесплатным бывает только сыр в мышеловке! Бесплатный хостинг окупается за счет размещения рекламы на сайтах, которую вставляют на них провайдеры в обмен на бесплатное предоставление своих услуг. Постоянная посторонняя реклама на коммерческом сайте производит негативное впечатление на посетителей, в то время как на платном хостинге посторонняя реклама отсутствует, если, конечно, вы не опубликуете ее сами. Кроме того, компании, которые предоставляют бесплатный хостинг, как правило, не несут ответственности за различные сбои на сервере и потерю информации.

Рассмотрим теперь ситуацию, когда адрес вашего сайта имеет доменное имя второго уровня, например **http://www.firma.ru**. В этом случае благополучие сайта не зависит от благополучия компании, предоставляющей платный хостинг, так как домен второго уровня фирмы зарегистрирован в DNS Интернета. Провайдер занимается лишь его поддержкой. В случае закрытия этой хостинг-компания адрес вашей фирмы в Сети останется прежним, нужно будет лишь отдать его на обслуживание другому хостинг-провайдеру. Это еще одна причина, по которой коммерческим проектам рекомендуется регистрировать домены второго уровня и использовать платный хостинг. Ведь в случае какой-нибудь неполадки на сервере провайдера вы всегда сможете отстаивать свои права или даже потребовать компенсации, если хостинг вас не устраивает.

Хостинг-площадка

Чтобы выбрать хостинг-площадку, сделайте сравнительный анализ нескольких провайдеров, которых вы можете без труда найти в любой поисковой системе. Прочитайте, что пишет хостинг-провайдер сам про себя, то есть его описание на странице поиска.

3. Определитесь, каким по объему будет ваш сайт, вернее, сколько информации по объему он будет в себе нести. Для маленьких сайтов (краткая история фирмы с контактными данными и перечень услуг, оказываемых ею) вполне хватит даже 1 Мбайта. Если же вы собираетесь размещать фотографии, тогда выбирайте объем 10-20 Мбайт. В том случае, если же сайт представляет собой крупную корпорацию и будет содержать большое количество информации, подразделы, фотографии, прайс-листы и архивы, тогда целесообразнее выбрать сразу 100 Мбайт. Для реализации крупных проектов, например, интернет-магазина с большим количеством товаров, поисковой системы с базой данных и т. п., вам может потребоваться 500 Мбайт, а то и больше. Кстати, в этом случае виртуального сервера может оказаться недостаточно, так как, с точки зрения разработчика большого проекта, рядовые провайдеры предоставляют достаточно малый набор услуг. Будем полагать, что с размером сайта мы определились.

4. Теперь нужно выбрать услуги, которые будут необходимы для функционирования вашего интернет-проекта.

Большинство хостинг-компаний предоставляют следующие программные возможности:

- запуск своих скриптов на языках PHP, PERL;
- содержание базы данных MySQL;

- гостевые книги;
- др. сервис.

Трафик

Вопрос предоставляемого хостинг-компанией трафика может решаться по-разному (он может быть ограниченным определенным значением либо вообще неограниченным). Первым делом лучше обратить внимание именно на ограничения трафика (или на отсутствие такового). Ведь если трафик, который будут создавать посетители вашего сайта, не придется оплачивать дополнительно, это выгодно и удобно!

Возможность использования PHP, PERL, CGI, SSI, MySQL, ASP, C++

Кроме трафика, надо обратить внимание на возможность использования PHP, PERL, CGI, SSI, MySQL, ASP, C++ и других программных средств. Впрочем, если проект предполагается сделать достаточно примитивным, простым и не интерактивным, тогда они попросту не нужны. Напротив, если вы собираетесь устраивать различные голосования, организовать конференции или чат, то размещение скриптов и сценариев просто необходимо. Если же вы уже имеете хостинг у хостинг-компания, а по предоставленному (ранее выбранному вами) плану вас не устраивает перечень возможностей для функционирования сайта или требуются др. возможности, - смените план. Переход на другой план у той же хостинг-компания осуществляется без каких либо проблем. Все сведется к дополнительной оплате услуг на выбранный Вами тарифный план и продолжительность оплаты по нему хостинга.

Самые популярные сегодня программные средства — это PHP и PERL, чуть менее MySQL. Если провайдер не предоставляет возможности использовать ASP, CronTab и C++, в этом нет ничего страшного. Для решения ваших задач они могут просто не понадобиться. Предоставляемых хостинг-компанией услуг уже готовых программных средств может оказаться достаточным.

Домены третьего уровня для вашего сайта

Если провайдер предоставляет возможность зарегистрировать домены третьего уровня для вашего сайта, то это несомненный плюс. В этом случае вы сможете создать на своем домене еще домены вида **http://forum.fir-ma.ru**, **http://maiLfirma.ru** и т. п. Не последнюю роль играет служба поддержки клиентов. Если компания отвечает на вопросы четко и доступно, то это большой плюс. Будет замечательно, если при покупке хостинга на год вам бесплатно зарегистрируют домен второго уровня в какой-нибудь другой зоне, отличной от той, в которой вы зарегистрированы. А еще лучше, если в стоимость услуг хостинга будет включена бесплатная первоначальная раскрутка проекта. Ведь всем известно, что, каким бы красивым не был сайт, о нем должны знать и его должны посещать.

Помните, что многие хостинг-компании стремятся к тому, чтобы их сервис запомнился своими особенностями и отличался от других, вводя различные бонусы и подарки. Этим стоит пользоваться.

Подведем краткие итоги и обозначим основные свойства, которыми должен обладать хороший хостинг:

- **Надежность.** Сервер должен бесперебойно работать. Вы должны быть уверены, что в один прекрасный момент вас не «отрубят» и все ваши старания не пропадут.
- **Качество доступа.** Сервер должен быстро отвечать на запросы пользователей. Никто не будет ждать полчаса загрузки страниц вашего сайта.

• **Сервис.** Для создания полноценного сайта вам не обойтись без загрузки файлов по FTP-протоколу, поддержки PHP, PERL и MySQL для организации форумов, гостевых книг, голосований, баз данных, магазинов.

• **Раскрутка.** Недостаточно просто разместить сайт в Интернете. Необходимо, чтобы его посещали пользователи. Несмотря на то, что эта задача относится к интернет-агентствам, все же хорошая хостинг-компания должна позаботиться о первоначальной раскрутке сайта клиента. Эти мероприятия включают целый комплекс вопросов, которые частично раскрыты в следующем вопросе лекции.

Это далеко не все аргументы, по которым необходимо судить о качестве хостинга. Бывает, что с виду хостинг хорош, а на самом деле либо каналы связи работают медленно, либо периодически без ведомой причины сервер отключается. Так что готовых рецептов на все случаи жизни просто не существует и многое следует просто опробовать, чтобы получить опыт в данном вопросе.

WEB-сервера, назначение, принципы организации

WEB-сервера - это сервера в сети Интернет, предоставляющие гипертекст, размеченный с помощью языка HTML.

Программа для просмотра и получения документов с WEB-сервера называется браузером. Обычно браузер получает информацию с WEB-сервера по HTTP-протоколу. Скорость передачи информации, которую показывают многие браузеры - это средняя скорость получения информации с учетом всех задержек (число всех полученных байт, деленное на затраченное на их получение время).

Прежде чем перейти к изучению протокола HTTP, систематизируем сведения по уже изученным протоколам:

1. IP-протокол предназначен для доставки пакетов от одной машины в сети к другой. При этом совершенно неважно, какие это данные и как они будут в дальнейшем использоваться. Вообще говоря, этот протокол даже не гарантирует доставку пакета.

2. TCP-протокол предназначен для установки двунаправленного соединения между двумя машинами. Одна машина - клиент - запрашивает соединение с другой машиной - сервером, и протокол TCP используется для организации канала, по которому эти две машины могут обмениваться данными. В качестве нижележащего протокола используется протокол IP. Упаковка данных в IP-пакеты контролируется TCP. Он пытается передать пакет до тех пор, пока не получит сообщение, что пакет доставлен и получен без искажений.

3. Над этими двумя протоколами находятся протоколы доставки почтовых сообщений, самый распространенный из которых - SMTP (Simple Mail Transfer Protocol) - отвечает за доставку почтовых сообщений между серверами. Он передает следующую информацию: от кого письмо, кому письмо и тело письма, установку соединения и корректность передачи контролирует TCP-протокол, а выбор с кем соединиться на совести почтовой программы (задается в ее настройках или берется из DNS). Протокол POP3 обеспечивает возможность пользователю получать почтовые сообщения из ящика, находящегося на сервере.

А теперь обсудим HTTP-протокол. Этот протокол также работает над TCP. Одной из его задач является регламентация передачи данных на WEB-страницах. Он используется при передаче любых данных в соответствующем формате и содержит средства для указания, какие данные он передает в настоящий момент и как с ними следует поступить.

Рассмотрим с точки зрения сети этапы просмотра WEB-страницы

1. Клиентская программа просмотра WEB-страниц (браузер) устанавливает соединение с сервером (TCP).

2. Получив ответ об установленном соединении, программа направляет запрос серверу. В запросе указывается, что ей нужен такой-то ресурс (файл, динамическая страница или каталог), желательно в такой-то кодировке и на таком-то языке.

3. Сервер направляет ответ из 3 частей:

- Код ответа (все в порядке, или документ в другом месте, или этот документ не имеют права читать все подряд, сообщите свое имя и пароль, или пароль неверный, или нет таких документов, или сервер не работает)

- Заголовки страницы (тип данных (текст+кодировка, звук, рисунок и т.д.), иногда размер, можно ли сохранять этот документ в кэше, срок хранения и т.п.)

- сам запрошенный документ.

Браузер, получив этот ответ, сохраняет его у себя либо в оперативной памяти, либо на диске (в дисковом кэше) и, исходя из заголовка документа, решает, что с ним делать... Если выводился текст в формате HTML, то там могут быть ссылки на другие ресурсы. Для каждого ресурса, например, картинки, устанавливается новое соединение с сервером, и она точно так же запрашивается.

Файловые сервера. Пересылка и получение файлов.

FTP-протокол регламентирует работу с файлами на FTP-серверах, например такие операции, как копирование файлов, удаление файлов, перемещение, получение файлов с сервера на локальную станцию, или наоборот - передачу локальных файлов на сервер, просмотр каталогов и т.д.. Мы будем рассматривать только просмотр каталогов и получение файлов с сервера. FTP-протокол передает любые данные в одном из двух форматов - текстовом или двоичном.

Этапы получения файла по FTP-протоколу:

1. Установка контрольного (управляющего, для команд) соединения с сервером; выполняет FTP-клиент. В этот этап входит получение ответа от сервера о том, что соединение установлено.

2. Сообщение серверу имени пользователя, который собирается с ним работать. В отличие от HTTP-протокола имя пользователя нужно всегда. В случае необходимости анонимного доступа используются зарезервированные имена: anonymous или ftp.

3. Сообщение серверу пароля данного пользователя. FTP-клиент может это сделать автоматически. В случае анонимного доступа стандарт требует, чтобы указывался почтовый адрес пользователя.

4. Далее следует процесс управления FTP-сессией с помощью команд. Самые типичные: сменить директорию (каталог), показать содержимое каталога.

5. Команда, подготавливающая передачу (пассивное или активное соединение). Тип зависит от того, кто будет устанавливать дополнительное соединение - сервер или станция. Если дополнительное соединение устанавливает сервер - активное, клиент - пассивное. Не все сервера и не все клиенты поддерживают пассивное соединение.

6. Команда на передачу файла

7. Собственно передача файла. Она происходит по независимому соединению. После команды на передачу файла клиент или сервер устанавливают дополнительное соединение - так называемое соединение для данных. С помощью ТСР-протокола организуется еще один канал связи, по которому и передается файл.

8. Далее клиент решает, что делать с файлом - запустить на выполнение, просмотр, записать на диск и т.п.

9. Последний этап - команда на разрыв соединения.

Регистрация ресурса и дальнейший промоушен Web-страниц и сайтов.

Общий порядок размещения и продвижения Web-страниц и готовых сайтов в сети Интернет.

1. "Теговая" подготовка htm-страниц перед публикацией
2. Загрузка подготовленных материалов на сервер, предоставляющий услуги размещения или хостинга.
3. Регистрация ресурса в поисковых системах и каталогах.
4. Регистрация на Досках объявлений.
5. Организация обмена баннерами в Баннерных системах и линками.
6. Выполнение последующих мероприятий по увеличению посещаемости ресурса.

Регистрация.

Одним из важных этапов в раскрутке сайта является регистрация его в поисковых системах и тематических каталогах России, а если сайт имеет английскую версию, то и в мировых. Бывалые веб-дизайнеры утверждают, что тратят по несколько часов на этот этап, а веб-студии берут за регистрацию только что испеченного сайта от \$5 до \$20.

Способы регистрации сайта:

Во-первых, стоит поговорить о бесплатных (и условно-бесплатных) услугах как [Add Me](#), [Submit It](#) и о нашем аналоге [Submitter.ru](#). На этих сайтах Вам необходимо заполнить специальную форму информацией о Вашем сайте (название, описание, ключевые слова ...). Затем выбрать из списка поисковых систем нужные и система сама автоматически регистрирует Ваш сайт в выбранных поисковиках.

Какой же эффект от подобных сайтов? Обещают 80-100% точность регистрации. На самом деле подтверждение о регистрации на e-mail приходит с не более 30% выбранных поисковиков. Так что судите сами. Наверное, не стоит пренебрегать данным способ регистрации, но и надеяться только на него было бы глупо.

Ниже сравнительная характеристика самых известных мировых сайтов регистраторов:

Add Me	34	0	Very popular
Submit It	400	59	Free Trial
Self Promotion	100	0	-
Usubmit	800	0	-
Submit Plus	1000	39	Free Trial
LinkoMatic	450	0	Submit to FFA
FreeSubmit	16	0	-
Broadcaster	?	82	-
Add4Free	?	0	-
Position Agent	?	0	Only rating

Submit Shack	?	0	-
Recommend-it	?	0	Affiliate Program

Есть также программы регистраторы, которые автоматически регистрируют Ваш сайт в выбранных поисковых системах и тематических каталогах. Принцип действия схож с сайтами регистраторами. При автоматической регистрации тоже используются встроенные скрипты. Безусловно, эффект от таких программ гораздо больше, так как за каждую надо заплатить от \$15 до \$300.

Итак, что же делать, если регистрация с помощью специализированных сайтов не достаточна, а платить по \$20 за программы не хочется. Конечно же, делать всё своими руками. А в чем сложность регистрации руками?

Страницы, где можно заполнить регистрационную форму и зарегистрировать свой сайт:

Aport	+	Поисковая система работает со всеми русскими кодировками, имеет гибкий язык запросов, есть возможность перевода запроса с русского на английский язык и наоборот. Результаты поиска сортируются по степени значимости, вместе со ссылкой отображается фрагмент текста, где встречается термин, а также дата и время последней модификации файла.
-------	---	--

Rambler	+	Информационно-поисковая система. 2млн. страниц на 13 тыс. серверов. Учет времени создания. Поиск в группах новостей. Одна из самых популярных российских поисковых систем. Поисковая система Rambler поддерживает рейтинг русских страниц Top100. Списки страниц разбиты на группы и многие пользователи используют данный рейтинг как каталог.
---------	---	---

List	+	Интересным аспектом каталога является наличие гидов - реальных людей, поддерживающих определенные разделы каталога. Сегодня данный каталог является самым обширным.
------	---	---

LOOK	+	Каталог ресурсов Интернета. Каждый ресурс имеет описание и баннер (88x31). Ведутся списки TOP10.
------	---	--

WebList	+	"List of Russian Servers" - один из старейших каталогов. Имеется русская и английская версия каталога. Второй адрес www.yahoo.ru .
---------	---	--

Ivan Susanin	+	Каталог только российских интернет-ресурсов. Каждая ссылка имеет подробный комментарий. Ресурсы выстроены по названию в алфавитном порядке. URL-адрес ресурса появляется внизу страницы при наведении курсора на название ресурса.
-----------------	---	--

Весь RUcский интерNET	+	Тематический каталог аннотированных ресурсов. Обновляется ежедневно.
-----------------------------	---	--

UP	+	UP.ru; каталог РуНета, бизнес, рефераты, чаты, пр...
----	---	--

Find It! + Каталог ресурсов Интернета. Поиск, халява, путешествия, библиотеки, бизнес, развлечения, архитектура и др. Рейтинг.

Вторая проблема, на которую требуется больше всего времени это заполнение регистрационных форм на сайтах соответствующих поисковых систем и тематических каталогов. Слишком долго и неприятно набирать одну и ту же информацию о своём сайте по десять раз.

Так как информация, необходимая для заполнения всегда одна, то можно один раз записать её в какой-нибудь текстовый файл и потом, используя буфер обмена вставлять в формы регистрации.

В текстовый файл необходимо записать:

Название сайта - отнеситесь к названию сайта очень серьёзно. Обычно сайт называется по названию проекта.

Владелец сайта - то есть Ваше ФИО.

URL сайта - не забудьте, что адрес сайта должен начинаться с <http://...>

E-mail владельца - на этот адрес будет приходить уведомление о том, что Вы занесены в каталог.

Логин - Часто поисковые системы идентифицируют новые сайты по ID (числу), но иногда используется логин для изменения и удаления информации о ресурсе. Желательно слово из 4-6 символов.

Пароль - Иногда поисковые системы высылают свой пароль, но чаще предлагают задать его при регистрации. Желательно слово из 4-6 символов.

Описание сайта - Краткое описание сайта от 150 до 250 символов. Попробуйте передать суть сайта, а не рекламировать свои продукты или услуги. Желательно не использовать слова “единственный”, “лучший” и т. п., так как за такое описание Вам могут отказать в регистрации или поменяют описание по своему усмотрению. Так, например, поступают в тематическом каталоге List.ru.

Ключевые слова - Около двадцати слов, по которым поисковые системы найдут сайт. Можно посмотреть на сайт конкурентов, у которых хороший рейтинг и использовать их ключевые слова (неужели и это плагиат?). Лучше всего записать ключевые слова два раза: через запятую и пробел.

Когда текстовый файл создан можно смело начинать регистрацию, используя таблицу поисковых систем Рунета из этой статьи, созданный текстовый файл и буфер обмена. На регистрацию сайта на одном сервере уходит от 30 секунд до 3 минут. Т. е. На регистрацию в десяти поисковиках необходимо от 5 минут до получаса.

Программа автоматической регистрации [AddSite](http://AddSite.ru). Она загружает страницу добавления сайта, используя большие базы данных по российским поисковым системам и тематическим каталогам и автоматически заполняет форму регистрации. Остается проверить правильность заполнения и нажать кнопку “Послать”. Еще программа может систематизировать информацию, полученную от поисковых систем после регистрации (HTML-код, ID, свои комментарии...). За программу надо заплатить от 1600 до 10000 рублей в зависимости от возможностей. Но есть Trial-период 20 дней без функциональных ограничений.

Итак, пора подводить некоторые итоги. Хочется сказать, что Рунет далеко не впереди планеты всей в области регистрации сайтов в поисковиках. Мало программ и сайтов облегчающих регистрацию. Но, все же воспользовавшись способами можно хорошо повысить трафик сайта, хотя большую роль играют описания и (самое

главное) содержание сайта. Если приходится регистрировать десятки сайтов ежемесячно, то, наверное, целесообразней было бы купить какую-нибудь программу, т.к. она очень быстро окупиться.

Полезные ссылки.

<http://www.iwr.com/free/submitmaster.htm> - перечень разнообразных средств для регистрации вашего сайта в десятках основных и второстепенных поисковых машин, каталогов и на бесплатных досках объявлений.

<http://myfreeofficelinks.com/lsubmit/index.cgi?s=myfreeoffice&p=losttema> - одно нажатие и вы прописались в 9 главных поисковых машинах.

<http://www.all4one.com/all4submit/> - регистрация в главных поисковых машинах с верификацией.

<http://jimtools.com/> - мощное средство для регистрации сайта в огромном количестве каталогов: 52 поисковых машины, 1160 FFA (Free For All message boards), 144 каталога.

<http://www.aaa.com.au/submit/> - одно нажатие и вы прописались в 13 главных поисковых машинах.

http://www.tti.co.jp/add_url/ - одно нажатие и вы прописались в 33 главных поисковых машинах.

<http://linkplace.com/winner/> - регистрация в поисковых машинах, линкообменниках, обмен баннерами и много других "вкусностей".

SUBMITBLASTER - ваша ссылка будет добавлена в каталог 21-й поисковой машины и вы увидите результат.

WEST COLORADO SUBMIT - ссылка на Ваш сайт будет добавлена в каталог 9 поисковых машин.

<http://iwr.com/free/reciprocal.htm> - обмен баннерами.

3. Вопросы для закрепления.

- 1) Дайте понятие Web – хостинга?
 - 2) Дайте понятие web – сервера .
 - 3) Назовите способы регистрации сайта в сети Интернет
- 4. Домашнее задание:** Подготовить доклад на тему: «Публикация сайтов в Интернет и их дальнейшее сопровождение, «Установка web-сервера Apache», «Администрирование web-сервера Apache»