

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования
«ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

А.В. Кудинов

**МУЛЬТИМЕДИА ТЕХНОЛОГИИ.
ЦИКЛ ЛАБОРАТОРНЫХ РАБОТ.**

Методические указания к циклу лабораторных работ по дисциплине
«Мультимедиа технологии» для Инновационной образовательной
программы «Компьютерный анализ и интерпретация данных»
подготовки магистров по направлению «Информатика и
вычислительная техника»

УДК 681.3.06

К64

Кудинов А.В.

К64 Мультимедиа технологии. Цикл лабораторных работ.: Методические указания к циклу лабораторных работ /А.В. Кудинов. — Томск: Изд. ТПУ, 2007. — 41 с.

В методических указаниях к циклу лабораторных работ обсуждаются проблемы использования современных средств и технологий для создания мультимедиа программных продуктов. Пособие подготовлено на кафедре вычислительной техники, соответствует программе дисциплины и предназначено для магистрантов направления «Информатика и вычислительная техника» по магистерской программе «Компьютерный анализ и интерпретация данных».

УДК 681.3.06

Рекомендовано к печати Редакционно-издательским Советом
Томского политехнического университета

Рецензент

Кандидат технических наук,
начальник отдела информационных технологий ОАО «ТомскНИПИнефть ВНК»

А.А. Напрюшкин

Научный редактор

Доктор технических наук,
заведующий кафедрой вычислительной техники ТПУ Н.Г. Марков

© Томский политехнический университет, 2007
© Оформление. Издательство ТПУ, 2007

ВВЕДЕНИЕ

Предлагаемое методическое пособие предназначено для магистрантов направления «Информатика и вычислительная техника» по магистерской программе «Компьютерный анализ и интерпретация данных» факультета автоматизации и вычислительной техники. Пособие используется при выполнении лабораторных работ и индивидуальных заданий по дисциплине «Мультимедиа технологии».

Мультимедиа – комплекс программных и аппаратных средств, позволяющих пользователю работать в диалоговом режиме с разнородными данными (графикой, текстом, звуком, видео), организованными в виде единой информационной среды.

Целью курса является ознакомление с областями применения мультимедиа приложений, изучение конфигурации технических средств мультимедиа, знакомство с программными средствами мультимедиа, а также этапами и технологией создания продуктов мультимедиа.

В процессе выполнения лабораторных работ студенты должны используя возможности всех имеющихся аппаратных средств и программного обеспечения для создания мультимедиа проекта получить следующие навыки:

- подготовки и проведения мультимедийной презентации;
- создание анимированных изображений;
- создание документов в HTML-формате.

Содержание дисциплины «Мультимедиа технологии» базируется на знаниях, полученных в курсах «Теория информации», «Технология программирования», «Базы и банки знаний», «Компьютерная геометрия и графика», «Интерфейсы информационных систем».

1. СОЗДАНИЕ МУЛЬТИМЕДИЙНОЙ ПРЕЗЕНТАЦИИ

1.1. Подбор материала, подготовка доклада и сценария презентации

Целью данной работы является получение студентом навыков, необходимых для самостоятельной подготовки и проведения мультимедийной презентации.

Выполнение лабораторной работы должно включать следующие этапы:

1. Выбор темы (по выбору студента из заданного списка или тема, предлагаемая студентом самостоятельно по согласованию с преподавателем).
2. Сбор материала по выбранной теме.
3. Составление устного доклада на 5 мин.
4. Разработка сценария презентации по указанной форме.
5. Сбор и оформление графического материала.
6. Оформление презентации в Microsoft PowerPoint.
7. Проведение презентации перед группой и комиссией из числа преподавателей кафедры.

Основные требования, предъявляемые к устному докладу:

1. Доклад должен точно соответствовать выбранной теме и полностью ее раскрывать.
2. Объем текста доклада должен соответствовать регламенту презентации (5 мин. – примерно 1,5 страницы печатного текста при среднем темпе речи).
3. Доклад должен быть адаптирован к устной речи.
4. Доклад должен начинаться с приветствия, представления автора и темы, обозначения краткого плана.

N кадра (слайда)	Голос за кадром	Примечания (действия лектора)

Основные требования, предъявляемые к сценарию презентации:

1. Сценарий рекомендуется выполнять в Microsoft Office Excel.
2. В графе "Голос за кадром" пишется ВЕСЬ текст, который нужно сказать при показе конкретного слайда.
3. В графе "Примечания" указывается те действия, которые должен выполнить лектор при выступлении во время показа данного слайда:
 - a. указать мышкой или указкой на ту часть слайда, о которой идет речь;
 - b. нажать клавишу для появления или исчезновения объекта в случае развивающегося слайда;
 - c. запустить или остановить демонстрацию динамического объекта (звук, фильм);
 - d. и т.д.

1.2. Оформление мультимедийной презентации при помощи Microsoft Office PowerPoint

Требования к мультимедийной презентации в формате Microsoft Office PowerPoint:

1. Текст на слайдах не должен дословно повторять текст доклада.
2. Текст на слайдах должен быть разборчивым: следует обратить внимание на выбор шрифтов и размера символов.

3. Выбирая шаблон оформления (стиль) презентации следует учитывать его соответствие теме, аудитории и способу проведения (демонстрация через проектор требует более контрастных сочетаний фона и текста).
4. На слайды в обязательном порядке выносятся:
 - a. графический материал, иллюстрирующий доклад: рисунки, фотографии, схемы и т.д.;
 - b. все приведенные в докладе даты и числа, аббревиатуры с расшифровкой, имена собственные, имена и термины на иностранных языках с переводом;
 - c. термины и другие понятия, на которые в докладе делается акцент;
 - d. другие материалы, тяжело воспринимаемые на слух.
5. Первым слайдом презентации должен быть титульный слайд, на который выносятся тема и имя автора. В случае если авторов несколько – имя докладчика подчеркивается.
6. На втором слайде приводится краткий план презентации (3-5 пунктов).
7. Подбор материала для слайда необходимо вести таким образом, чтобы не перегружать его: один слайд, как правило, не должен оставаться на экране дольше 1 мин. Также не следует стремиться к слишком частой смене слайдов (менее 15-20 сек на слайд).
8. В случае отсутствия графического материала для слайда на него выносятся тезисы автора.
9. Презентация должна способствовать концентрации внимания зрителя на докладе: следует стремиться к постепенному выводу требуемых материалов на экран в точном соответствии с темпом доклада.
10. Использование эффектов анимации (звука, видео) должно быть направлено исключительно на акцентирование внимания зрителя на каких-либо моментах доклада.
11. В конце презентации приводится слайд со ссылками на использованные источники.

12. Последним обычно располагают слайд с благодарностью зрителям за внимание и контактной информацией об авторах.

Программное средство Microsoft PowerPoint 2000 входит в состав пакета Microsoft Office 2000. PowerPoint представляет собой мощное и гибкое средство с помощью которого возможно создание:

- презентаций на экране;
- Web-страниц для размещения в Интернете;
- цветных и черно-белых прозрачек;
- цветных и черно-белых распечаток;
- 35-мм слайдов;
- выдач;
- заметок докладчика.

Презентации возможно создавать, как для экрана монитора, так и для проектора. В пакете имеется встроенный инструментарий, позволяющем подготавливать презентации для различных типов проекторов, переносить презентацию с переносного компьютера с выводом на проектор на стационарный с монитором и т.п.

Каждая «страница» презентации называется слайдом. Совокупность «страниц» называется презентацией. Презентация может содержать неограниченное число слайдов, хранящихся в одном файле.

Для удобства работы с презентациями в PowerPoint существуют различные режимы. Режимами называются способы отображения и работы над презентацией. В пакете существуют следующие типы режимов:

- обычный;
- область структуры;
- область слайда;
- сортировщик слайдов;
- показ слайдов.

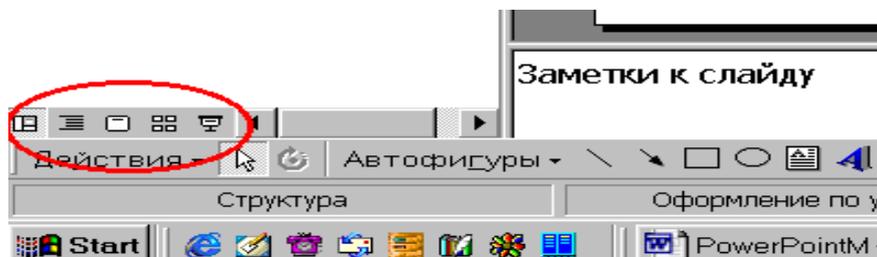


Рис. 1.1. Кнопки переключения режимов

Переключение режимов осуществляется кнопками в левом нижнем углу окна PowerPoint .

При работе со слайдом в обычном режиме вид интерфейса PowerPoint выглядит следующим образом.

В области слева находится структура текста, которая содержит в себе дополнительные сведения по области структуры. В области справа находится текущий слайд со всеми объектами в котором содержатся дополнительные сведения по области слайда. Ниже области слайда расположена область заметок. Заметки служат для удобства понимания содержимого слайда и его месторасположении в презентации. В обычном режиме возможно редактирование текстовых окон в области структуры текста, расположенной слева. Текстовые надписи в PowerPoint аналогичны «надписям» в MS Word или «кадрам» в более ранних версиях Word.

Режим область структуры

В области структуры показан текст презентации без графических объектов. Она предназначена для быстрого ввода, правки и упорядочивания текста. Размер области структуры можно изменять. Также для увеличения рабочей области можно перейти в режим структуры. Изменение порядка слайдов или пунктов маркированного списка осуществляется простым перетаскиванием их на новое место.

Режим область слайда

В этой области можно просмотреть каждый слайд презентации. Она предназначена для добавления графических объектов, текста, диаграмм и других объектов. Переход к другим слайдам осуществляется с помощью полосы

прокрутки. Размер области слайда можно изменять. Также для увеличения рабочей области можно перейти в режим слайдов.

Режим сортировщик слайдов

Этот режим предназначен для удобства просмотра всей презентации. Также в режиме сортировщика удобно менять местами слайды т.е. изменять порядок их просмотра. В этом режиме все слайды презентации показаны в виде маленьких эскизов. Чтобы поменять местами один или несколько слайдов, выберите их и перетащите в новое место.

Режим показ слайдов

Этот режим используется для немедленного просмотра презентации.

В этом режиме также можно выполнить репетицию показа презентации или провести презентацию перед аудиторией в электронном виде. Презентация будет показана в полноэкранном режиме с учетом анимации и способов смены слайдов. Переход к следующему слайду осуществляется щелчком мыши, а перемещение вперед и назад - клавишами управления курсором.

Создание первой презентации

При первом запуске PowerPoint выберите вариант мастер автосодержания.

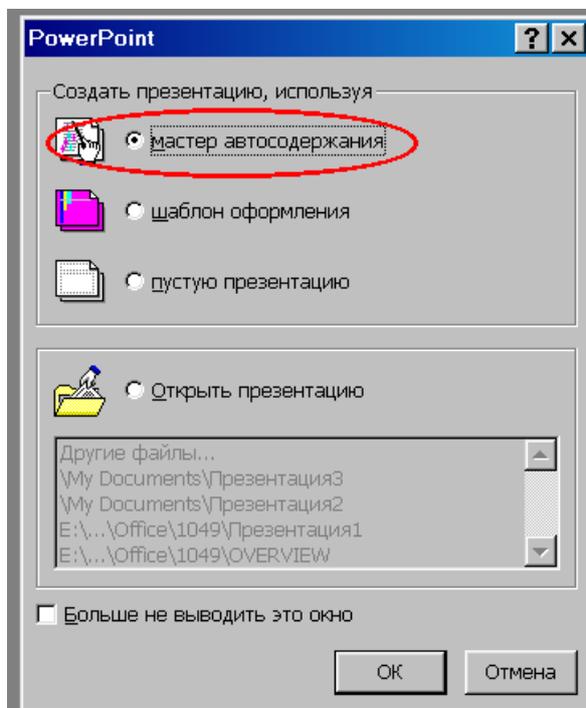


Рис.1.2. Мастер автосодержания

Мастер автосодержания создает презентацию с титульным слайдом и несколькими слайдами с заголовками. Содержимое и организация презентации основываются на ответах пользователя. Следуйте пошаговым инструкциям мастера. По завершении мастер автосодержания создает презентацию с титульным слайдом и несколькими слайдами с заголовками.

Выделите предложенный текст и замените его собственным. Для перехода к следующему слайду воспользуйтесь полосой прокрутки или колесиком мыши (для устройств имеющих такую возможность) .

Мастер автосодержания также поможет создать презентацию, если вы не знаете с чего начать.

Чтобы задать вид презентации, при запуске PowerPoint выберите вариант шаблон оформления.

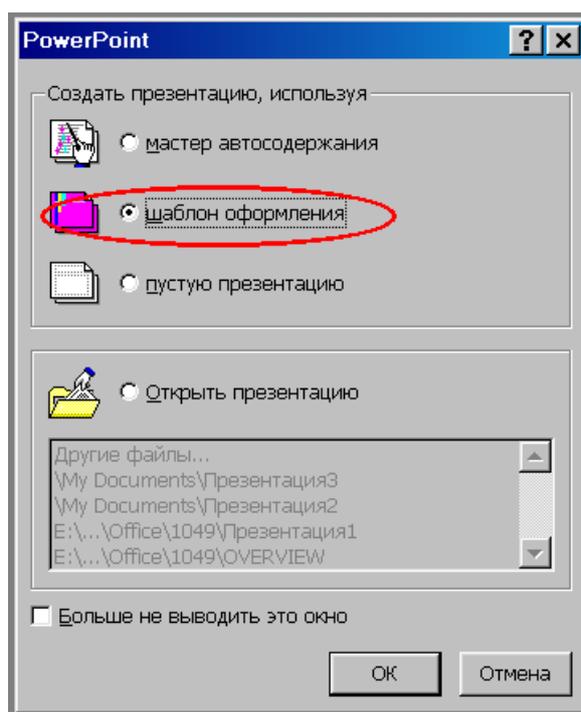


Рис.1.3. Шаблон оформления

Текущий выбранный шаблон оформления отображается в окне просмотра. Теперь выберите макет. Автомакеты PowerPoint обеспечивают выравнивание и задают расположение текста и объектов на слайде.

Выберите автомакет, наиболее подходящий для размещения имеющейся информации.

Чтобы изменить текст в рамке на слайде, щелкните рамку и начните ввод.

Маркеры добавляются автоматически.

Для увеличения отступа (также называемого уровнем) служит клавиша Tab.

Для добавления нового слайда выберите команду **Новый слайд** из меню **Вставка**.

Изменение первой презентации

Шаблоны оформления являются примерами профессионального дизайна, которые в любой момент можно применить к текущей презентации. Каждый шаблон оформления содержит образцы слайдов, задающие тип фона, цвета, стили маркированных списков и размер шрифта.

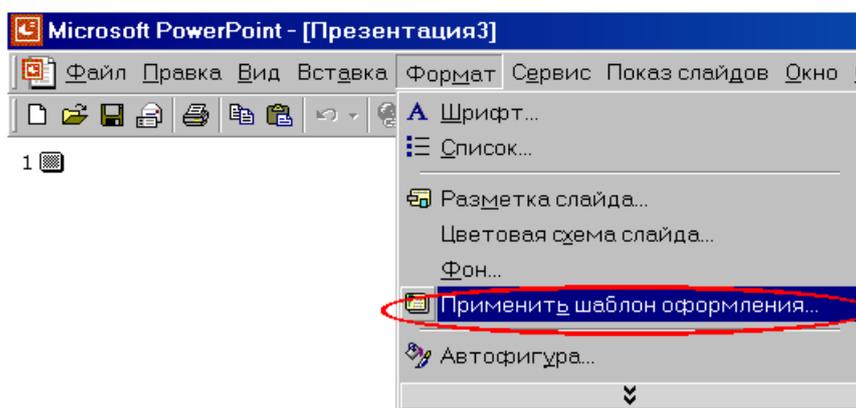


Рис.1.4. Меню **Формат**

Выберите в меню **Формат** команду **Применить шаблон оформления**, затем выберите новый дизайн для всей презентации. Содержание презентации при этом не изменится.

Выбор новой цветовой схемы

Цветовой схемой называется набор из 8 гармонирующих цветов для заголовков, текста и графических объектов. В каждом шаблоне оформления

можно выбрать одну из нескольких имеющихся цветовых схем. Новую цветовую схему можно применить к нескольким или ко всем слайдам текущей презентации.

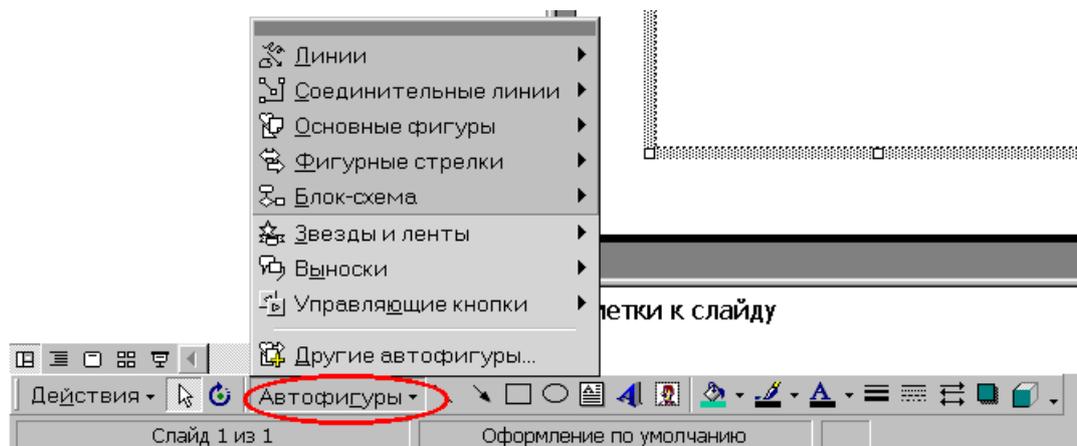


Рис. 1.5. Автофигуры

Добавление фигур и схем

Для создания фигур и схем в PowerPoint используются автофигуры.

Щелкните автофигуру и начните ввод. Вводимый текст автоматически будет размещен в автофигуре.

Добавление диаграммы

Вставьте новый слайд и выберите макет с диаграммой. Дважды щелкните рамку диаграммы, чтобы создать собственную диаграмму. Замените данные, вставленные для примера, собственными. Чтобы вернуться в PowerPoint, щелкните слайд вне диаграммы. Чтобы продолжить изменение диаграммы, дважды щелкните ее.

Добавление картинки

Чтобы открыть коллекцию клипов, нажмите кнопку Добавить картинку. Выберите любое из сотен имеющихся изображений. Картинку можно перемещать по слайду и изменять ее размер.

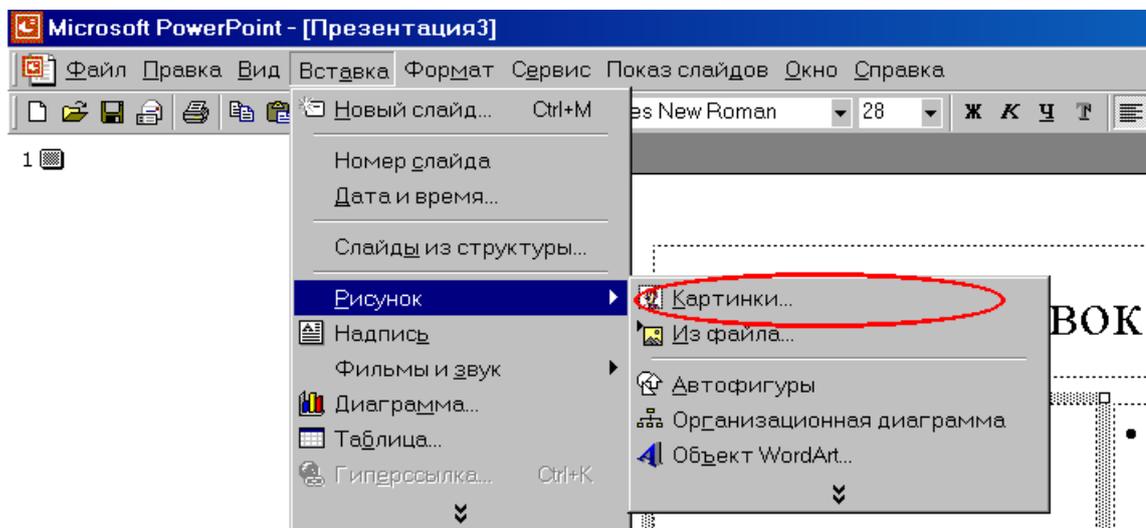


Рис.1.6. Меню **Вставка**

На компакт-диске Microsoft Office и в Интернете имеются сотни разных картинок.

Добавление эмблемы и внесение изменений сразу во все слайды

Чтобы добавить или изменить какой-либо объект сразу на всех слайдах, воспользуйтесь образцом слайдов.

Образцы слайдов задают форматирование заголовков, текста, маркированных списков и элементов фона, присутствующих на каждом слайде презентации.

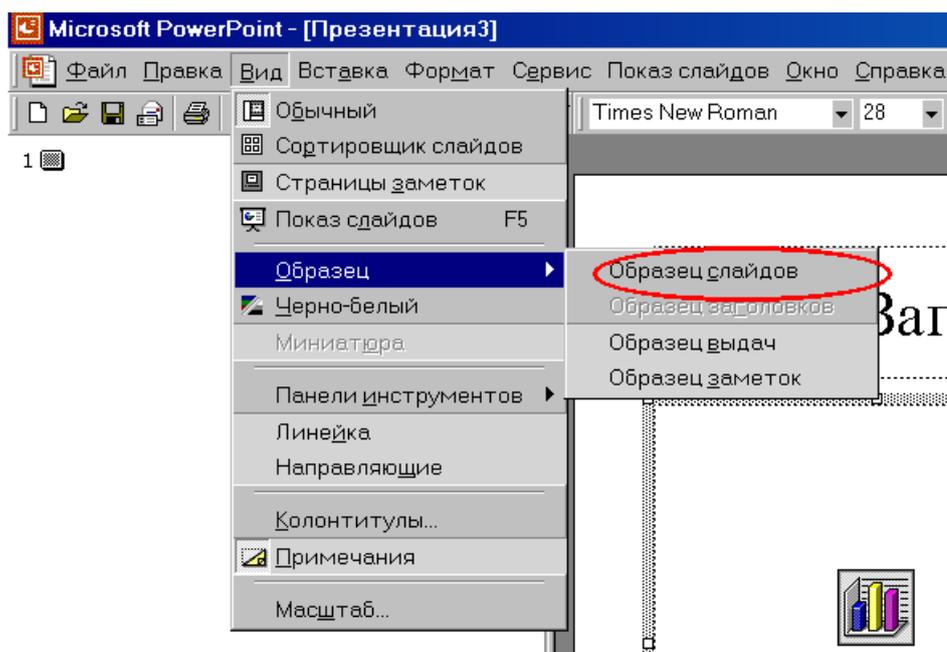


Рис. 1.7. Меню **Вид**

В меню **Вид** укажите на команду **Образец** и выберите пункт **Образец слайдов**. Вставьте на образец слайдов эмблему компании. По завершении работы над образцом слайдов переключитесь в обычный режим.

Добавление других графических объектов

На слайды можно добавить множество других объектов

- Рисунки
- Организационные диаграммы
- Таблицы
- Графические объекты WordArt
- Клипы мультимедиа
- Фильмы и звуки
- Комментарий

1.3. Рекомендуемые темы презентаций

1. Современные методологии разработки ПО. Сравнительный анализ.
2. Обзор рынка CASE-средств.
3. Инструментальные средства визуального программирования.
4. Современные СУБД.
5. Средства коллективной разработки: основные возможности и обзор рынка.
6. Методология и инструментальные средства тестирования ПО.
7. Компьютерные вирусы: классификация и методы защиты.
8. Microsoft Active Directory: назначение и основные возможности.
9. Топологии вычислительных сетей: сравнительный анализ.
10. Активное и пассивное сетевое оборудование.
11. История создания глобальной сети Internet.
12. Архитектура ОС Microsoft Windows 2000.

13. Архитектура ОС Linux в сравнении с Microsoft Windows.

14. Современные геоинформационные системы.

2. СОЗДАНИЕ АНИМИРОВАННЫХ ИЗОБРАЖЕНИЙ В СРЕДЕ MACROMEDIA FLASH

2.1. Общие сведения

Технология Flash основана на использовании векторной графики в формате Shockwave Flash (SWF). Хотя это далеко не первый векторный формат, создателям SWF удалось найти наиболее удачное сочетание между изобразительными возможностями графики, инструментальными средствами для работы с ней, и механизмом включения результата в Web-страницы. Дополнительным преимуществом SWF является его переносимость, то есть этот формат может использоваться на любой аппаратно-программной платформе (в частности, на компьютерах Macintosh, работающих под управлением операционной системы MacOS, и на компьютерах IBM с ОС Windows). И еще одна особенность SWF: созданные на его основе изображения не только могут быть анимированы, но также дополнены интерактивными элементами и звуковым сопровождением.

Переносимость и возможность создания интерактивных мультимедийных приложений обусловили быстрый рост популярности формата SWF среди Web-дизайнеров. Поэтому почти одновременно с появлением самого формата фирмой Macromedia были созданы встраиваемые компоненты (Plug-In) для двух основных браузеров Сети: Internet Explorer и Netscape Communicator. А это, в свою очередь, способствовало еще более широкому распространению SWF на просторах Всемирной Паутины. В результате разработчики этих браузеров объявили о намерении включить поддержку SWF непосредственно в ядро своих продуктов.

Поддержали подобный подход и другие ведущие производители программного обеспечения (в частности, фирма Adobe).

И все-таки повторим еще раз: вряд ли формат SWF завоевал бы так много поклонников, если бы Macromedia не сопровождала его достаточно простыми и удобными инструментами. Необходимо отметить, что в настоящее время существует целый набор таких инструментов.

Одни из них ориентированы на создание мультимедийных презентаций (Macromedia Director Shockwave Studio), другие предназначены для подготовки графических изображений (Macromedia FreeHand и Macromedia Fireworks), третьи обеспечивают создание интерактивных обучающих курсов (Macromedia Authorware и Macromedia CourseBuilder). Однако среди разработчиков Web-публикаций наибольшей популярностью пользуется Macromedia Flash, поскольку именно этот продукт позволяет создавать законченные Web-страницы, которые способны обеспечить популярность любому сайту. Видимо, благодаря этому формат SWF все чаще стали называть просто Flash. Итак, составляющими Flash-технологии являются:

- векторная графика;
- поддержка нескольких видов анимации;
- возможность создания интерактивных элементов интерфейса;
- поддержка взаимодействия с импортируемыми графическими форматами (в том числе растровыми);
- возможность включения синхронного звукового сопровождения;
- обеспечение экспорта Flash-фильмов в формат HTML, а также в любой из графических форматов, используемых в Интернете;
- платформенная независимость;
- возможность просмотра Flash-фильмов как в автономном режиме, так и посредством Web-браузера;

- наличие инструментов визуальной разработки, избавляющих со-здателя Flash-фильмов от многих рутинных операций, а также от детального изучения технических аспектов реализации Flash-технологии.

Рассмотрим перечисленные составляющие Flash-технологии подробнее.

2.2. Возможности анимации

Анимация во Flash основана на изменении свойств объектов, используемых в «мультике». Например, объекты могут исчезать или появляться, изменять свое положение, форму, размер, цвет, степень прозрачности и т. д.

Во Flash предусмотрено три различных механизма анимирования объектов:

- покадровая («классическая») анимация, когда автор сам создает или импортирует из других приложений каждый кадр будущего «мультика» и устанавливает последовательность их просмотра;
- автоматическое анимирование (так называемая tweened-анимация), при использовании которой автор создает только первый и последний кадры мультипликации, а Flash автоматически генерирует все промежуточные кадры; различают два вида tweened-анимации: анимация, основанная на перемещении объекта (motion animation), и анимация, основанная на трансформации (изменении формы) объекта (shape animation);
- анимация на основе сценариев; сценарий представляет собой описание поведения объекта на собственном языке Flash, который называется ActionScript; синтаксис этого языка напоминает синтаксис других языков сценариев, используемых в Web-публикациях (например, JavaScript и VBScript).

Каждый из этих механизмов имеет как достоинства, так и недостатки. В частности, tweened-анимация обладает двумя несомненными достоинствами:

- во-первых, автор избавлен от необходимости создавать каждый кадр в отдельности;

- во-вторых, для воспроизведения такого «мультика» Flash достаточно хранить только первый и последний кадры, что обеспечивает значительное уменьшение объема такого фильма.

Вместе с тем, tweened-анимация пригодна для создания лишь наиболее простых сюжетов, в которых свойства объектов изменяются равномерно.

С помощью сценариев на ActionScript можно описать достаточно сложное поведение объектов, однако для этого нужно изучить язык ActionScript.

Другими словами, прежде чем приступить к созданию собственного «мультика», следует определиться с выбором механизма его реализации.

Чтобы познакомиться с теми возможностями, которые предоставляет Flash для создания анимированных Web-страниц, мы настоятельно рекомендуем вам посетить сайт компании Disney (wwwv.Disney.com), одна из страниц которого показана на рис. 2.1.

Кроме того, на сайте компании Macromedia имеется специальная рубрика, которая называется «Сайт дня» (Site of the Day). В ней содержатся ссылки на сайты, которые, по мнению сотрудников компании Macromedia, наиболее полно и эффективно используют преимущества Flash-технологии.

Flash позволяет создавать интерактивные фильмы, в ходе воспроизведения которых посетитель сайта может использовать клавиатуру или мышь, чтобы перейти к различным фрагментам фильма, переместить объекты, ввести информацию в форму, а также выполнить многие другие операции.

Интерактивность Flash-фильмов реализуется путем описания соответствующих действий в виде последовательности команд (инструкций), написанных на языке ActionScript. Действие выполняется, когда происходит связанное с ним событие. Такими событиями, в частности, могут быть переход к определенному кадру фильма или нажатие пользователем кнопки на странице либо клавиши на клавиатуре.

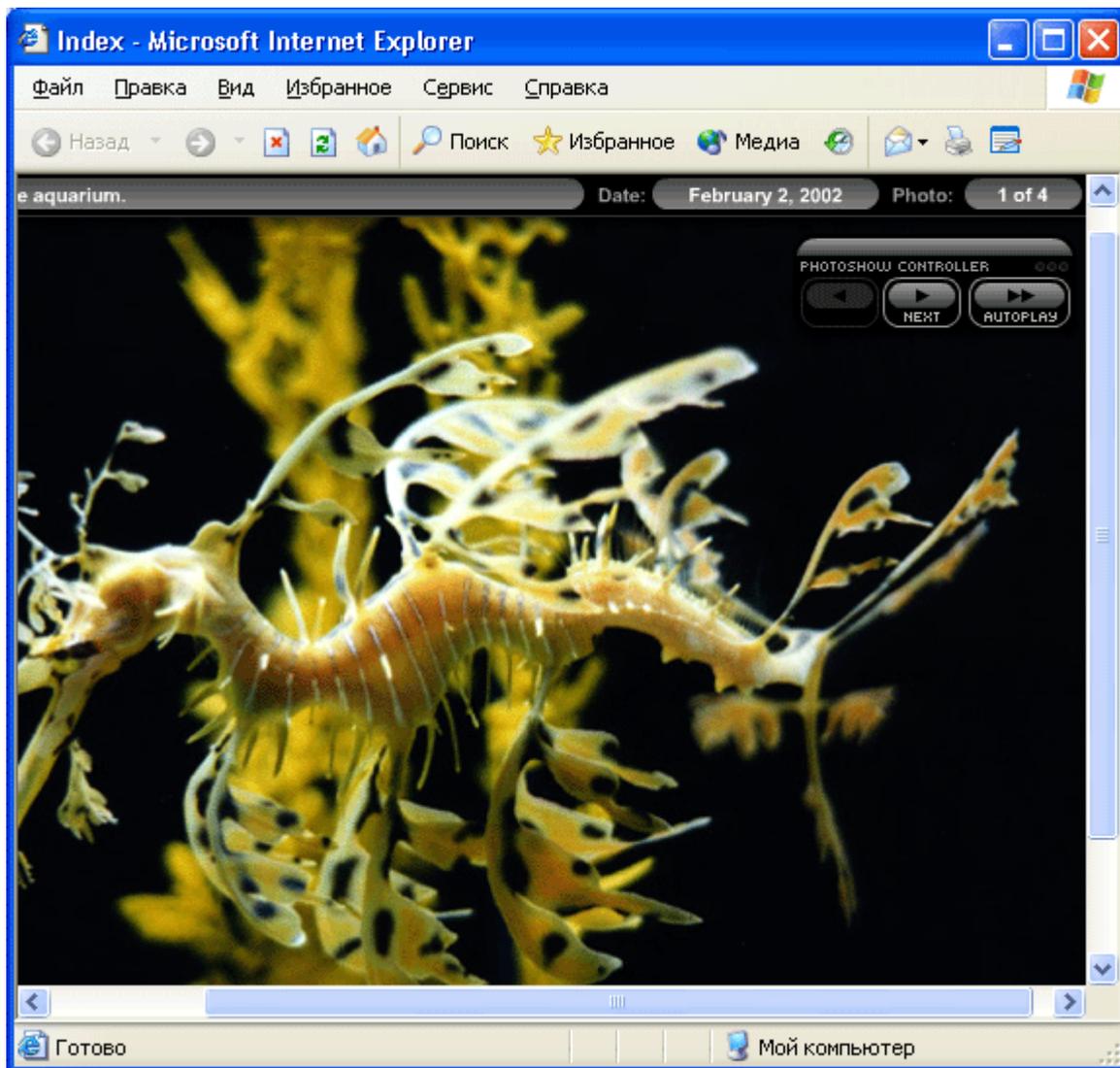


Рис. 2.1. Одна из страниц сайта

Интерактивными могут быть кнопки (например, сгруппированные в виде специальной панели управления), фрагменты фильма, либо отдельные кадры. При этом многие стандартные действия могут быть назначены тем или иным элементам фильма без непосредственного программирования на ActionScript. К стандартным действиям относятся, например, запуск и остановка фильма, включение и выключение звукового сопровождения, переход по заданному URL, и некоторые другие. Всего же стандартных действий насчитывается более десятка.

При необходимости можно создавать достаточно сложные сценарии, в которых анализируются те или иные условия, и только потом выполняется

требуемое действие. Правда, создание таких сценариев требует наличия некоторых навыков в программировании.

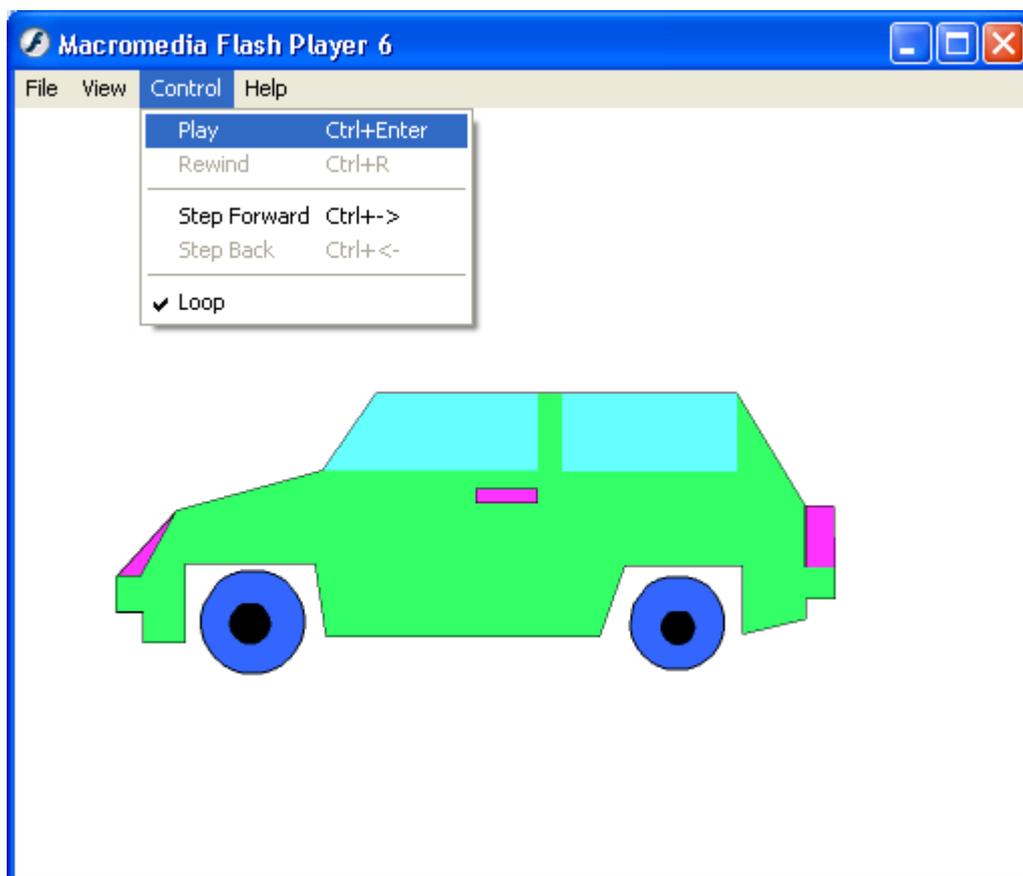


Рис. 2.2. Пример интерактивного Flash-фильма

На рис. 2.2 показан простой пример интерактивного фильма, позаимствованный из подборки примеров, входящей в комплект поставки продукта.

Приведенная на рис. 2.2. Flash-страница состоит как бы из двух частей: справа расположен своеобразный экран, на котором воспроизводится управляемый объект (в данном случае — автомобиль), а слева размещена панель управления с интерактивными «кнопками» и «ползунками»; с их помощью посетитель страницы может выбрать цвет автомобиля.

При построении любого Flash-фильма используется объектно-ориентированный подход. Это означает, что все элементы фильма интерпретируются как объекты того или другого типа, для каждого из которых

заданы некоторые свойства и определен набор допустимых операций. Например, для объекта «Текст» должен быть установлен размер символов, способ начертания, цвет и т. д. Текст можно определенным образом редактировать, вырезать; копировать, создавать на его основе текстовые гиперссылки и т. п. То же самое можно сказать о графических изображениях и о звуке. Тем не менее, при работе с Flash вместо понятия «объект» чаще используется термин символ (Symbol). Основное различие между ними состоит в следующем.

Символ представляет собой своеобразный шаблон объекта с определенным набором свойств. Символ хранится в специальной библиотеке символов (Library) и может быть многократно использован как в одном и том же фильме, так и в нескольких фильмах. Каждая новая копия символа, помещенная в фильм, называется экземпляром символа (Instance). Экземпляр наследует все свойства самого символа, и между ними устанавливается связь: при изменении свойств символа соответствующие изменения автоматически применяются ко всем его экземплярам. Очевидно, что такой подход существенно экономит силы и время создателя фильма. Кроме того, механизм символов позволяет сократить и размер фильма: если в нем используется несколько экземпляров символа, то информация о его свойствах не дублируется. Вместе с тем, вы можете изменять некоторые свойства конкретного экземпляра, что не влияет на свойства символа-оригинала. Например, можно изменить размер и цвет экземпляра, а если речь идет о звуковом символе — добавить тот или иной эффект.

Как правило, динамика в Flash-фильмах обеспечивается за счет того, что в течение некоторого интервала времени изменяются те или иные свойства экземпляра (например, координаты, цвет, размер, прозрачность и т. д.), то есть изменяется состояние экземпляра. С каждым состоянием экземпляра связан отдельный кадр фильма (Frame). Кадр, соответствующий изменению состояния экземпляра, называется ключевым кадром (Keyframe). Ключевой кадр сам рассматривается как объект соответствующего типа, свойства которого пользователь может изменять. Для ключевых кадров предусмотрены специальные функции и команды, предназначение которых мы поясним немного позже.

Динамика смены кадров фильма описывается с помощью временной диаграммы (Timeline). В качестве параметров временной диаграммы можно указывать частоту смены кадров, моменты начала и завершения движения объектов и т. д.

В фильме может использоваться несколько различных объектов. Состояние каждого из них может изменяться независимо от других, либо вообще оставаться неизменным (если, например, некоторый объект используется в качестве фона). Чтобы упростить описание поведения различных элементов фильма, каждый из них помещается, как правило, на отдельный слой (Layer). Разработчики Flash для пояснения роли слоев в фильме сравнивают их с листами прозрачной кальки. Собрав воедино «стопку» таких листов, вы можете получить некую сцену, на которой действуют различные «персонажи». Кстати о сценах. Сцена (Scene) - это еще один термин, используемый при работе с Flash. Каждая сцена представляет собой определенное сочетание слоев. Для простых фильмов бывает достаточно создать и описать одну-единственную сцену, содержащую один слой. Для более сложных может потребоваться создать несколько различных сцен. Переход от одной сцены к другой определяется уже не временной диаграммой, а несколько иным механизмом. В простейшем случае сцены фильма выполняются последовательно, в соответствии с их порядковыми номерами. Для более сложного построения фильма используются средства языка ActionScript.

При создании сложных фильмов достаточно важную роль играет еще одно понятие — клип (Clip, или Movie clip). Клип — это специальный тип символа. Он представляет собой как бы мини-фильм, для которого создается собственная временная диаграмма и устанавливаются собственные параметры (например, частота смены кадров). Клип, как и любой другой элемент фильма, может быть включен в библиотеку символов для многократного использования в фильме. Каждому экземпляру клипа может быть назначено собственное имя.

Любой элемент фильма может быть использован и внутри клипа. Разрешается также создавать «вложенные» клипы. Если требуется описать некоторые дополнительные условия активизации клипа внутри фильма, то для

этого могут быть использованы средства языка ActionScript. В состав клипа могут включаться также интерактивные элементы (например, кнопки).

При запуске большинства Windows-приложений автоматически создается «заготовка» (пустой документ, бланк) того типа, для работы с которым предназначено данное приложение. Например, при открытии окна текстового процессора Word в нем появляется пустой документ, в который вы можете ввести требуемый текст, добавить графику и т. д. Для нового документа используются параметры, установленные по умолчанию (размер и стиль шрифта, ширина полей и т. п.). Чтобы документ выглядел так, как хотели бы именно вы, целесообразно установить требуемые значения соответствующих параметров.

Подобным образом организована и работа с редактором Flash. При его запуске автоматически создается «заготовка» будущего фильма (точнее, его первой сцены). Если это необходимо, автор может установить собственные значения общих свойств фильма, таких как частота смены кадров, размеры окна, используемого для отображения фильма, цвет фона, и некоторые другие.

После установки общих параметров фильма можно переходить к формированию первой сцены фильма. Решив, какие именно объекты должны присутствовать на этой сцене, вы последовательно помещаете каждый из объектов на новый слой. Установка параметров слоя — это один из наиболее ответственных этапов построения фильмов. Поскольку в фильме слои (или помещенные на них объекты) тем или иным образом взаимодействуют друг с другом, после описания очередного слоя может оказаться, что требуется отредактировать созданные ранее. В этом нет ничего страшного, это вполне естественно, и Flash предоставляет автору возможность скорректировать предыдущие действия.

Очередной объект, подлежащий включению в фильм, может быть либо взят из библиотеки символов, либо создан заново.

Если объект пока существует только в вашем воображении или на бумаге, то Flash превращается в графический редактор, предоставляющий все основные инструменты для рисования, закрашивания, ретуширования и комбинирования

различных изображений. При этом как отдельные компоненты создаваемого объекта, так и законченные изображения могут быть импортированы из других графических форматов.

Те объекты, которые предполагается использовать в фильме неоднократно (хотя бы дважды), целесообразно сразу создавать как новые символы.

2.3. Задание на лабораторную работу

Используя возможности автоматической анимации Macromedia Flash создать анимированный ролик из 15-20 кадров на произвольную тему. Обязательно должны быть использованы функции motion tween и shape tween.

3. РАЗРАБОТКА ГИПЕРТЕКСТОВЫХ СТРАНИЦ ПРИ ПОМОЩИ HTML

3.1. Общие сведения

Термин HTML (HyperText Markup Language) означает "язык маркировки гипертекстов". Первую версию HTML разработал сотрудник Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.

Со времени создания первой версии HTML претерпел некоторые изменения. Как и многое другое в компьютерном мире, версии, или спецификации, HTML оказались пронумерованными. Известны спецификации 2.0, 3.0, 3.2 и 4.0. Текущую спецификацию HTML всегда можно найти на сервере W3C (<http://www.w3.org/>).

HTML-документ — это просто текстовый файл с расширением *.html (Unix-системы могут содержать файлы с расширением *.html). Вот самый простой HTML-документ:

```

<html>
  <head>
    <title>
      Пример 1
    </title>
  </head>
  <body>
    <H1>
      Привет!
    </H1>
    <P>
      Это простейший пример HTML-документа.
    </P>
    <P>
      Этот *.html-файл может быть
      одновременно открыт и в Notepad, и в Netscape.
      Сохранив изменения в Notepad, просто нажмите
      кнопку Reload ('перезагрузить') в Netscape,
      чтобы увидеть эти изменения реализованными
      в HTML-документе.
    </P>
  </body>
</html>

```

Для удобства чтения введены дополнительные отступы, однако в HTML это совсем не обязательно. Более того, браузеры просто игнорируют символы конца строки и множественные пробелы в HTML-файлах.

Как видно из примера, вся информация о форматировании документа сосредоточена в его фрагментах, заключенных между знаками "<" и ">". Такой фрагмент (например, <html>) называется меткой (по-английски — tag, читается "тэг").

Большинство HTML-меток — парные, то есть на каждую открывающую метку вида <tag> есть закрывающая метка вида </tag> с тем же именем, но с добавлением "/".

Метки можно вводить как большими, так и маленькими буквами. Например, метки <body>, <BODY> и <Body> будут восприняты браузером одинаково.

Многие метки, помимо имени, могут содержать атрибуты — элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку. В нашем простейшем документе, нет ни одного атрибута. Работу с атрибутами мы рассмотрим в следующем разделе.

Обязательные метки

```
<html> ... </html>
```

Метка `<html>` должна открывать HTML-документ. Аналогично, метка `</html>` должна завершать HTML-документ.

```
<head> ... </head>
```

Эта пара меток указывает на начало и конец заголовка документа. Помимо наименования документа (см. описание метки `<title>` ниже), в этот раздел может включаться множество служебной информации, о которой мы обязательно поговорим чуть позже.

```
<title> ... </title>
```

Все, что находится между метками `<title>` и `</title>`, толкуется браузером как название документа. Netscape Navigator, например, показывает название текущего документа в заголовке окна и печатает его в левом верхнем углу каждой страницы при выводе на принтер. Рекомендуется название не длиннее 64 символов.

```
<body> ... </body>
```

Эта пара меток указывает на начало и конец тела HTML-документа, каковое тело, собственно, и определяет содержание документа.

```
<h1> ... </h1> — <h6> ... </h6>
```

Метки вида `<hi>` (где i — цифра от 1 до 6) описывают заголовки шести различных уровней. Заголовок первого уровня — самый крупный, шестого уровня, естественно — самый мелкий.

```
<p> ... </p>
```

Такая пара меток описывает абзац. Все, что заключено между `<p>` и `</p>`, воспринимается как один абзац.

Метки `<hi>` и `<p>` могут содержать дополнительный атрибут `ALIGN` (читается "элайн", от английского "выравнивать"), например:

```
<h1 ALIGN=CENTER>Выравнивание заголовка  
по центру</h1>
```

или

```
<P ALIGN=RIGHT>Образец абзаца с выравниванием  
по правому краю</P>
```

Подытожим все, что мы знаем, с помощью примера 2:

```
<html>  
<head>  
<title>Пример 2</title>  
</head>  
<body>  
<H1 ALIGN=CENTER>Привет!</H1>  
<H2>Это чуть более сложный пример  
HTML-документа</H2>  
<P>Теперь мы знаем, что абзац можно выравнивать  
не только влево, </P>  
<P ALIGN=CENTER>но и по центру</P>  
<P ALIGN=RIGHT>или по правому краю.</P>  
</body>  
</html>
```

Есть метки, которые не подчиняются двум основным правилам HTML: все они непарные, а некоторые (так называемые &-последовательности) к тому же должны вводиться только маленькими буквами.

```
<BR>
```

Эта метка используется, если необходимо перейти на новую строку, не прерывая абзаца. Очень удобно при публикации стихов (см. пример 3).

```
<html>  
<head>  
<title>Пример 3</title>  
</head>  
<body>  
<H1>Стих</H1>  
<H2>Автор неизвестен</H2>  
<P>Однажды в студеную зимнюю пору<BR>  
Сижу за решеткой в темнице сырой.<BR>  
Гляжу - поднимается медленно в гору<BR>
```

```
Вскормленный в неволе орел молодой.</P>
<P>И шествуя важно, в спокойствии чинном,<BR>
Мой грустный товарищ, махая крылом,<BR>
В больших сапогах, в полушубке овчинном,<BR>
Кровавую пищу клюет под окном.</P>
</body> </html>

<HR>
```

Метка `<HR>` описывает вот такую горизонтальную линию:

Метка может дополнительно включать атрибуты `SIZE` (определяет толщину линии в пикселах) и/или `WIDTH` (определяет размах линии в процентах от ширины экрана). В примере 4 приведена небольшая коллекция горизонтальных линий.

```
<html>
<head>
<title>Пример 4</title>
</head>
<body>
<H1>Коллекция горизонтальных линий</H1>
<HR SIZE=2 WIDTH=100%><BR>
<HR SIZE=4 WIDTH=50%><BR>
<HR SIZE=8 WIDTH=25%><BR>
<HR SIZE=16 WIDTH=12%><BR>
</body>
</html>
```

&-последовательности

Поскольку символы "`<`" и "`>`" воспринимаются браузерами как начало и конец HTML-меток, возникает вопрос: а как показать эти символы на экране? В HTML это делается с помощью &-последовательностей (их еще называют символьными объектами или эскейп-последовательностями). Браузер показывает на экране символ "`<`", когда встречает в тексте последовательность `<`; (по первым буквам английских слов *less than* — меньше, чем). Знак "`>`" кодируется последовательностью `>`; (по первым буквам английских слов *greater than* — больше, чем).

Символ "&" (амперсанд) кодируется последовательностью `&`;

Двойные кавычки (") кодируются последовательностью `"`;

Помните: точка с запятой — обязательный элемент `&`-последовательности.

Кроме того, все буквы, составляющие последовательность, должны быть в нижнем регистре (т.е., маленькие). Использование меток типа `"` или `&` не допускается.

Вообще говоря, `&`-последовательности определены для всех символов из второй половины ASCII-таблицы (куда, естественно, входят и русские буквы). Дело в том, что некоторые серверы не поддерживают восьмибитную передачу данных, и поэтому могут передавать символы с ASCII-кодами выше 127 только в виде `&`-последовательностей.

Комментарии

Браузеры игнорируют любой текст, помещенный между `<!--` и `-->`. Это удобно для размещения комментариев.

```
<!-- Это комментарий -->
```

Форматирование шрифта

HTML допускает два подхода к шрифтовому выделению фрагментов текста. С одной стороны, можно прямо указать, что шрифт на некотором участке текста должен быть жирным или наклонным, то есть изменить физический стиль текста. С другой стороны, можно пометить некоторый фрагмент текста как имеющий некоторый отличный от нормального логический стиль, оставив интерпретацию этого стиля браузеру. Поясним это на примерах.

Физические стили

Под физическом стилем принято понимать прямое указание браузеру на модификацию текущего шрифта. Например, все, что находится между метками `` и ``, будет написано жирным шрифтом. Текст между метками `<I>` и `</I>` будет написан наклонным шрифтом.

Несколько особняком стоит пара меток `<TT>` и `</TT>`. Текст, размещенный между этими метками, будет написан шрифтом, имитирующим пишущую машинку, то есть имеющим фиксированную ширину символа.

Логические стили

При использовании логических стилей автор документа не может знать заранее, что увидит на экране читатель. Разные браузеры толкуют одни и те же метки логических стилей по-разному. Некоторые браузеры игнорируют некоторые метки вообще и показывают нормальный текст вместо выделенного логическим стилем. Вот самые распространенные логические стили.

```
<EM> ... </EM>
```

От английского *emphasis* — акцент.

```
<STRONG> ... </STRONG>
```

От английского *strong emphasis* — сильный акцент.

```
<CODE> ... </CODE>
```

Рекомендуется использовать для фрагментов исходных текстов.

```
<SAMP> ... </SAMP>
```

От английского *sample* — образец. Рекомендуется использовать для демонстрации образцов сообщений, выводимых на экран программами.

```
<KBD> ... </KBD>
```

От английского *keyboard* — клавиатура. Рекомендуется использовать для указания того, что нужно ввести с клавиатуры.

```
<VAR> ... </VAR>
```

От английского *variable* — переменная. Рекомендуется использовать для написания имен переменных.

Пример

Подытожим наши знания о логических и физических стилях с помощью примера 5. Заодно вы сможете увидеть, как ваш браузер показывает те или иные логические стили.

```
<html>
<head>
<title>Пример 5</title>
</head>
<body>
<h1>Шрифтовое выделение фрагментов текста</h1>
<p>Теперь мы знаем, что фрагменты текста можно
```

выделять
жирным или <I>наклонным</I> шрифтом.
Кроме того, можно
включать в текст фрагменты с фиксированной шириной
символа
<TT>(имитация пишущей машинки)</TT></P>
<P>Кроме того, существует ряд логических стилей:</P>
<P>EM - от английского emphasis - акцент

STRONG - от английского strong emphasis -
сильный акцент

<CODE>CODE - для фрагментов
исходных текстов</CODE>

<SAMP>SAMP - от английского sample -
образец </SAMP>

<KBD>KBD - от английского keyboard -
клавиатура</KBD>

<VAR>VAR - от английского variable -
переменная </VAR></P>
</body>
</html>

3.2. Организация текста внутри документа

HTML позволяет определять внешний вид целых абзацев текста. Абзацы можно организовывать в списки, выводить их на экран в отформатированном виде, или увеличивать левое поле.

**Ненумерованные списки: ... **

Текст, расположенный между метками и , воспринимается как ненумерованный список. Каждый новый элемент списка следует начинать с метки . Например, чтобы создать вот такой список:

Иван;
Данила;
белая кобыла

необходим вот такой HTML-текст:

```
<UL>
<LI>Иван;
<LI>Данила;
<LI>белая кобыла
</UL>
```

Обратите внимание: у метки `` нет парной закрывающей метки.

Нумерованные списки: `` ... ``

Нумерованные списки устроены точно так же, как ненумерованные, только вместо символов, выделяющих новый элемент, используются цифры. Если слегка модифицировать наш предыдущий пример:

```
<OL>
<LI>Иван;
<LI>Данила;
<LI>белая кобыла
</OL>
```

получится вот такой список:

1. Иван;
2. Данила;
3. белая кобыла

Списки определений: `<DL>` ... `</DL>`

Список определений несколько отличается от других видов списков. Вместо меток `` в списках определений используются метки `<DT>` (от английского *definition term* — определяемый термин) и `<DD>` (от английского *definition definition* — определение определения). Разберем это на примере. Допустим, у нас имеется следующий фрагмент HTML-текста:

```
<DL>
<DT>HTML
<DD>Термин HTML (HyperText Markup Language) означает
'язык маркировки гипертекстов'. Первую версию HTML
разработал сотрудник Европейской лаборатории физики
элементарных частиц Тим Бернерс-Ли.
<DT>HTML-документ
```

```
<DD>Текстовый файл с расширением *.html
(Unix-системы могут содержать файлы
с расширением *.html1).
</DL>
```

Этот фрагмент будет выведен на экран следующим образом:

HTML

Термин HTML (HyperText Markup Language) означает 'язык маркировки гипертекстов'. Первую версию HTML разработал сотрудник Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.

HTML-документ

```
Текстовый файл с расширением *.html (Unix-системы могут
содержать файлы с расширением *.html1).
```

Обратите внимание: точно так же, как метки , метки <DT> и <DD> не имеют парных закрывающих меток.

Если определяемые термины достаточно коротки, можно использовать модифицированную открывающую метку <DL COMPACT>. Например, вот такой фрагмент HTML-текста:

```
<DL COMPACT>
<DT>А
<DD>Первая буква алфавита
<DT>Б
<DD>Вторая буква алфавита
<DT>В
<DD>Третья буква алфавита
</DL>
```

будет выведен на экран примерно так:

```
А    Первая буква алфавита
Б    Вторая буква алфавита
В    Третья буква алфавита
```

Вложенные списки

Элемент любого списка может содержать в себе целый список любого вида. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками все же не следует.

Вложенные списки очень удобны при подготовке разного рода планов и оглавлений.

Наши знания о списках можно вкратце свести в пример 6:

```
<html>
<head>
<title>Пример 6</title>
</head>
<body>
<H1>HTML поддерживает несколько видов списков </H1>
<DL>
<DT>Ненумерованные списки
<DD>Элементы ненумерованного списка выделяются
    специальным
символом и отступом слева:
<UL>
<LI>Элемент 1
<LI>Элемент 2
<LI>Элемент 3
</UL>
<DT>Нумерованные списки
<DD>Элементы нумерованного списка выделяются
отступом слева, а
также нумерацией:
<OL>
<LI>Элемент 1
<LI>Элемент 2
<LI>Элемент 3
</OL>
<DT>Списки определений
<DD>Этот вид списков чуть сложнее, чем два предыдущих,
```

```
но и выглядит более эффектно.  
<P>Помните, что списки можно встраивать один в другой,  
но не  
следует закладывать слишком много уровней вложенности.  
</P>  
<P>Обратите внимание, что внутри элемента списка  
может находиться  
несколько абзацев. Все абзацы при этом будут иметь  
одинаковое левое поле. </P>  
</DL>  
</body>  
</html>
```

Форматированный текст: <PRE> ... </PRE>

В самом начале мы говорили о том, что браузеры игнорируют множественные пробелы и символы конца строки. Из этого правила, однако, есть исключение.

Текст, заключенный между метками <PRE> и </PRE> (от английского *preformatted* — предварительно форматированный), выводится браузером на экран как есть — со всеми пробелами, символами табуляции и конца строки. Это очень удобно при создании простых таблиц.

Текст с отступом: <BLOCKQUOTE> ... </BLOCKQUOTE>

Текст, заключенный между метками <BLOCKQUOTE> и </BLOCKQUOTE>, выводится браузером на экран с увеличенным левым полем.

Связывание

В HTML переход от одного фрагмента текста к другому задается с помощью метки вида:

```
<A HREF="[адрес перехода]">  
выделенный фрагмент текста</A>
```

В качестве параметра [адрес перехода] может использоваться несколько типов аргументов. Самое простое — это задать имя другого HTML-документа, к которому нужно перейти. Например:

```
<A HREF="index.html">Перейти к оглавлению</A>
```

Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента `Перейти к оглавлению`, при нажатии на который в текущее окно будет загружен документ `index.html`.

Обратите внимание: если в адресе перехода не указан каталог, переход будет выполнен внутри текущего каталога. Если в адресе перехода не указан сервер, переход будет выполнен на текущем сервере.

Из этого следует одно очень важное практическое соображение. Если вы подготовили к публикации некоторую группу HTML-документов, которые ссылаются друг на друга только по имени файла и находятся в одном каталоге на вашем компьютере, вся эта группа документов будет работать точно так же, если ее поместить в любой другой каталог на любом другом компьютере, на локальной сети или на Интернет. Таким образом, у Вас появляется возможность разрабатывать целые коллекции документов без подключения к Интернет, и только после окончательной готовности, подтвержденной испытаниями, помещать коллекции документов на Интернет целиком.

На практике, однако, часто бывает необходимо дать ссылку на документ, находящийся на другом сервере. Например, если вы хотите дать ссылку на это руководство со своей странички, Вам придется ввести в свой HTML-документ примерно такой фрагмент:

```
<A HREF="http://www.tpu.ru/CE/Kudinov/  
index.html">
```

```
Практическое руководство по HTML </A>
```

При необходимости можно задать переход не просто к некоторому документу, но и к определенному месту внутри этого документа. Для этого необходимо создать в документе, к которому будет задан переход, некоторую опорную точку, или анкер. Разберем это на примере.

Допустим, что необходимо осуществить переход из файла `1.html` к словам "Переход закончен" в файле `2.html` (файлы находятся в одном каталоге). Прежде всего, необходимо создать вот такой анкер в файле `2.html`:

```
<A NAME="AAA">Переход закончен</A>
```

Слова "Переход закончен" при этом никак не будут выделены в тексте документа.

Затем в файле 1.html (или в любом другом) можно определить переход на этот анкер:

```
<A HREF="2.html#AAA">Переход к анкеру AAA</A>
```

Кстати говоря, переход к этому анкеру можно определить и внутри самого документа 2.html — достаточно только включить в него вот такой фрагмент:

```
<A HREF="#AAA">Переход к анкеру AAA</A>
```

На практике это очень удобно при создании больших документов. В начале документа можно поместить оглавление, состоящее из ссылок на анкеры, расположенные в заголовках разделов документа.

Во избежание недоразумений рекомендуется задавать имена анкерov латинскими буквами. Следите за написанием имен анкерov: большинство браузеров отличают большие буквы от маленьких. Если имя анкера определено как AAA, ссылка на анкер aaa или AaA не выведет Вас на анкер AAA, хотя документ, скорее всего, будет загружен корректно.

Пока что мы обсуждали только ссылки на HTML-документы. Однако возможны ссылки и на другие виды ресурсов:

```
<A HREF="ftp://server/directory/file.ext">
```

```
Выгрузить файл</A>
```

Такая ссылка, если ей воспользоваться, запустит протокол передачи файлов и начнет выгрузку файла file.ext, находящегося в каталоге directory на сервере server, на локальный диск пользователя.

```
<A HREF="mailto:user@mail.box">Послать письмо</A>
```

Если пользователь совершит переход по такой ссылке, у него на экране откроется окно ввода исходящего сообщения его почтовой программы. В строке To: ("Куда") окна почтовой программы будет указано user@mail.box.

Разберем все, что мы знаем о связывании, с помощью примера 7.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Пример 7</TITLE>
```

```
</HEAD>
<BODY>
<H1>Связывание </H1>
<P>С помощью ссылок можно переходить к другим файлам
(например, к <A HREF="index.html">оглавлению этого
руководства</A>).</P>
<P>Можно выгружать файлы (например,
<A HREF="ftp://yi.com/home/ChuvakhinNikolai/
html-pr.doc">это руководство в формате Microsoft
Word 2.0</A>) по FTP.</P>
<P>Можно дать пользователю возможность послать почту
(например, <A HREF="mailto:nc@iname.com">
автору этого руководства</A>).</P>
</BODY>
</HTML>
```

Изображения в HTML-документе

Встроить изображение в HTML-документ очень просто. Для этого нужно только иметь это самое изображение в формате GIF (файл с расширением *.gif) или JPEG (файл с расширением *.jpg или *.jpeg) и одну строчку в HTML-тексте.

Допустим, нам нужно включить в документ изображение, записанное в файл picture.gif, находящийся в одном каталоге с HTML-документом. Тогда строчка будет вот такая:

```
<IMG SRC="picture.gif">
```

Метка может также включать атрибут ALT="[текст]", например:

```
<IMG SRC="picture.gif" ALT="Картинка">
```

Встретив такую метку, браузер покажет на экране текст Картинка и начнет загружать на его место картинку из файла picture.gif. Атрибут ALT может оказаться необходимым для старых браузеров, которые не поддерживают изображений, а также на случай, если у браузера отключена автоматическая

загрузка изображений (при медленном подключении к Интернет это делается для экономии времени).

Файл, содержащий изображение, может находиться в другом каталоге или даже на другом сервере. В этом случае стоит указать его полное имя.

Разберем все, что мы знаем об изображениях, с помощью примера 8.

```
<HTML>
<HEAD>
<TITLE>Пример 8</TITLE>
</HEAD>
<BODY>
<H1>Изображения </H1>
<P>Изображение можно встроить очень просто: </P>
<P><IMG SRC="picture.gif"></P>
<P>Кроме того, изображение можно сделать "горячим",
  то есть осуществлять переход при нажатии на
  изображение:</P>
<P><A HREF="index.html"><IMG SRC="picture.gif">
</A></P>
</BODY>
</HTML>
```

Обратите внимание на вторую часть примера. Если ссылка на изображение находится между метками `` и ``, изображение фактически становится кнопкой, при нажатии на которую происходит переход по ссылке.

Атрибуты определяющие цвет

Цветовая гамма HTML-документа определяется атрибутами, размещенными внутри метки `<BODY>`. Вот список этих атрибутов:

`bgcolor`

Определяет цвет фона документа.

`text`

Определяет цвет текста документа.

`link`

Определяет цвет выделенного элемента текста, при нажатии на который происходит переход по гипертекстовой ссылке.

```
vlink
```

Определяет цвет ссылки на документ, который уже был просмотрен ранее.

```
alink
```

Определяет цвет ссылки в момент, когда на нее указывает курсор мыши и нажата ее правая кнопка, то есть непосредственно перед переходом по ссылке.

Цвет кодируется последовательностью из трех пар символов. Каждая пара представляет собой шестнадцатиричное значение насыщенности заданного цвета одним из трех основных цветов (красным, зеленым и синим) в диапазоне от нуля (00) до 255 (FF). Разберем несколько примеров.

```
bgcolor=#FFFFFF
```

Цвет фона. Насыщенность красным, зеленым и синим одинакова — FF (это шестнадцатиричное представление числа 255). Результат — белый цвет.

```
text=#000000
```

Цвет текста. Насыщенность красным, зеленым и синим одинакова — 00 (ноль). Результат — черный цвет.

```
link=#FF0000
```

Цвет гипертекстовой ссылки. Насыщенность красным — FF (255), зеленым и синим — 00 (ноль). Результат — красный цвет.

Кроме того, метка <BODY> может включать атрибут `background="[имя файла]"`, который задает изображение, служащее фоном для текста и других изображений. Как и любое другое изображение, фон должен быть представлен в формате GIF (файл с расширением *.gif) или JPEG (файл с расширением *.jpg или *.jpeg).

Браузеры заполняют множественными копиями изображения-фона все пространство окна, в котором открыт документ, подобно тому, как при строительстве большие пространства стен покрывают маленькими (и одинаковыми) плитками.

Важно отметить, что цвет фона и изображение-фон никак не отображаются на бумаге при выводе HTML-документа на печать. Из этого есть одно важное практическое следствие: старайтесь не использовать текст белого цвета.

3.3. Задание на лабораторную работу

Собранный материал по теме, выбранной в ходе лабораторной работы № 1, необходимо оформить в виде раздела гипертекстового учебника. Объем не менее 5 html-страниц. При создании не использовать быстрых средств разработки web-сайтов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Базовый курс PowerPoint. Изучаем Microsoft Office. Минск, Современная школа, 2007, 48 с.
2. Иллюстрированный самоучитель по Flash MX – http://www.books.everonit.ru/Flash_MX/index.html (01.07.2007)
3. Николай Чувахин. Практическое руководство по HTML – <http://ru.ncbase.com/net/pr.htm> (01.07.2007)

Антон Викторович Кудинов

МУЛЬТИМЕДИА ТЕХНОЛОГИИ.

ЦИКЛ ЛАБОРАТОРНЫХ РАБОТ.

Методические указания к лабораторным работам

Подписано к печати

Формат 60x84/16. Бумага офсетная.

Плоская печать. Усл. печ. л. . Уч.-изд. л. .

Тираж экз. Заказ . Цена свободная.

Издательство ТПУ. 634050, Томск, пр. Ленина, 30.