

Macromedia Flash MX

Э К С П Р Е С С
К У Р С



- ▶ Инструменты рисования
- ▶ Трансформация и импорт графики
- ▶ Шрифты и текстовые блоки
- ▶ Анимация и звук
- ▶ Интерактивные сценарии ActionScript

Владимир Дронов

Macromedia

Flash MX

Э К С П Р Е С С -
К У Р С

Санкт-Петербург

«БХВ-Петербург»

2003

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Macromedia Flash MX. Экспресс-курс. — СПб.: БХВ-Петербург,
2003. — 352 с.: ил.
ISBN 5-94157-325-1

В книге описываются возможности пакета интернет-графики Macromedia Flash MX. На примере создания Web-сайта музыкального интернет-магазина рассматриваются современные средства Web-дизайна: векторная и растровая графика, анимация, интерактивные элементы, написание сценариев на Языке ActionScript, использование звука и цвета. Навыки и знания, полученные из этой книги, помогут Вам при разработке собственного Web-сайта осуществить самые смелые творческие замыслы.

Для широкого круга пользователей

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Светлана Гатилова</i>
Компьютерная верстка	<i>Екатерины Трубниковой</i>
Корректор	<i>Виктория Голуб</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 26.05.03.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 28,38.

Тираж 5000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-325-1

© Дронов В. А., 2003
© Оформление, издательство "БХВ-Петербург", 2003

Содержание

Предисловие	1
Введение	3
Как создаются Web-страницы?	3
Две разновидности компьютерной графики	7
Как работает Flash	8
Что дальше?	9
Часть I. Статичная графика	13
Глава 1. Введение во Flash	15
Создание нового документа	16
Создание пустого документа	16
Среда Flash	21
Главное окно программы	21
Инструментарии	23
Панели	25
Работа в окне документа Flash	28
Окно документа	28
Как создается графика Flash	30
Управление окном документа	32
Работа с интерактивной справкой Flash MX	38
Что дальше?	40
Глава 2. Рисование простейших изображений	41
Инструменты рисования	41
Базовые средства управления цветом	42
Линия	42
Прямоугольник и эллипс	44
Перо	46
Ведро с краской	49
Карандаш	51
Кисть	53
Работа с нарисованной графикой	55
Выделение графики	55
Фрагментация и слияние графики	56
Группировка	58

Правка графики.....	59
Что дальше?	69
Глава 3. Трансформация графики	71
Простейшие трансформации.....	71
Выравнивание и распределение	71
Перемещение и изменение размеров.....	75
Зеркальное отражение.....	76
Более сложные трансформации.....	77
Вращение и сдвиг	77
Изменение размеров.....	79
Искажение формы	80
Деформация	81
Свободная трансформация	82
Что дальше?	83
Глава 4. Использование слоев.....	85
Создание слоев и управление ими	86
Использование папок для организации слоев	89
Что дальше?	91
Глава 5. Работа с цветом	93
Инструменты выбора цвета	93
Использование селектора цвета	93
Область Colors главного инструментария	95
Создание новых цветов. Смеситель.....	96
Работа с линиями.....	98
Задание цвета линии. Пузырек с чернилами	99
Задание толщины и стиля линий.....	99
Работа с заливками	100
Задание цвета заливки.....	101
Создание сложных заливок	101
Пипетка	108
Работа с палитрами	109
Что дальше?	111
Глава 6. Работа с текстовыми блоками.....	113
Текстовые блоки и работа с ними.....	113
Создание текстовых блоков.....	114
Форматирование текста	116
Параметры текстового блока.....	121
Трансформация текстовых блоков.....	122
Специальные текстовые блоки.....	123
Поддержка шрифтов во Flash	124
Что дальше?	126
Глава 7. Импорт растровой графики.....	127
Импорт растровой графики	127
Настройка параметров растрового изображения	129
Что дальше?	131

Глава 8. Использование образцов и библиотек	133
Создание образцов и экземпляров	134
Создание нового образца	134
Создание экземпляров на основе образца	137
Трансформации экземпляров	137
Смена экземпляра	140
Преобразование экземпляра в обычный графический элемент	141
Работа с образцами	141
Панель Library	141
Изменение образцов	143
Дублирование образцов	144
Управление образцами	146
Использование папок для упорядочения образцов	147
Копирование образцов из документа в документ	150
Растровые образцы	150
Библиотеки общего использования	151
Что дальше?	151
Часть II. Анимация и звук	153
Глава 9. Создание простейшей анимации	155
Основы анимации	155
Создание движения	157
Простейшее движение	157
Более сложное движение	160
Другие разновидности движения	164
Создание морфинга	167
Типичный случай морфинга	167
Маркеры морфинга и их использование	170
Дополнительные возможности работы с анимацией	173
Работа с кадрами	173
Работы с временной шкалой	175
Использование сцен	177
Что дальше?	180
Глава 10. Создание составной анимации	181
Создание составной анимации	182
Что дальше?	184
Глава 11. Создание кнопок	185
Что дальше?	189
Глава 12. Использование слоев для создания анимации	191
Направляющие слои	191
Маскирующие слои	194
Что дальше?	198
Глава 13. Работа со звуком	199
Импорт звука	199
Использование звуков	200

Создание звуков-сигналов	200
Создание фонового звука	202
Задание параметров проигрывания звука	203
Правка звука средствами Flash	204
Задание параметров экспорта звука	207
Что дальше?	209
Часть III. Интерактивность	211
Глава 14. Написание простейших сценариев	213
Как выполняются сценарии. События	214
Создание сценариев Flash	214
Привязка сценариев к кадрам	214
Знакомство с панелью Actions	216
Привязка сценариев к клипам	219
Привязка сценариев к кнопкам	222
Работа с панелью Actions	224
Два режима панели Actions	224
Дополнительные функции панели Actions	225
Что дальше?	229
Глава 15. Язык ActionScript	231
Типографские соглашения	231
Начала ActionScript	232
Основные понятия ActionScript	232
Данные AconSticript	234
Переменные	235
Операторы	236
Комментарии	242
Сложные выражения	243
Блочные выражения	243
Условные выражения	243
Выражения выбора	246
Циклы	247
Функции	250
Создание функций	251
Вызов функций	252
Встроенные функции ActionScript	253
Массивы	253
Объекты	254
Понятия объекта и экземпляра	254
Работа с объектами	255
Встроенные объекты ActionScript	256
Что дальше?	260
Глава 16. Работа с объектами Flash	261
Объект <i>movieClip</i>	261
Имена и пути доступа к клипам	262
Наложение клипов	265

Создание и удаление клипов	266
Изменение параметров встроенных клипов	269
Управление воспроизведением клипа	271
Загрузка и выгрузка клипов	272
Объект <i>Color</i>	273
Методы объекта <i>Color</i>	273
Задание цветовых преобразований	274
Объект <i>Sound</i>	276
Свойства и методы объекта <i>Sound</i>	276
Задание звуковых трансформаций	277
Загрузка и выгрузка звуков	278
Использование таймеров	279
Динамические текстовые блоки	281
Создание динамического текстового блока	281
Объект <i>TextField</i>	285
Создание текстовых блоков с помощью сценария	287
Поля ввода	287
Кнопки. Объект <i>Button</i>	289
Взаимодействие с Web-обозревателем	290
Что дальше?	292
Глава 17. Публикация и экспорт	293
Публикация	294
Задание комплекта публикации	294
Настройки форматов публикации	296
Предварительный просмотр публикуемой графики	307
Экспорт	307
Выполнение экспорта графики	308
Создание остальных Web-страниц сайта	313
Заключение	317
Приложение. Клавиатурные комбинации Macromedia Flash MX	321
Предметный указатель	329

Предисловие

Уважаемый читатель! Знаете ли вы, что такое Интернет? Да что это за глупый вопрос, конечно, знаете. Кто в наше время не знает, что такое Интернет! Статистика утверждает, что каждый пятый житель России хотя бы время от времени пользуется Всемирной паутиной, т. е. процент весьма велик, а значит, вы с большой степенью вероятности в него попадете.

Вы уже не раз путешествовали по сайтам в поисках нужной информации или просто от нечего делать. И вы наверняка видели потрясающий красоты рекламные баннеры, заманивающие на никому не нужные сайты. Вы наверняка видели мультфильмы о похождениях Масьни — нового феномена русской культуры. Вероятно, вы играли в онлайн-игры — их тоже хватает в сети. Ну и, конечно, должны были видеть на Web-страницах шикарную графику с красивыми кнопками и умопомрачительной анимацией.

Значит, вы все это видели? Ну, и как вам все это? Впечатляет? Да уж, Web-дизайнеры (люди, создающие Web-страницы) постарались на славу.

Но как они это сделали?

Есть такая программа — Macromedia Flash MX. Она-то и позволяет создавать все эти красоты. А если говорить сухим техническим языком, то это пакет интернет-графики.

Об этой-то программе и пойдет речь в данной книге.

Но зачем, спросите вы, нужно было выдумывать какой-то там Flash MX? На свете существует бесчисленное множество программ, позволяющих создавать графику для Интернета. И они могут заставить изображение двигаться. Разве что Flash MX обладает какими-то преимуществами, которых нет у других программ...

Есть такие преимущества! Целых три.

1. *Компактность.* Изображение, сохраненное в формате Flash, имеет очень маленькие размеры. Можно создать огромную картинку, заставить ее двигаться, крутиться и вертеться самым причудливым образом, а результирующий файл будет размером в несколько Кбайт. А если он настолько

мал, то и загружаться по сети он будет практически мгновенно! Ведь о высокоскоростных интернет-каналах мы пока что можем только мечтать...

2. *Анимация*. Здесь Flash MX — царь и бог. Никакой другой программе не под силу сделать такую сложную анимацию и при этом не "раздуть" результирующий файл до умопомрачительных размеров.
3. *Интерактивность*. Это значит, что изображение, созданное вами, может откликаться на действия пользователя. Скажем, в ответ на движение мыши может появляться или, наоборот, исчезать какая-либо картинка, а при щелчке кнопки — запускаться фильм. Это достигается написанием особых программ — *сценариев*, которые могут управлять фрагментами изображения Flash и анимацией, а также вызывать другие (внешние по отношению к Flash) программы. Эти сценарии пишутся на особом языке программирования, называемом *ActionScript*. Остальные пакеты интернет-графики... нет, даже не отдыхают, а мертвецки дрыхнут!

Вы не верите, уважаемый читатель? Ну, тогда, чтобы развеять все сомнения, мы с вами сами создадим интернет-сайт с использованием графики Flash. И не надо бояться, что вы не справитесь, — все операции по созданию наших первых изображений мы сделаем вместе, шаг за шагом. А по пути изучим некоторые термины, имеющие хождение среди профессиональных Web-дизайнеров — без этих терминов в мире интернет-графики никуда.

Итак, установите на свой компьютер программу Macromedia Flash MX, если она там не установлена, и приготовьтесь. Начнем...

Введение

Вот мы решили создать свой Web-сайт. Что для этого нужно? Ну, прежде всего, выяснить, для чего же он нам нужен. Этим мы сейчас и займемся.

В настоящее время во всем мире наблюдается бум интернет-торговли. Давайте же и мы создадим сайт интернет-магазина. Этот пример ничем не хуже сайта фирмы "Рога и копыта" или новостного портала "Тьмутаракань online". И продавать этот магазин будет хорошую музыку: арт-рок, прогрессивный рок, психоделический рок, авангард, в общем, то, что невозможно услышать по радио и увидеть по телевизору.

И назовем мы этот магазин "CD-Shop". Это, конечно, не очень оригинально, но в задачу этой книги не входит мастер-класс по придумыванию зубодробительных названий для сайтов. Нам нужно изучить Flash MX и создать с его помощью что-то полезное.

Но прежде чем мы возьмемся за Flash, давайте поговорим о современных интернет-технологиях. Это нам пригодится в дальнейшем.

Как создаются Web-страницы?

Сначала поговорим о том, как создаются Web-страницы, которые вы видите каждый раз, когда выходите в Интернет.

Web-страницы — это обычные текстовые файлы, созданные в стандартном Блокноте или аналогичном простейшем текстовом редакторе. Они содержат текст, являющийся содержимым Web-страницы, а также набор особых команд, задающих форматирование текста или помещающих на страницу изображения. Эти команды называются *тегами*, а набор правил, которым должен подчиняться Web-дизайнер при написании страниц — языком *HTML* (HyperText Markup Language — язык гипертекстовой разметки).

Давайте прямо сейчас создадим небольшую Web-страничку. Для этого запустим Блокнот (или аналогичную программу), наберем в нем приведенный ниже код, проверим его на ошибки и сохраним в файле под именем `Sample.htm`. Только когда будем вводить имя в диалоговом окне сохранения

файла, заключим его в кавычки, иначе Блокнот автоматически добавит к его имени расширение txt, и наш файл получит имя Sample.htm.txt. А это нам совсем не нужно.

Теперь можно открыть полученный файл в Web-обозревателе, для чего достаточно дважды щелкнуть на нем мышью. Мы увидим то, что показано на рис. В.1.

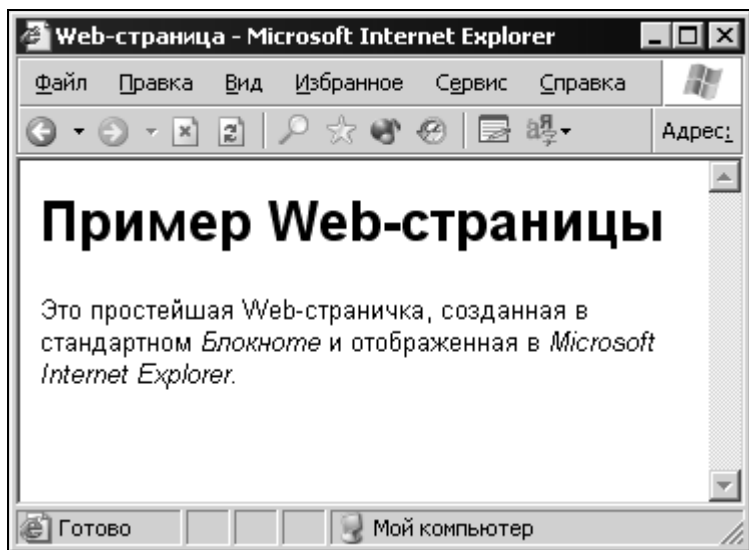


Рис. В.1. Простейшая Web-страничка

Теперь давайте рассмотрим сам HTML-код нашей первой странички.

```
<HTML>
  <HEAD>
    <TITLE>Web-страница</TITLE>
  </HEAD>
  <BODY>
    <H1>Пример Web-страницы</H1>
    <P>Это простейшая Web-страничка, созданная в стандартном
      <I>Блокноте</I>
      и отображенная в <I>Microsoft Internet Explorer</I>.</P>
  </BODY>
</HTML>
```

Выглядит устрашающе... Однако хорошо заметен текст, который отобразится в окне Web-обозревателя на рис. В.1. Давайте рассмотрим его подробнее.

```
<H1>Пример Web-страницы</H1>
```

```
<P>Это простейшая Web-страничка, созданная в стандартном <I>Блокноте</I>  
и отображенная в <I>Microsoft Internet Explorer</I>.</P>
```

Слова, заключенные в угловые скобки "<" и ">", и есть теги HTML. Они задают форматирование текста. Скажем, строка `<I>Блокноте</I>` будет выведена курсивом, так как теги `<I>` и `</I>` задают курсивное начертание текста. Причем, тег `<I>` помечает начало курсивного фрагмента (*открывающий* тег), а тег `</I>` — конец (*закрывающий*). А собственно фрагмент, заключенный между открывающим и закрывающим тегами, называется *содержимым* тега.

Еще один пример тега HTML — это `<B>...</B>`. Этот тег позволяет вывести фрагмент текста полужирным шрифтом.

А что же теги `<P>...</P>` и `<H1>...</H1>`? Ну, это совсем просто! Первый тег задает обычный текстовый абзац, а второй тег позволяет создать заголовок. Посмотрите еще раз на рис. В.1 — и вы увидите и заголовок, и абзац.

Здесь мы говорили о так называемых *парных* тегах, когда открывающему тегу соответствует закрывающий. HTML определяет также набор *одинарных* тегов. Один из них — тег вставки графического изображения `<IMG>`.

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

Здесь мы видим, что само графическое изображение хранится в отдельном файле `picture.gif`. Тег `<IMG>` содержит в себе параметр `SRC`, задающий имя файла изображения. (Такие параметры называются *атрибутами* тега.) Web-обозреватель, встретив в HTML-коде страницы тег `<IMG>`, загружает файл, чье имя задано атрибутом `SRC`, и отображает его.

Для связи Web-страниц друг с другом используются *гиперссылки* — особые связи, ведущие от одной Web-страницы к другой. Именно на них мы щелкаем мышью, если хотим перейти на другую страницу. Гиперссылки создаются с помощью особого парного тега `<A>` и имеют следующий вид:

```
<A HREF="http://www.somesite.ru/pages/page125.html">Страница N125</A>
```

Видно, что тег `<A>` содержит атрибут `HREF`, который задает *интернет-адрес* страницы, на которую будет выполнен переход при щелчке на гиперссылке. Этот атрибут должен присутствовать в любом теге `<A>`, задающем гиперссылку (*обязательный* атрибут).

А теперь сделаем небольшой фокус. Изменим код гиперссылки, приведенный выше, таким образом (изменения выделены полужирным шрифтом):

```
<P><A HREF="http://www.somesite.ru/pages/page125.html"  
❏TARGET="_blank">Страница N125</A></P>
```

Мы поместили в тег `<A>` атрибут `TARGET`, задающий *цель гиперссылки*. Цель гиперссылки задает окно Web-обозревателя, в которое будет выведена Web-

страница. Если этому атрибуту присвоено значение `_blank`, страница будет выведена в отдельное окно Web-обозревателя. Чтобы задать обычное поведение гиперссылки (новая страница выводится в то же окно), нужно присвоить атрибуту `TARGET` значение `_self` или вообще убрать его из кода гиперссылки.

В отличие от атрибута `HREF` тега `<A>`, атрибут `TARGET` не является обязательным. Он так и называется — *необязательный* атрибут.

Осталось рассмотреть совсем немного тегов, используемых для служебных целей и не отображаемых Web-обозревателем, так называемых *невидимых* тегов.

Парный тег `<BODY>...</BODY>` используется для выделения *тела* Web-страницы, т. е. той ее части, которая будет отображаться в окне Web-обозревателя. Кроме тела, Web-страница также должна содержать задаваемый парным тегом `<HEAD>...</HEAD>` *HTML-заголовок*, где помещается служебная информация. (Этот заголовок не выводится Web-обозревателем, а используется для внутренних нужд. Не путайте заголовок Web-страницы и обычный текстовый заголовок, задаваемый тегом `<h1>!`) Среди этой служебной информации может быть *название* страницы, отображаемое в заголовке окна Web-обозревателя; оно задается парным тегом `<TITLE>...</TITLE>`.

Имеется также возможность разделить окно Web-обозревателя на несколько небольших "форточек", в каждой из которых может быть отображена отдельная Web-страница. Такие "форточки" называются *фреймами*. Ниже приведен код Web-страницы, задающей набор фреймов.

```
<HTML>
  <HEAD>
    <TITLE>Набор фреймов</TITLE>
  </HEAD>
  <FRAMESET ROWS="100,800">
    <FRAME NAME="header" SRC="Header.html">
    <FRAME NAME="content" SRC="About.html">
  </FRAMESET>
</HTML>
```

Видно, что эта страница не содержит тега `<BODY>`. Вместо него используется тег `<FRAMESET>`, задающий размеры фреймов этого набора. Размер задает атрибут `ROWS`, принимающий в качестве значения список значений, разделенных запятыми. Так, в нашем примере создаются два вертикально расположенных фрейма, имеющие размеры 100 и 800 пикселей соответственно.

Размеры фреймов могут быть заданы в процентах. Например, так:

```
<FRAMESET ROWS="10%,90%">
```

А значение "*" создает фрейм, занимающий всю оставшуюся площадь окна Web-обозревателя:

```
<FRAMESET ROWS="100, *">
```

Если нужно создать набор горизонтально расположенных фреймов, используется атрибут COLS:

```
<FRAMESET COLS="200, *">
```

Тег <FRAME> задает характеристики самого фрейма. Атрибут NAME задает имя фрейма, а атрибут SRC — интернет-адрес Web-страницы, которая будет в него загружена.

Чтобы вывести Web-страницу в какой-либо фрейм, его имя присваивается уже знакомому нам атрибуту TARGET тега <A>:

```
<P><A HREF="http://www.somesite.ru/pages/page125.html"
```

```
  ⚡ TARGET="content">Страница N125</A></P>
```

Так, приведенный выше код загружает страницу page125.html во фрейм content.

Как видите, ничего особо сложного в языке HTML нет. Единственная сложность — это запомнить все его теги, но это вопрос времени и опыта.

Для того чтобы различные программы Web-обозревателей правильно отображали одну и ту же Web-страницу, язык HTML должен быть стандартизирован. Его стандартизацией (а также множеством других стандартов Интернета) занимается особая организация, называемая *World Wide Web Consortium* или, сокращенно, *WWWC*. WWWC издает весьма увесистые труды, описывающие различные версии стандарта HTML. Последняя версия этого языка — 4.01 — вышла в конце 90-х годов прошлого века. Все современные программы Web-обозревателей поддерживают эту версию HTML.

После того, как файлы Web-страниц и помещенных на них изображений, созданы, они должны быть помещены на Web-сервер. *Web-сервером* называется компьютер, на котором хранятся файлы Web-страниц, предназначенных для всеобщего доступа. Также Web-сервером называется программа, принимающая запросы от Web-обозревателей и отправляющая им нужные файлы. Поскольку эта программа выполняется на том же компьютере, на котором находятся файлы Web-страниц, эти понятия фактически равнозначны.

Две разновидности компьютерной графики

А теперь поговорим еще об одной важной вещи: о компьютерной графике и двух ее разновидностях.

Компьютерная графика может быть растровой и векторной. Каждая из этих разновидностей имеет свои достоинства и недостатки. Сейчас мы их рассмотрим.

Растровое изображение представляет собой набор точек разных цветов. Пример такого изображения показан на рис. В.2.

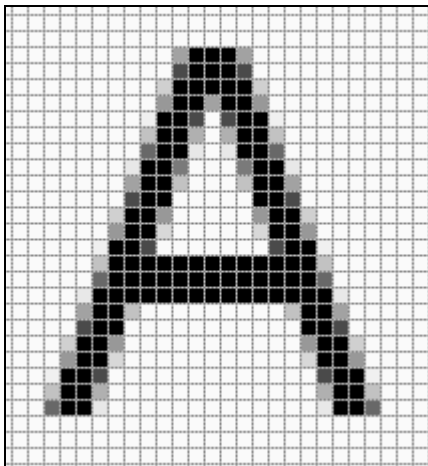


Рис. В.2. Пример растрового изображения — литера "А"

Растровыми являются подавляющее большинство графических форматов, в частности, распространенные в Интернете GIF (Graphic Interchange Format — формат обмена графикой) и JPEG (Joint Picture Encoding Group — группа кодировки неподвижных изображений). Растровыми также являются форматы BMP (BitMaP — битовая матрица), TIFF (Tag Image File Format — теговый файловый формат изображений) и PSD ("родной" формат Photoshop).

Преимуществами растровой графики являются высокая скорость вывода (в самом деле, все, что выводится на экран компьютера или принтер, — суть растровая графика) и высокая точность передачи цветов и полутонов. Недостатки: большие размеры файлов, зависимость от разрешения устройства вывода и плохая масштабируемость. Попробуйте загрузить в какую-либо программу растрового графического редактора изображение и увеличить его раза в четыре — вы увидите, как оно "распадется" на пиксели.

Векторное изображение состоит не из отдельных точек, а из набора простейших кривых — *примитивов*. Пример такого изображения показан на рис. В.3. На нем приведена та же литера "А", но уже в векторном виде.

В векторном виде сохраняются изображения форматов CDR ("родной" формат пакета CorelDRAW) и WMF (Windows MetaFile — метафайл Windows). Также векторной является графика Flash.

Преимущества векторной графики: небольшой размер результирующего файла, прекрасная масштабируемость и, как следствие, независимость от разре-

шения устройства вывода. Недостаток всего один: высокая сложность вывода, поскольку для этого нужно выполнить преобразование в растровый вид.

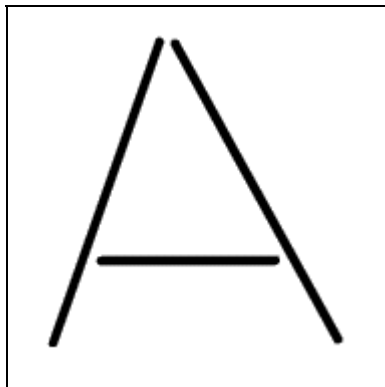


Рис. В.3. Векторное представление литеры "А"

В настоящее время растровая и векторная графика очень часто используются совместно в одном изображении. Это позволяет избежать недостатков обеих разновидностей графики, сохранив их достоинства. Такая графика называется *гибридной*. Фактически, сейчас любой векторный графический формат является гибридным, в том числе и изображения Flash.

Итак, с разновидностями компьютерной графики мы разобрались. Теперь поговорим непосредственно о графике Flash.

Как работает Flash

Графика Flash может быть сохранена в двух разных форматах. Эти форматы имеют разное назначение и называются по-разному, чтобы не было путаницы. Сейчас мы выясним, зачем вообще нужно было придумывать сразу два формата графики Flash и почему нельзя было обойтись одним.

Изображение, над которым ведется работа в среде Flash, сохраняется на диске в виде особого файла, называемого *документом Flash*. Этот файл содержит всю графику, а также анимацию и сценарии, созданные пользователем, и имеет расширение **fla**. Файл документа может быть открыт во Flash для просмотра или изменения обычным щелчком мыши в окне **Проводника**. Его также можно передать другому человеку, чтобы он просмотрел и, если нужно, изменил его.

Однако файл документа Flash не предназначен для распространения готового изображения через Интернет или каким-либо иным способом. Во-первых, этот файл содержит множество информации, нужной только для редактирования изображения, но не для его просмотра. Это значит, что кто

удобно может "влезть" внутрь сделанного вами изображения и позаимствовать оттуда все, что угодно. Вряд ли это входит в ваши планы. Во-вторых, файл документа Flash очень велик — среднее изображение может занимать сотни килобайт. Соответственно, он будет очень долго загружаться, и у желающего посмотреть ваше творение просто не хватит терпения дожидаться окончания его загрузки.

Поэтому перед распространением готового изображения его нужно сохранить в формате *Shockwave/Flash*. Файл, сохраненный в таком формате, включает только ту информацию, которая нужна для вывода изображения на экран, и имеет расширение swf. Размер этого файла очень мал — на порядок меньше, чем у исходного документа Flash. К тому же, файл Shockwave/Flash невозможно отредактировать (хотя можно использовать в другом изображении Flash).

Для просмотра изображения Flash (имеется в виду файл Shockwave/Flash) нужна особая программа, называемая *проигрывателем Flash*. Проигрыватель Flash может быть оформлен в виде обычной программы, которую можно запустить щелчком мыши и открыть в ней нужный файл Shockwave/Flash, либо в виде модуля расширения Web-обозревателя. В последнем случае изображение Flash можно просмотреть прямо в окне Web-обозревателя, загрузив его (изображение) из Интернета.

Программу проигрывателя Flash можно найти в папке <Папка, где установлена программа Flash>/Players. Соответствующий файл носит имя SAFlashPlayer.exe. А в папке <Папка, где установлена программа Flash>/Players/Release находится дистрибутив модуля расширения, носящий имя Install Flash Player 6.exe (универсальная версия) или Install Flash Player 6 AX.exe (версия для Microsoft Internet Explorer). Также модуль расширения можно установить, зайдя на сайт фирмы Macromedia (<http://www.macromedia.com>) и щелкнув гиперссылку "Download".

Надо сказать, что практически все современные Web-обозреватели уже поставляются с модулем расширения Flash. Единственное "но": этот модуль может быть (и, скорее всего, будет) устаревшим, и вам так или иначе придется установить более "свежую" его версию.

Что дальше?

Дальше мы начнем создание нашего Web-сайта. Но сначала кое о чем условимся.

Во-первых, мы используем набор из двух вертикально расположенных фреймов. В верхнем фрейме будет отображаться заголовок сайта с кнопками перехода на разные Web-страницы (*кнопками навигации*). В нижнем фрейме, занимающем большую часть окна Web-обозревателя, будет отображаться основное содержимое сайта.

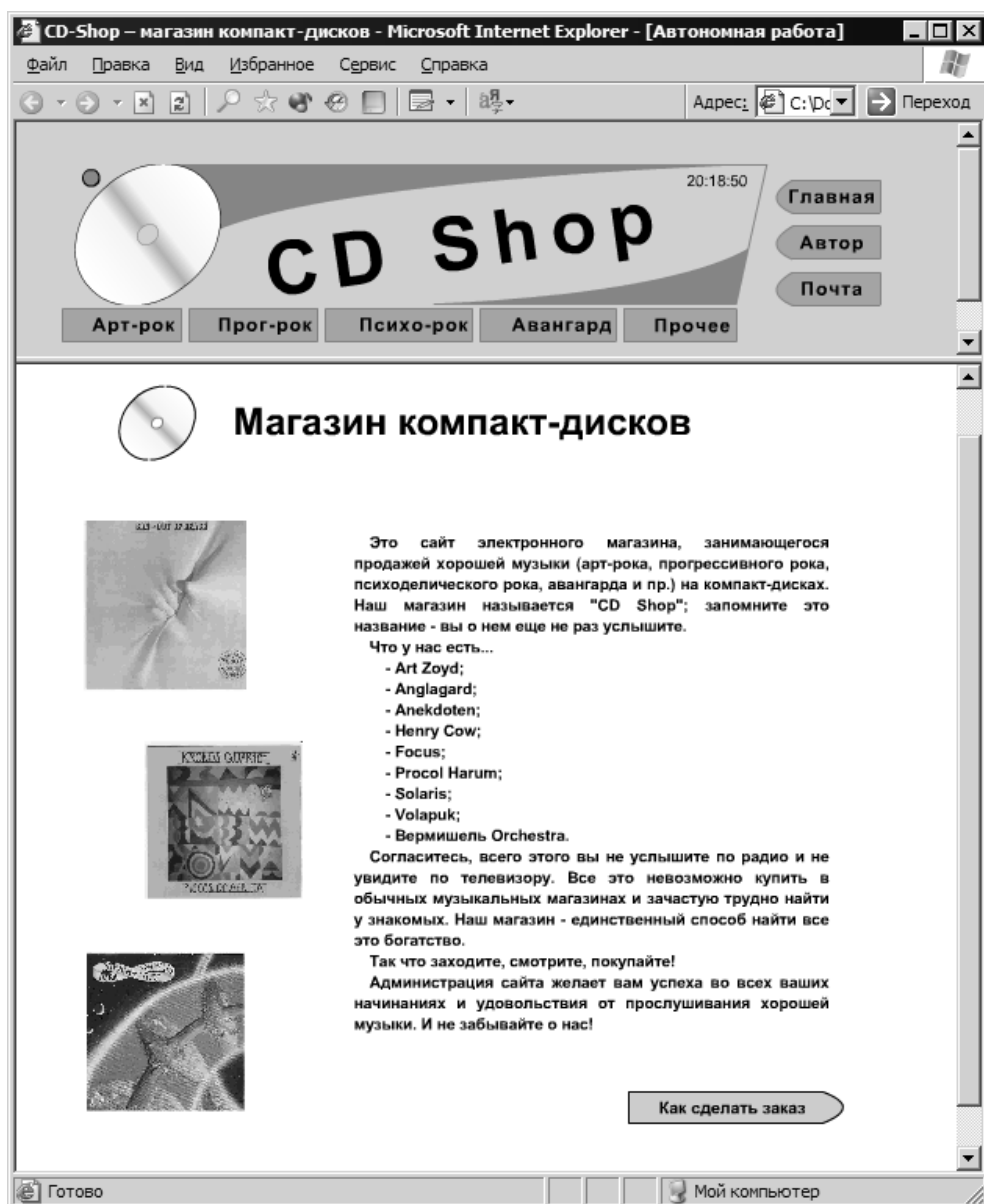


Рис. В.4. Готовый сайт интернет-магазина

Во-вторых, в формате Flash мы сделаем только две страницы сайта. А именно, заголовок и начальную страницу, содержащую сведения о сайте и описание заказа. Страницы каталога компакт-дисков мы сделаем на языке HTML. Дело в том, что обычные Web-страницы легче редактировать, а нам

это понадобится, когда мы будем добавлять в каталог новые поступления. Для редактирования Web-страницы достаточно открыть ее в Блокноте, внести нужные изменения и сохранить. А чтобы отредактировать изображение Flash, нам придется открыть исходный документ Flash, внести изменения, сохранить его, после чего создать файл Shockwave/Flash. Как видите, это займет намного больше времени.

Готовый сайт показан на рис. В.4. Обратите внимание на изображение компакт-диска в логотипе сайта — оно будет вращаться. Также мы заставим логотип плавно появляться, а кнопки навигации — "сползать" сверху. Так что преимущества Flash, о которых говорилось в *предисловии*, будут использованы полностью.

Ну все! Кажется, мы обо всем договорились. Пора за работу



ЧАСТЬ I

СТАТИЧНАЯ ГРАФИКА



Глава 1

Введение во Flash

В этой главе мы изучим основные принципы работы с пакетом Macromedia Flash MX. Мы перечислим их здесь, чтобы далее при описании возможностей или особенностей Flash не останавливаться на самых элементарных понятиях.

Итак, что же будет рассмотрено в этой главе?

Во-первых, как создается новый документ Flash MX. Ведь — согласитесь! — прежде чем начинать работу над документом, его нужно создать.

Во-вторых, пользовательский интерфейс Flash. Мы узнаем, что находится в главном окне программы, и как всем этим пользоваться. Мы научимся работать с панелями, инструментариями и собственно окном документа Flash. Конечно, мы изучим не все его возможности, а только те, что понадобятся нам в самом ближайшем времени; остальное мы рассмотрим потом.

В-третьих, как вообще выполняется работа над документами Flash. Как рисуется изображение, как оно редактируется, и какими приемами при этом вы можете воспользоваться.

В-четвертых, вспомогательные, но отнюдь не второстепенные возможности Flash по рисованию и правке графики. Ими тоже нужно уметь пользоваться.

В-пятых, что делать, если в процессе работы над изображением возникли вопросы. Любой серьезный программный продукт имеет встроенную систему интерактивной помощи, и Flash MX не стал в этом смысле исключением. Если вы будете знать, как ей воспользоваться, то сможете решить многие проблемы.

Ну и, конечно же, мы начнем создавать наш сайт электронного магазина. А именно, создадим необходимые документы Flash и зададим их параметры. Как вы помните из введения, в нашем сайте будут использованы два изображения Flash:

- ☐ заголовок сайта с логотипом магазина и кнопками навигации;
- ☐ страница сведений о самом магазине и оформлении заказа.

С заголовка-то мы и начнем.

Создание нового документа

Прежде чем начать работать с документом, этот самый документ нужно создать. Давайте выясним, как это делается.

Создание пустого документа

Новый *пустой* (не содержащий никакой графики) документ создается автоматически сразу после запуска программы Flash, так что для его создания вам не нужно выполнять никаких действий. (Точно так же ведут себя многие другие программы, работающие с документами.) Если вы хотите создать новый документ уже во время работы с программой, выберите в меню **File** (Файл) пункт **New** (Новый) или нажмите комбинацию клавиш <Ctrl>+<N>.

Каждый документ имеет определенный набор *параметров*: размер, цвет фона, частоту кадров и пр. При создании документа используются параметры, заданные по умолчанию. К сожалению, вы не имеете возможности задать параметры документа при его создании, однако можете их изменить после того, как документ будет создан.

Итак, запустите Flash, если он еще не запущен. И давайте зададим параметры заголовка нашего будущего сайта.

Внимание!

При первом запуске Flash MX выводит на экран еще одно небольшое окно. Это так называемое *окно-приглашение*, предлагающее пользователю прочитать некоторые справочные данные или запустить интерактивные презентации, объясняющие, как работать во Flash. Вы можете просмотреть их или сразу же закрыть это окно. При последующих запусках окно-приглашение больше не будет появляться.

Чтобы задать параметры документа, либо выберите пункт **Document** (Документ) в меню **Modify** (Изменить), либо выберите пункт **Document Properties** (Свойства документа) в контекстном меню листа, либо нажмите комбинацию клавиш <Ctrl>+<J>. На экране появится диалоговое окно **Document Properties** (рис. 1.1).

Поля ввода **Width** (Ширина) и **Height** (Высота), находящиеся в группе **Dimensions** (Размеры), служат для указания соответственно ширины и высоты изображения. Они задаются в текущих единицах измерения, по умолчанию это пиксели. Размеры изображения по умолчанию — 550×400, минимальный размер — 1×1, максимальный — 2880×2880 пикселей.

Ниже находится группа кнопок **Match** (Совпадение), позволяющая быстро задать размеры изображения. Кнопка **Printer** (Принтер) делает их равными текущему размеру бумаги, заданному в настройках текущего принтера. Кнопка **Contents** (Содержимое) позволит вам задать такой размер изображения, чтобы пустого места оставалось как можно меньше. Кнопка **Default**

(По умолчанию) задает размер изображения по умолчанию, т. е. 550×400 пикселей.

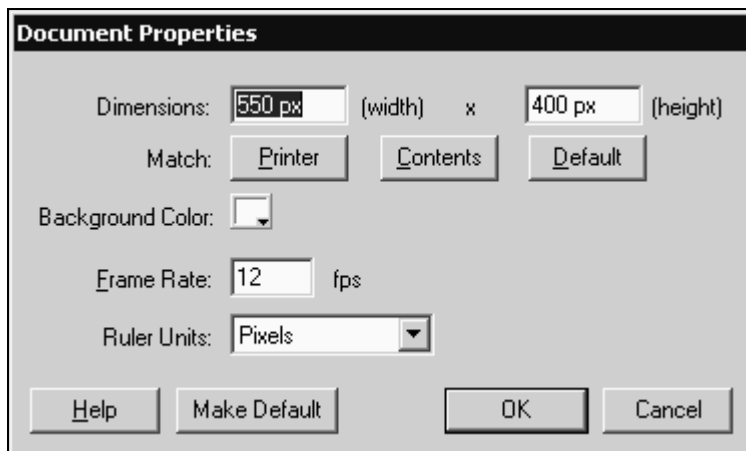


Рис. 1.1. Диалоговое окно **Document Properties**

Для заголовка сайта мы зададим размеры 550×145 пикселей. (Ширина — 550, а высота — 145 пикселей.) Не беспокойтесь, если эти размеры впоследствии окажутся слишком малы или слишком велики — вы всегда сможете их изменить. А пока что сойдут и такие.

Еще ниже кнопок **Match** находится *селектор цвета* **Background Color** (Цвет фона), имеющий вид маленького квадратика и позволяющий задать цвет фона изображения. По умолчанию этот квадратик имеет белый цвет — это значит, что для документа выбран белый фон.

Чтобы изменить цвет фона, щелкните мышью на селекторе цвета. После этого на экране появится небольшое окошко, содержащее набор цветов, доступных для выбора (рис. 1.2). Выберите мышью нужный цвет — и окошко селектора закроется. Вы также можете закрыть его, снова щелкнув цветной квадрат или нажав клавишу <Esc>.

В левом верхнем углу окна селектора цвета находится поле ввода, в котором имеется какой-то странный текст, что-то наподобие "#6699AA". Это так называемый *код цвета* — особая комбинация шестнадцатеричных цифр, уникально обозначающая тот или иной цвет. В дальнейшем для обозначения нужного цвета автор будет использовать как раз его код, так что будьте внимательны.

Когда селектор цвета открыт, курсор мыши имеет вид небольшой пипетки. Вы можете щелкнуть мышью любую точку экрана, чтобы "позаимствовать" ее цвет для работы. Как только вы закроете селектор цвета, курсор мыши примет свой обычный вид.

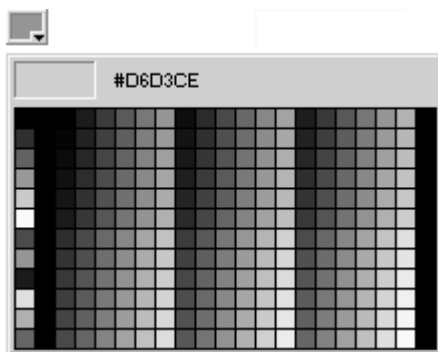


Рис. 1.2. Открытый селектор цвета

Замечание

Более подробно работа с селектором цвета будет описана в *главе 5*.

В самом деле, чтобы заголовок сайта отличался от основного содержания, нужно задать ему какой-нибудь цвет фона, отличный от белого. Щелкнем селектор цвета и выберем симпатичный и не очень яркий оранжевый цвет, имеющий код #FFCC66.

В поле ввода **Frame Rate** (Частота кадров) задается частота кадров создаваемого фильма. (Конечно, этот параметр имеет значение только в том случае, если вы создаете анимированное изображение, т. е. фильм.) Для фильмов, публикуемых в Интернете, обычно задается частота от 8 до 12 кадров в секунду. Значение по умолчанию — 12 кадров в секунду; рекомендуется его оставить, если только вы не хотите сделать свой фильм компактнее в ущерб качеству.

Раскрывающийся список **Ruler Units** (Единицы измерения) позволяет задать текущую единицу измерения, используемую, в частности, в координатных линейках. Этот список предоставляет для выбора шесть пунктов:

- ☐ **Inches** (Дюймы);
- ☐ **Inches (decimal)** (Дюймы (десятичные));
- ☐ **Points** (Пункты);
- ☐ **Centimeters** (Сантиметры);
- ☐ **Millimeters** (Миллиметры);
- ☐ **Pixels** (Пиксели) — значение по умолчанию.

Давайте оставим значение по умолчанию, т. е. пиксели (пункт **Pixels**).

Задав параметры фильма, нажмите кнопку **OK**, чтобы применить их. Если вы передумали, нажмите кнопку **Cancel** (Отмена). Если же вы хотите, чтобы заданные вами для данного документа параметры использовались по умол-