

Владимир Дронов



Macromedia
Dreamweaver
MX



Наиболее
полное
руководство

В ПОДЛИННИКЕ®

Владимир Дронов

Macromedia
Dreamweaver
MX

Санкт-Петербург

«БХВ-Петербург»

2003

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Macromedia Dreamweaver MX. — СПб.: БХВ-Петербург,
2003. — 736 с.: ил.

ISBN 5-94157-190-9

Книга посвящена описанию визуального Web-редактора Macromedia Dreamweaver MX. Изложение материала ведется от простого к сложному, начиная со способов создания примитивных Web-страничек и заканчивая основами построения сложных сайтов, включая серверное программирование и работу с базами данных. Представлены приемы, позволяющие наиболее эффективно применять для этих целей Dreamweaver MX. Приводятся примеры использования мощных средств этого Web-редактора, описаны создаваемые им теги HTML и коды JavaScript, даются рекомендации по предупреждению возможных ошибок. Рассмотрены особенности Web-дизайна, включая табличный и фреймовый его виды. Книга написана простым языком в увлекательном стиле, что позволяет даже неискушенному в Web-технологиях читателю быстро освоить новейшие возможности Dreamweaver MX.

Для Web-дизайнеров и Web-программистов

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>Анна Кузьмина</i>
Редактор	<i>Алексей Кожедуб</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульниковца</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.01.03.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 59.34.

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.

Содержание

Введение	13
ЧАСТЬ I. КАК СДЕЛАТЬ ПРОСТЕЙШИЙ WEB-САЙТ	15
Глава 1. Как создаются Web-страницы	17
Что такое Интернет.....	17
Как создаются Web-страницы.....	18
Гиперссылки	23
Клиенты и серверы Интернета.....	25
Зачем нужны HTML-редакторы	29
Что дальше?	31
Глава 2. Основные принципы работы с Dreamweaver	32
Среда Dreamweaver MX	32
Выбор рабочей среды.....	33
Главное окно программы	34
Управление окнами и панелями Dreamweaver.....	40
Работа с Web-страницами	42
Работа в окне документа.....	42
Три режима отображения Web-страницы.....	46
Работа с кодом HTML.....	48
Поиск и замена текста.....	54
Использование регулярных выражений.....	58
Просмотр Web-страницы.....	62
Вызов справки	63
Настройка Dreamweaver.....	65
Учим русский.....	66
Настраиваем скорость интернет-соединения.....	70
Добавляем программы просмотра Web-страниц.....	70
Добавляем внешний HTML-редактор.....	73
Что дальше?	74

Глава 3. Начинаем с текста	75
Создание новой Web-страницы	75
Работа с текстом	76
Ввод текста	76
Форматирование абзацев	78
Форматирование отдельных символов	83
Вставка специальных символов	92
Использование HTML-стилей	98
Работа с гиперссылками	102
Создание обычных гиперссылок	102
Создание почтовых гиперссылок	105
Другие гиперссылки	106
Создание гиперссылок на FTP-серверы	107
Новости (USENET)	107
Использование "якорей"	108
Общие свойства Web-страницы	110
Дополнительные возможности Dreamweaver	113
Использование цветовых схем	113
Вставка и чтение комментариев	115
Вставка даты	116
"Чистка" HTML-кода	118
Проверка совместимости HTML-кода	119
Что дальше?	121
Глава 4. Рисунки, звуки, фильмы	122
Работа с графическими изображениями	123
Два вида графических изображений	124
Вставка графического изображения	126
Изображения-гиперссылки	132
Активные изображения	133
Карты-изображения	135
Графика Macromedia Flash	140
Графика Macromedia Shockwave	146
Фоновые изображения	146
Мультимедиа	148
Поддержка мультимедийных данных	149
Модули расширения	151
Элементы ActiveX	155
Что дальше?	158
Глава 5. Таблицы	159
Текст фиксированного формата	160
Простые таблицы	163
Создание таблиц	163
Работа с таблицами	165
Формирование таблиц	168

Форматирование таблиц	169
Выделение элементов таблиц	169
Параметры ячейки	171
Параметры строки	173
Параметры таблицы	174
Предопределенные форматы таблиц	177
Сортировка таблицы	179
Вставка табличных данных	182
Слияние ячеек таблиц	184
Использование таблиц	188
Текст в рамке	188
Текст в графической рамке	192
Текст с отступами	195
Сложные таблицы	198
Составные изображения	204
Проблемы с таблицами и их решение	206
Общие недостатки таблиц и их преодоление	206
Проблемы с таблицами в старых версиях Navigator	207
Что дальше?	210
Глава 6. Работа с Web-сайтом	211
Начала сайтостроения	212
Планирование сайта	212
Основные этапы планирования сайта	212
Логическая структура Web-сайта	213
Физическая структура Web-сайта	216
Публикация Web-сайта	219
Управление Web-сайтом в Dreamweaver	221
Регистрация сайта в Dreamweaver	221
Настройка прокси-сервера или брандмауэра	227
Панель <i>Site</i>	229
Работа с файлами сайта	233
Взаимодействие панели <i>Site</i> и окна документа	236
Абсолютные и относительные интернет-адреса	237
Проверка правильности ссылок и HTML-кода	240
Работа с сервером	244
Публикация сайта	244
Работа с копиями сайта	249
Дополнительные возможности работы с Web-страницами и Web-сайтами	250
Примечания	250
Активы	255
Список избранных элементов	257
Библиотека	259
Дополнительные возможности поиска и замены	263
Другие возможности	263
Что дальше?	264

ЧАСТЬ II. ЗАНИМАЕМСЯ ПРОФЕССИОНАЛЬНЫМ WEB-ДИЗАЙНОМ.....	265
Глава 7. Фреймы.....	267
Введение во фреймы.....	267
Фрейм и набор фреймов.....	267
Схемы наборов фреймов.....	269
Сложные наборы фреймов.....	273
Работа с фреймами.....	274
Создание фреймов.....	275
Работа с фреймами и наборами фреймов.....	281
Свойства наборов фреймов.....	281
Свойства фреймов.....	284
Замещение и работа с ним.....	286
Создание содержимого фреймов.....	288
Заполнение фреймов.....	288
Создание остальных страниц сайта.....	291
Цель гиперссылки.....	292
Полоса навигации.....	293
Оптимизация фреймов.....	299
Уменьшение объема и сложности HTML-кода.....	299
Ускорение обработки фреймов.....	301
Решение проблем с фреймами.....	302
Недостатки фреймов и их преодоление.....	302
Проблема с фреймами в старых версиях Navigator.....	303
Что дальше?.....	305
Глава 8. Табличный дизайн.....	306
Основы табличного дизайна.....	306
Зачем нужны таблицы.....	306
Схемы табличного дизайна.....	311
Построение таблиц разметки вручную.....	314
Построение таблиц разметки автоматически.....	318
Недостатки способа создания таблиц разметки вручную.....	318
Режим разметки страницы Dreamweaver.....	319
Таблицы и ячейки разметки.....	323
Форматирование таблиц и ячеек разметки.....	329
Параметры ячеек разметки.....	329
Задание ширины ячеек.....	330
Параметры таблицы разметки.....	333
Заполнение начальной страницы.....	334
Тонкая настройка и оптимизация таблицы разметки.....	338
Недостатки табличного дизайна.....	341
Недостатки таблиц.....	341
Недостатки табличного дизайна.....	345
Что дальше?.....	346

Глава 9. Использование шаблонов	347
Введение в шаблоны Dreamweaver	348
Работа с шаблонами	349
Создание шаблона	349
Редактирование шаблона	351
Создание изменяемых областей	353
Создание Web-страниц на основе шаблонов	356
Применение шаблонов к уже созданным Web-страницам	360
Гиперссылки в шаблонах	363
Экспорт Web-страниц, основанных на шаблонах	365
Обновление страниц, созданных на основе шаблонов	366
Управление шаблонами в окне шаблонов	366
Новые возможности шаблонов	368
Изменяемые атрибуты	368
Необязательные области	370
Повторяющиеся области	373
Необязательные изменяемые области	377
Табличная повторяющаяся область	378
Вложенные шаблоны	380
Недостатки шаблонов и их преодоление	381
Что дальше?	382
Глава 10. Каскадные таблицы стилей	383
Введение в каскадные таблицы стилей	383
Зачем они нужны	384
Три способа задания стиля	390
Почему "каскадные"	392
Псевдостили гиперссылок	394
Работа с таблицами стилей в Dreamweaver	396
Создание стилей	397
Определение стиля	399
Параметры шрифта	400
Параметры фона	403
Параметры абзаца	404
Параметры размеров и размещения	407
Параметры рамки	412
Параметры маркеров списка	415
Параметры местонахождения	417
Дополнительные параметры	417
Применение стилей	420
Управление стилями	422
Управление таблицами стилей	424
Поддержка внутренних стилей	427
Таблицы стилей и шаблоны	429
Временные таблицы стилей	430
Недостатки таблиц стилей и их преодоление	431
Что дальше?	434

ЧАСТЬ III. ИСПОЛЬЗУЕМ НОВЕЙШИЕ ТЕХНОЛОГИИ.....	435
Глава 11. Свободно позиционируемые элементы	437
Введение в свободно позиционируемые элементы	438
Что такое свободно позиционируемый элемент.....	438
Как создается свободно позиционируемый элемент.....	440
Зачем нужны свободно позиционируемые элементы	445
Работа со свободно позиционируемыми элементами.....	446
Создание свободно позиционируемых элементов.....	446
Параметры свободно позиционируемых элементов.....	450
Работа с группой свободно позиционируемых элементов.....	452
Работа со свободно позиционируемыми элементами.....	453
Пример использования свободно позиционируемых элементов.....	456
Недостатки свободно позиционируемых элементов и их преодоление	462
Что дальше?	466
Глава 12. Анимация элементов Web-страниц.....	467
Введение в Web-сценарии	468
Ограничения HTML.....	468
Web-сценарии	469
События.....	471
Основные принципы анимации	472
Простейшая анимация	472
Анимация реального времени.....	474
Анимация — подход Dreamweaver.....	477
Зачем нужна анимация	479
Создание анимации в Dreamweaver	481
Создание простейшей анимации.....	481
Более сложная анимация.....	486
Управление анимациями	491
Анимация графических изображений.....	494
Недостатки анимации, основанной на Web-сценариях, и их преодоление	496
Что дальше?	498
Глава 13. Использование сценариев.....	499
Подробнее о Web-сценариях.....	500
Зачем нужны Web-сценарии	500
Язык JavaScript.....	502
Объекты	508
Объектная модель документа (DOM).....	513
Как пишутся Web-сценарии.....	516
Простейший Web-сценарий	519
Более сложный Web-сценарий.....	520
Web-сценарии — подход Dreamweaver. Поведения.....	525
Работа с поведением.....	529
Панель <i>Behaviors</i>	529
Создание поведений.....	532

Вызов JavaScript-кода (<i>Call JavaScript</i>).....	533
Изменение значения свойства (<i>Change Property</i>).....	533
Перенаправление на другую страницу в зависимости от версии Web-обозревателя (<i>Check Browser</i>).....	534
Проверка наличия модуля расширения (<i>Check Plugin</i>).....	536
Управление фильмом Shockwave или Flash (<i>Control Shockwave or Flash</i>).....	538
Перетаскивание свободно позиционируемого элемента (<i>Drag Layer</i>).....	539
Переход на заданный кадр анимации (<i>Go To Timeline Frame</i>).....	542
Переход на другую Web-страницу (<i>Go to URL</i>).....	543
Скрытие меню гиперссылок (<i>Hide Pop-Up Menu</i>).....	543
Открытие нового окна Web-обозревателя (<i>Open Browser Window</i>).....	543
Проигрывание аудиоклипа (<i>Play Sound</i>).....	545
Запуск проигрывания анимации (<i>Play Timeline</i>).....	545
Вывод предупреждения (<i>Popup Message</i>).....	545
Предварительная загрузка графических изображений (<i>Preload Images</i>).....	546
Изменение изображения-элемента полосы навигации (<i>Set Nav Bar Image</i>).....	548
Задание нового содержимого фрейма (<i>Set Text of Frame</i>).....	550
Задание нового содержимого свободно позиционируемого элемента (<i>Set Text of Layer</i>).....	551
Вывод текста в строке статуса окна Web-обозревателя (<i>Set Text of Status Bar</i>).....	552
Вывод на экран меню гиперссылок (<i>Show Pop-Up Menu</i>).....	553
Показ и скрытие свободно позиционируемых элементов (<i>Show-Hide Layers</i>).....	558
Останов проигрывания анимации (<i>Stop Timeline</i>).....	559
Изменение графического изображения (<i>Swap Image</i>).....	560
Восстановление всех изначальных изображений (<i>Swap Image Restore</i>).....	561
Написание своих Web-сценариев.....	561
Особые случаи создания поведений.....	562
Привязка поведений к тексту.....	562
Поведения и анимации.....	563
Отладка Web-сценариев.....	565
Какие ошибки бывают в программах.....	565
Встроенный отладчик Dreamweaver.....	567
Альтернативные технологии.....	571
Что дальше?.....	574
Глава 14. Метатеги и серверные директивы.....	575
Реклама в Интернете.....	575
Поисковые машины.....	576
Как работают поисковые агенты.....	579
Метатеги.....	580
Пассивная интернет-реклама.....	582
Работа с метатегами в среде Dreamweaver.....	584
Описание Web-страницы.....	585
Ключевые слова.....	586
Базовый интернет-адрес.....	587

Перезагрузка.....	589
Связи между Web-страницами.....	590
Специальные метатеги.....	593
Серверные директивы.....	595
Введение в серверные директивы.....	595
Стандартный набор серверных директив.....	596
<i>include</i>	597
<i>echo</i>	598
Как использовать серверные директивы.....	598
Поддержка Dreamweaver серверных директив.....	602
Что дальше?	604

ЧАСТЬ IV. ПИШЕМ СЕРВЕРНЫЕ ПРОГРАММЫ 605

Глава 15. Введение в серверное программирование 607

Что такое серверное программирование.....	607
Зачем нужны серверные программы.....	607
Как Web-сервер обрабатывает данные пользователя.....	608
Как Web-обозреватель отправляет введенные данные	612
Как данные передаются по Сети.....	615
Серверное программирование — подход Dreamweaver.....	617
Введение в базы данных.....	619
Что дальше?	621

Глава 16. Формы 622

Работа с формами в Dreamweaver.....	622
Создание формы.....	623
Элементы управления	624
Поле ввода	625
Кнопка	628
Флажок.....	629
Переключатели.....	630
Группа переключателей.....	631
Список	633
Поле ввода имени файла.....	635
Графическая кнопка	636
Скрытое (невидимое) поле	638
Дополнительные элементы управления	639
Метка.....	639
Группа	640
Раскрывающийся список гиперссылок.....	641
Поведения, предназначенные для работы с формами.....	644
Создание списка гиперссылок (<i>Jump Menu</i>).....	644
Создание кнопки перехода для списка гиперссылок (<i>Jump Menu Go</i>).....	644
Задание нового значения поля ввода (<i>Set Text of Text Field</i>)	645
Проверка данных, введенных в форму (<i>Validate Form</i>).....	645
Простейшая Web-форма.....	647

Использование таблиц и стилей для создания форм.....	651
Основные принципы разработки форм.....	653
Что дальше?	655
Глава 17. Простейшие серверные приложения	656
Подготовка к созданию серверных приложений.....	656
Установка соединения с базой данных.....	659
Создание источника данных ODBC.....	660
Регистрация базы данных в Dreamweaver.....	663
Создание серверных страниц в Dreamweaver.....	666
Создание простейших серверных страниц	666
Более сложные серверные страницы.....	670
Создание набора данных.....	670
Страница, отображающая данные.....	674
Создание навигатора.....	676
Создание строки статуса набора данных.....	678
Страница для одновременного просмотра нескольких записей.....	679
Привязка элементов управления к данным.....	682
Создание сложных наборов данных.....	684
Работа с динамическими атрибутами.....	686
Получение данных от другой Web-страницы.....	688
Создание фильтров.....	689
Необязательные области серверной страницы.....	691
Быстрое создание серверных страниц	693
Что дальше?	694
Глава 18. Создание интерактивных сайтов	695
Принципы создания интерактивных сайтов	696
Административные страницы сайта	697
Как администрируются Web-сайты	697
Страница списка высказываний	699
Страница добавления записи	701
Страница изменения записи	702
Средства удаления записи	706
Страница входа на сайт	708
Защита страниц от несанкционированного доступа	711
Реализация выхода с сайта	713
Страницы общего доступа.....	714
Страница списка категорий	714
Страница списка высказываний	716
Страница регистрации посетителя	717
Реализация поиска высказываний.....	719
Что дальше?	720
Заключение.....	721
Предметный указатель.....	724

Введение

Всякая книга имеет введение, где излагается, зачем же она написана и о чем рассказывает. Эта книга — не исключение. Итак...

О чем эта книга

В связи с развитием Интернета во всем мире появился спрос на профессию Web-дизайнера. *Web-дизайнер* — это человек, обладающий художественным вкусом и сведущий в интернет-технологиях, который создает Web-страницы и объединяет их в Web-сайты. Художественный вкус (а желательно и соответствующее образование) — это главное, что должен иметь Web-дизайнер (и не только Web-дизайнер, но и его коллега из области полиграфии), т. к. только знание интернет-технологий еще не служит гарантией получения качественного результата. Тем более что современные программные средства позволяют ему не вникать во всякие тонкости.

Но что делать, если вы чувствуете в себе творческий порыв, желаете самореализоваться в области Web-дизайна, но с трудом представляете, что такое HTML, CSS и JavaScript? (Конечно, знание всего этого совсем не помешает, но...) Не беда! Специально для вас разработаны программы визуальных Web-редакторов, позволяющие создавать Web-страницы и целые Web-сайты, не вникая в тонкости интернет-технологий и стандартов, так, как вы создаете документы в Microsoft Word.

Эта книга рассказывает о Macromedia Dreamweaver MX — последней версии популярного Web-редактора компании Macromedia. Это исключительно мощная программа, поддерживающая все современные стандарты Интернета и невероятно облегчающая выполнение даже самых сложных задач. Кроме того, она содержит в своем составе развитую систему подсказки и интерактивных уроков, позволяющих начинающему пользователю быстро приступить к работе. По иронии судьбы, Dreamweaver "задвинут" в тень своего

более "раскрученного" собрата Microsoft FrontPage. Но — будьте уверены! — он не заслужил такой участи.

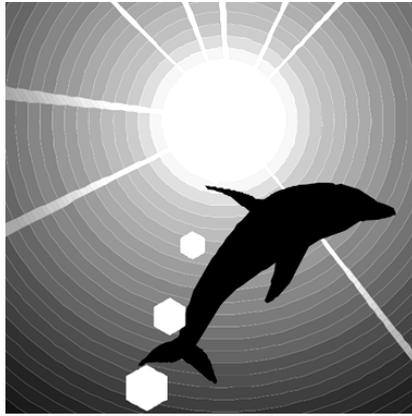
Может быть совсем и по-другому. Вы — опытный Web-дизайнер, предпочитающий все делать "врукопашную", но желающий немного автоматизировать свою работу. Dreamweaver поможет вам и в этом случае. Вы сможете быстро сделать "набросок" своей Web-страницы, а потом довести код "до ума" вручную. Поверьте, так работать гораздо удобнее — и вы быстро привыкнете.

Macromedia Dreamweaver — достаточно "старый" пакет. Первая его версия была разработана еще в 1998 году и получила популярность в России благодаря своей "благосклонности" к русскому языку и снисходительному отношению к множеству русскоязычных кодировок. Я пользуюсь Dreamweaver, начиная с версии 2.0. В прошлом году я написал книгу о предыдущей версии Dreamweaver — 4.0. Теперь же, с выходом Dreamweaver MX, я снова взялся за перо, так сказать.

Изучать Dreamweaver мы будем на конкретном примере. Мы будем делать личный Web-сайт гипотетического Web-дизайнера Ивана Ивановича Иванова. Сначала этот сайт будет совсем простеньким; на его основе мы изучим базовые инструменты программы и основные принципы Web-дизайна. Впоследствии сайт станет сложнее и красивее; мы будем использовать фреймы и таблицы, чтобы придать ему профессиональный вид. Далее мы изучим таблицы стилей, встроенные средства для анимации элементов страницы и сценарии, позволяющие "оживить" страницы нашего сайта. И на самой последней ступени мы изучим серверное программирование и сделаем для нашего сайта гостевую книгу. Ну и, конечно же, мы узнаем, как собрать разрозненные страницы в сайт и опубликовать его в Сети.

Но прежде чем начать разговор о Macromedia Dreamweaver MX, дадим несколько основных понятий и ответим на несколько вопросов, которые могут возникнуть у неподготовленного читателя. Конечно, если вы на короткой ноге с WWW, можете пропустить первую главу и начать читать сразу со второй. Но автор обязан позаботиться обо всех, кто будет читать эту книгу.

Итак, первый вопрос... Вернее, ответ. Вы найдете его в первой главе настоящей книги.



ЧАСТЬ I

Как сделать простейший Web-сайт

- Глава 1. Как создаются Web-страницы
- Глава 2. Основные принципы работы с Dreamweaver
- Глава 3. Начинаем с текста
- Глава 4. Рисунки, звуки, фильмы
- Глава 5. Таблицы
- Глава 6. Работа с Web-сайтом

ГЛАВА 1



Как создаются Web-страницы

Так как же делаются те красивые Web-странички, которые выводит нам Web-обозреватель? И откуда они берутся? А вот я слышал в автобусе слово "WWW" и не знаю, что это такое... И вообще, что такое Интернет и почему мы его не видим, как, например, монитор или сообщения об ошибках Windows?

Все-все! Сейчас попытаемся ответить на эти вопросы. И начнем с самого последнего.

Что такое Интернет

В самом деле, что такое Интернет? Электронный океан, таинственная стихия, заключенная в кремниевых кристаллах и медных проводах современных компьютеров. Несуществующая вселенная, иной раз кажущаяся более реальной, чем наш материальный, "настоящий" мир. То, что вторгается в каждый дом, опутывает всю планету и сознание всех людей тугой медно-кремниевой паутиной; нечто запредельное, непостижимое людскому разуму, никому не видимое, но всеми ощущаемое...

Но довольно! Вы слишком много читаете фантастики (или газет типа "Церковного вестника"). На самом деле, все намного проще. Возможно, вы даже немного разочаруетесь, узнав, что такое Интернет. Но правда должна быть сказана, чего бы это ни стоило. Ради этого и написана данная книга.

Итак, что такое Интернет и как он работает?

Вот объяснение для зануд и педантов. *Интернет* — это совокупность компьютерных сетей, связанных друг с другом и работающих по единым стандартам. Говоря простыми словами, это множество локальных сетей, плюс пользователи, подключающиеся к ним по модемам, плюс соединяющие эти сети высокоскоростные каналы связи, плюс еще пара миллионов тонн вся-

ких железяк, пара триллионов строк программного кода и пара-тройка тысяч стандартов, призванных навести в этом барахле подобие порядка.¹

Собственно, грубое описание Интернета дано. Конечно, на самом деле все неизмеримо сложнее, но для нас сейчас достаточно этого краткого описания. Если же вы хотите знать больше, читайте специальные книги. А пока что заканчиваем этот раздел, самый краткий в настоящей книге, и переходим к следующему...

Как создаются Web-страницы

А теперь поговорим о том, как создаются Web-страницы.

Для этого используется особый язык *HTML* (HyperText Markup Language — язык гипертекстовой разметки). Этот язык определяет набор специальных команд, называемых *тегами* и используемых для задания форматирования или назначения тех или иных *элементов* Web-страницы. Особые теги используются для размещения на Web-страницах графических изображений, аудио- и видеоклипов и прочих так называемых *внедренных объектов*.

Кажется, я вас напугал. Но, несмотря на кажущуюся сложность, Web-страницы не представляют собой ничего сложного. Это обычные текстовые файлы, созданные в стандартном Блокноте или аналогичном простейшем текстовом редакторе. И содержат они текст, тот самый текст, который вы хотите поместить на страницы, только размеченный особым образом...

Да что тут говорить! Лучше все показать на примере. Взгляните на рис. 1.1. Это простейшая Web-страничка, сделанная для примера в Блокноте.

Чтобы увидеть эту страничку воочию, откройте Блокнот, наберите приведенный ниже код, проверьте его на ошибки и сохраните в файле с именем 1.1.htm. Только когда будете вводить имя файла в стандартном окне сохранения, заключите его в кавычки, иначе Блокнот добавит расширение txt, и ваш файл получит имя 1.1.htm.txt. После этого откройте полученный файл в Web-обозревателе, для чего достаточно дважды щелкнуть по нему мышью.

Теперь давайте рассмотрим сам HTML-код нашей первой странички.

```
<HTML>
<HEAD>
  <TITLE>Web-страница</TITLE>
</HEAD>
<BODY>
```

¹ Говорят, в первой польской энциклопедии, изданной, кажется, в XVII столетии, термин "лошадь" описывался так: "что такое лошадь, знают все". Точно то же самое можно сейчас сказать об Интернете. (Вот только можно ли сейчас сказать то же самое о лошади?..)

```
<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 виден в окне Web-обозревателя на рис. 1.1. 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>
```

Вы видите какие-то слова, заключенные в угловые скобки < и >. Это и есть теги HTML. Они задают форматирование текста. Скажем, строка <i>Блокноте</i> будет выведена курсивом, т. к. теги <i> и </i> задают курсивное начертание текста. Причем, тег <i> помечает начало курсивного фрагмента (*открывающий* тег), а тег </i> — конец (*закрывающий* тег). А собственно фрагмент, заключенный между открывающим и закрывающим тегами, называется *содержимым* тега.

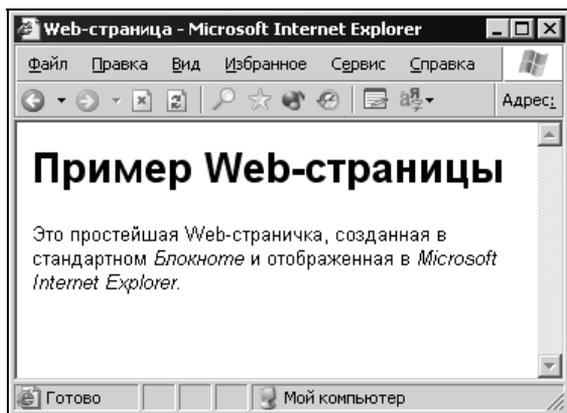


Рис. 1.1. Простейшая Web-страничка

Это были теги *физического форматирования* текста, т. е. просто задающие, как текст должен выглядеть в окне Web-обозревателя. Теперь рассмотрим теги *логического форматирования*, которые позволяют разбить текст на отдельные логические блоки. В нашем небольшом фрагменте это теги <p> и <h1> (и соответствующие им закрывающие теги </p> и </h1>). Они задают соответственно обычный текстовый параграф и заголовок первого уровня;

при этом Web-обозреватель будет знать, что `<n1>Пример Web-страницы</n1>` — это заголовок, и отобразит его соответствующим образом. Как и в предыдущем случае, открывающий тег помечает начало логического блока текста, а закрывающий — конец.

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

```

```

Здесь мы видим, что само графическое изображение хранится в отдельном файле `picture.gif`. То есть, изображение — это один из внедренных элементов Web-страницы. Тег `` содержит в себе параметр `src`, задающий имя файла изображения. (Такие параметры называются *атрибутами* тега.) Web-обозреватель, встретив в HTML-коде страницы тег ``, загружает файл, чье имя задано атрибутом `src`, и отображает его.

Осталось рассмотреть совсем немного тегов, используемых для служебных целей и не отображаемых Web-обозревателем. Они так и называются — *невидимые* теги.

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>` *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 `<n1>`!) Среди этой служебной информации может быть *название* страницы, показываемое в заголовке окна Web-обозревателя; оно задается парным тегом `<title>...</title>`.

Давайте рассмотрим заголовок нашей страницы.

```
<head>
  <title>Web-страница</title>
</head>
```

Собственно, заголовок содержит только название нашей страницы, помеченное тегом `<title>...</title>`. Но в подавляющем большинстве случаев заголовок содержит множество других данных, необходимых как Web-обозревателю, так и другим программам, обеспечивающим функционирование Интернета и WWW. Мы пока не будем их рассматривать.

Взгляните еще раз на HTML-код нашей страницы. Заметьте, что одни теги вложены в другие. Так, тег `<i>` вложен в тег `<p>`, тег `<p>` — в тег `<body>`, а тег `<body>` — в тег `<html>`. Такая *вложенность* или *иерархия* тегов в HTML встречается на каждом шагу. Давайте представим ее в виде схемы.

```
<HTML>
  <HEAD>
    <TITLE>
  <BODY>
    <H1>
    <P>
      <I>
```

Здесь мы убрали мешающие нам закрывающие теги и их содержимое. Величина отступа показывает *уровень вложенности* того или иного тега. Так, тег `<BODY>` имеет первый уровень вложенности, а тег `<H1>` — второй. Теги предыдущих уровней вложенности называются *родительскими тегами* или *родителями*, а теги последующих уровней — *дочерними тегами* или *потомками*. Например, для тега `<HEAD>` родительским тегом будет `<HTML>`, а дочерним — `<TITLE>`. Для тега `<BODY>` родителем будет тег `<HTML>`, а потомками — теги `<H1>`, `<P>` и `<I>`.

Весь HTML-код Web-страницы вложен внутрь парного тега `<HTML>...</HTML>`. Этот тег находится на самом высшем, нулевом, уровне вложенности и не имеет родителя.

Я здорово перегрузил вас информацией, и вам может показаться, что HTML — это что-то запредельно сложное. На самом деле, все это очень просто, и вы можете легко создавать простейшие Web-странички вручную. И в подтверждение этого давайте немного изменим код нашей страницы. Например, выделим название фирмы Microsoft жирным шрифтом. Тег, задающий "жирность", — `...`.

Далее приведем измененный фрагмент HTML-кода.

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

Заметьте, как мы вложили один тег в другой (эти теги выделены). Теперь слово "Microsoft" будет набрано и жирным шрифтом, и курсивно. Только обратите внимание на *порядок вложенности* тегов и не нарушайте его. Если же вы случайно допустите ошибку и поместите теги так:

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

то Web-обозреватель может и не отобразить ваше творение (хотя Internet Explorer славится своим умением исправлять мелкие ошибки Web-дизайнера). Запомните простое правило: закрывающие теги должны повторяться в порядке, обратном порядку соответствующих им открывающих тегов.

Сохраните полученный файл под именем 1.2.htm и откройте его в Web-обозревателе. То, что вы увидите, показано на рис. 1.2.

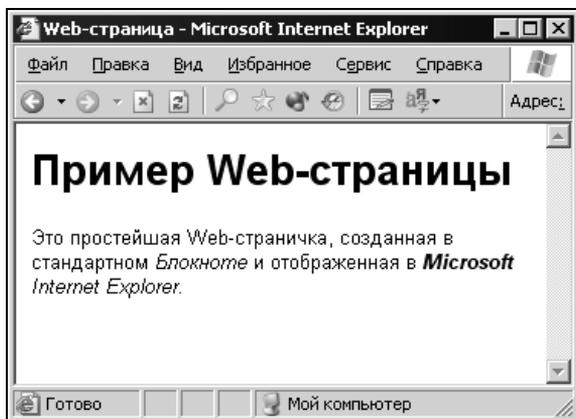


Рис. 1.2. Измененная Web-страничка

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*. Это название можно перевести как "Комитет Всемирной паутины"... нет, лучше не переводить!... ужасное название...

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 версии Web-обозревателей поддерживают эту версию HTML.

Наряду с тегами HTML, стандартизированными WWWC (*стандартные* теги), Web-обозреватели поддерживают множество *нестандартных* тегов. Эти теги были введены разработчиками той или иной программы Web-обозревателя, чтобы получить преимущество перед конкурентами. Они надеялись, что эти фирменные расширения впоследствии войдут в стандарт HTML, но этим надеждам так и не суждено было сбыться. Однако теги остались, и они все так же поддерживаются. Мы опишем эти теги наряду со стандартными, но предупреждаем вас, что они поддерживаются не всеми программами.

Версия 4.01 станет последней версией языка HTML. В дальнейшем он будет постепенно заменен своим потомком — языком *XHTML* (eXtensible HyperText Markup Language — расширяемый язык гипертекстовой разметки). Этот язык основан на популярном языке описания данных *XML* (eXtensible Markup Language — расширяемый язык разметки), он позволяет расширять

доступный набор тегов и считается более строгим и более подходящим для машинной обработки. К тому же, из XHTML выброшено множество устаревших тегов, все еще поддерживаемых HTML в целях совместимости.

Скоро ли наступит эра XHTML? Вряд ли. Старый добрый HTML поцарствует на земле еще не один год. Поэтому не стоит торопиться заучивать новые теги и переписывать уже написанные Web-страницы. Займитесь более полезными делами. Например, наведите порядок в ящиках своего стола или нанесите визиты своим одноклассникам.

Гиперссылки

Не только и не столько Web-страницы прославили Интернет. Свою роль внесло еще одно замечательное изобретение, буквально связавшее разрозненные документы в настоящую паутину. Она так и называется — Всемирная паутина, а по-английски — *World Wide Web* или WWW. (Вы уже встречали это название в наименовании World Wide Web Consortium — организации, ответственной за стандартизацию Интернета.) Благодаря этому изобретению вы можете с легкостью перемещаться по страницам, ведь любой уголок Всемирной сети находится от вас на расстоянии щелчка мыши.

Это *гиперссылки* — особые связи, ведущие от одной Web-страницы к другой. Именно по ним вы щелкаете мышью, если хотите 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>
```

Как видите, тег `<A>` содержит атрибут `HREF`, который задает *интернет-адрес* страницы, на которую будет выполнен переход при щелчке по гиперссылке. Этот атрибут обязательно должен присутствовать в любом теге `<A>`, задающем гиперссылку — это *обязательный* атрибут. (Как выяснится далее, тег `<A>` может задавать не только гиперссылки.)

Интернет-адрес нужной Web-страницы вам нужно выяснять самим. Сделать это не так уж и сложно. Обычно он состоит из интернет-адреса Web-сайта, на котором находится нужная страница, и полного пути доступа к файлу этой страницы. В нашем случае, `http://www.somesite.com` — это адрес сайта, а `/pages/page125.htm` — путь доступа к файлу страницы.

Если нужно получить доступ к странице, находящейся на том же сайте, что и текущая, вы можете опустить адрес сайта. В этом случае код гиперссылки будет таким:

```
<A HREF="/pages/page125.html">Страница N125</A>
```

Если же нужная страница находится еще и в той же папке, что текущая, можно вообще оставить только имя файла:

```
<A HREF="page125.html">Страница N125</A>
```

А теперь давайте создадим две простейшие Web-странички и свяжем их гиперссылкой. В качестве первой страницы мы возьмем уже написанную нами ранее и сохраненную в файле 1.2.htm (см. рис. 1.2). Только добавим внизу следующий код:

```
<P><A HREF="1.4.htm">Сведения об авторе</A></P>
```

Сохраните новую страницу в файле 1.3.htm.

Код второй страницы приведен ниже. Я уверен, что вы в нем разберетесь.

```
<HTML>
  <HEAD>
    <TITLE>Сведения об авторе</TITLE>
  </HEAD>
  <BODY>
    <P>Эту страничку написал я. И я очень горд этим!</P>
  </BODY>
</HTML>
```

Сохраните ее в файле 1.4.htm.

Теперь откройте в Web-обозревателе файл 1.3.htm и щелкните по гиперссылке "Сведения об авторе". В окне Web-обозревателя появится страница со сведениями об авторе, сохраненная в файле 1.4.htm.

А теперь сделаем небольшой фокус. Измените код гиперссылки, помещенный в HTML-код первой страницы, таким образом (изменения выделены полужирным шрифтом):

```
<P><A HREF="1.4.htm" TARGET="_blank">Сведения об авторе</A></P>
```

Мы поместили в тег <A> атрибут TARGET, задающий цель гиперссылки. Цель гиперссылки задает, куда будет выведена Web-страница, на которую она указывает. 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), присвойте атрибуту TARGET значение `_self` или вообще уберите его из кода гиперссылки.

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

Сохраните измененную Web-страницу и повторно откройте ее в Web-обозревателе, после чего щелкните по гиперссылке. На экране появится но-

вое окно Web-обозревателя, в котором вы увидите страницу со сведениями об авторе. Вы можете увидеть оба этих окна на рис. 1.3.

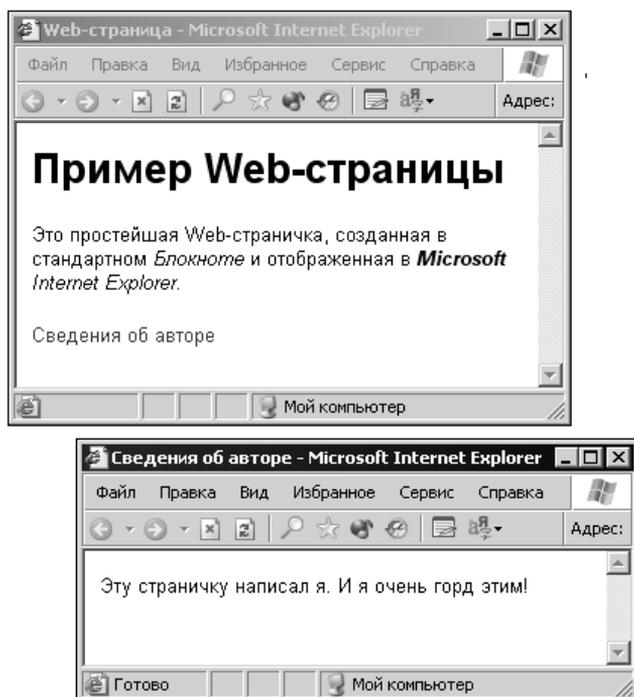


Рис. 1.3. Два окна Web-обозревателя, в которых загружены разные Web-страницы

Чтобы лучше узнать человека, нужно познакомиться с ним поближе. Так и с гиперссылками. В дальнейшем мы поговорим о них подробнее. А пока — закончим!

Клиенты и серверы Интернета

Итак, продолжим изучение основ Интернета и WWW.

Вы уже знаете, что Web-страницы сохраняются в текстовых файлах с расширением htm или html. Графические изображения, аудио- и видеоролики и прочие внедренные нетекстовые элементы сохраняются в виде отдельных файлов; их расширение зависит от формата файла. Web-обозреватель, загрузив страницу, содержащую внедренные элементы, загружает также нужные графические, звуковые и видеофайлы и формирует окончательный вид страницы.

Но как же Web-страницы, созданные кем-то, доставляются на ваш рабочий стол? По Интернету, скажете вы. И будете правы. Но как именно?

Сейчас мы это опишем.

Начнем с того, что все программы для Интернета делятся на две большие группы: клиенты и серверы. Программы-*серверы* предоставляют те или иные ресурсы для доступа программам-*клиентам*. Клиенты, если им понадобится какой-то файл или вообще какие-либо данные от сервера, формируют специальный *клиентский запрос* и посылают его на сервер. Сервер обрабатывает этот запрос и высылает *серверный ответ*, содержащий запрашиваемые данные или сообщение об ошибке, если нужные данные почему-то не доступны.

Такая компьютерная *архитектура*, иначе говоря, принцип построения вычислительной системы или сети, называется архитектурой "*клиент-сервер*" или *двухзвенной*. Этим она отличается от *однозвенной* или *одноранговой* архитектуры, когда все объединенные в сеть компьютеры равны между собой и могут как разделять, так и использовать сетевые ресурсы. Именно на основе двухзвенной архитектуры функционируют почти все сервисы Интернета. В том числе, и WWW.

В случае WWW клиентами выступают хорошо вам знакомые программы Web-обозревателей (или аналогичные программы, обрабатывающие Web-страницы, например, программы закладки Web-сайтов). Серверами же являются так называемые *Web-серверы*, обрабатывающие запросы Web-обозревателей и высылающие им нужные файлы.

С термином "Web-сервер" (как и термином "сервер") существует определенная путаница. Во-первых, так называют компьютер, подключенный к Сети по быстрому каналу и хранящий на своих жестких дисках файлы Web-страниц. Во-вторых, так называется программа, работающая на этом самом компьютере, принимающая от Web-обозревателей запросы и выдающая им соответствующие файлы. Но чаще и компьютер, и программу объединяют в единое целое и называют одним словом — Web-сервер. Так поступим и мы. (Хотя правильно называть Web-сервером именно программу.)

Когда вы набираете в поле адреса Web-обозревателя какой-либо интернет-адрес, обозреватель обращается к соответствующему Web-серверу. Сервер же извлекает со своих дисков нужные файлы (сама Web-страница, изображения, внедренные объекты, архивы, исполняемые файлы) и отправляет их Web-обозревателю. А уж он-то знает, как с ними поступить.

Люди, для того чтобы понимать друг друга, должны разговаривать на одном языке. Это же относится и к компьютерам. В их случае в качестве такого "языка общения" выступает *протокол* — набор правил обмена данными. Web-сервер и Web-обозреватель используют для обмена данными протокол *HTTP* (Hypertext Transfer Protocol — протокол обмена гипертекстом). Этот *высокоуровневый* протокол работает "поверх" обычного *низкоуровневого* протокола *TCP/IP* (Transfer Control Protocol/Internet Protocol — протокол управления обменом/протокол Интернета).

Примечание

Функции Web-сервера может выполнять даже файловая система вашего компьютера. Учитывая эту возможность, мы сохранили нашу первую страничку на жестком диске, а потом щелкнули по нему мышью. Файловая система в ответ на запрос пользовательской оболочки загрузила этот файл и отобразила его в связанной с ним программе — Web-обозревателе.

Web-серверы устанавливаются на больших компьютерах, принадлежащих различным организациям. Также в Интернете доступно множество бесплатных Web-серверов, предоставляющих место для сайтов всем желающим. (Конечно, бесплатные серверы имеют много ограничений, но для большинства небольших некоммерческих сайтов это идеальный способ выйти в Сеть. Как говорится, дешево и сердито.) И наконец, существуют коммерческие организации, предоставляющие место на дисках Web-сервера всем желающим, но за деньги. Такие коммерческие серверы не имеют многих ограничений бесплатных серверов и поэтому пользуются успехом среди коммерческих организаций.

Итак, автор, чтобы сделать свои Web-страницы доступными для широкой публики, должен поместить их на жесткие диски компьютера, на котором работает программа Web-сервера. Этот процесс называется *публикацией* страниц (опытные Web-дизайнеры также часто говорят "выложить" вместо "опубликовать"). Процесс публикации страниц будет описан далее, в *главе 6*.

Установкой и настройкой Web-сервера, а также устранением проблем и наблюдением за его работой занимается особый человек — *администратор* Web-сервера. От действий (или бездействия) администратора зависит, какие возможности поддерживает Web-сервер и насколько надежно он работает. Как правило, вы не можете повлиять на действия администратора (тем более, если он администрирует бесплатный общедоступный сервер), если, конечно, сами не являетесь администратором своего собственного Web-сервера.

Одна из Web-страниц, хранящихся на диске серверного компьютера, при настройке Web-сервера задается в качестве *страницы по умолчанию* или *главной страницы*. Она будет загружена, если пользователь наберет в строке адреса Web-обозревателя только интернет-адрес сайта (например, "http://www.site.ru"), без указания имени файла страницы. Как правило, такая страница имеет имя default или index.htm и расширение htm или html, хотя, опять же, все это в руках администратора.

После получения файлов Web-страницы Web-обозреватель сохраняет их на жестком диске клиентского компьютера в особой папке, называемой *кэшем*. Зачем это нужно? Да хотя бы затем, чтобы вы смогли впоследствии просмотреть эту страницу, не подключаясь к Интернету. Все современные Web-обозреватели поддерживают так называемый *автономный режим* (по-английски — offline mode), когда они отображают только те страницы, что

находятся в кэше. Если же вы попытаетесь просмотреть страницу, которой нет в кэше, Web-обозреватель предложит вам подключиться к Интернету и загрузить ее.

Теперь познакомимся с популярнейшими в настоящее время Web-обозревателями.

Самым популярным среди них, настоящим королем виртуальных просторов, является Microsoft Internet Explorer. Он установлен на любом компьютере, работающем под управлением Windows, что, как говорят злые языки, и обусловило его популярность. Однако это очень мощная, быстрая, весьма нетребовательная к ресурсам и исключительно удобная программа, что бы там ни говорили его критики. Автор этой книги для просмотра Web-страниц пользуется именно Internet Explorer. В настоящее время доступна версия 6.0 и, по всей вероятности, разрабатывается новая версия, хотя фирма Microsoft, по своему обыкновению, хранит по этому поводу молчание.

Второе место по популярности занимает норвежская разработка Opera, выпускаемая одноименной фирмой. Эта достаточно мощная и очень быстрая программа, тем не менее, весьма охоча до системных ресурсов, особенно при отображении сложных Web-страниц. Кроме того, она является платной; в бесплатной версии она показывает рекламу. Последняя имеющаяся у автора версия носит номер 6.05 и, скорее всего, после выхода книги она устареет, т. к. новые версии Opera появляются очень часто.

Некогда властелин WWW Netscape Navigator сейчас в глубоком загоне — в настоящее время им пользуются от силы 2% интернетчиков. Хотя последняя версия Navigator — 7 — выглядит весьма неплохо, поддерживает все стандарты WWWC, корректно отображает большинство Web-страниц и не очень требовательна к системным ресурсам. Но все равно Navigator по многим параметрам проигрывает и Internet Explorer, и Opera.

Не очень давно был наконец-то выпущен в свет новый Web-обозреватель — Mozilla. Эта программа распространяется бесплатно, более того, ее исходные тексты открыты для изучения и модификации. Она построена на том же программном ядре, что и Navigator 7, точнее, наоборот, Navigator 7 построен на основе Mozilla. (Собственно, Mozilla и создавался для обкатки нового программного ядра Navigator, но в дальнейшем вырос в самостоятельный продукт.) Этот новичок весьма неплох, поддерживает все Web-стандарты, нетребователен к системным ресурсам, довольно быстр и имеет множество интересных и весьма полезных возможностей, которыми пока не может похвастаться ни один из его конкурентов. Пока что он не очень популярен, но в дальнейшем, возможно, еще себя покажет. Последняя версия, доступная в момент написания книги, — 1.1.

Существует еще некоторое количество программ Web-обозревателей, но они малопопулярны. Так что в настоящее время WWW принадлежит четверке перечисленных нами программ. Есть также некоторое количество Web-

обозревателей, построенных на основе Internet Explorer и расширяющих его возможности. Но поскольку это не самостоятельные продукты, они рассматриваться не будут.

А сколько в мире существует популярных Web-серверов? О-о-о, вероятно, не меньше, если не больше, чем популярных Web-обозревателей. Самыми популярными сейчас являются распространяемый с открытыми исходниками Apache и разработанный фирмой Microsoft Internet Information Server. Также неплох используемый автором для своего сайта Web-сервер Sambar.

Кстати, насчет Web-серверов. Если вы работаете в среде Microsoft Windows 98/ME или Windows 2000/XP, то можете установить Web-сервер, поставляемый с этой системой, и попробовать себя в качестве администратора. Это Personal Web Server (поставляется с Windows 98/ME) или Internet Information Server (Windows 2000/XP). Он нетребователен к ресурсам системы, довольно быстр в работе и удобен в настройке. Документация к нему поставляется также в составе операционной системы, к тому же, существует довольно много книг, посвященных этому Web-серверу. Советую вам поработать с ним, т. к. в дальнейшем он нам понадобится.

Вот и все о клиентах и серверах Интернета.

Зачем нужны HTML-редакторы

На этом завершим краткое введение в интернет-технологии. Вы узнали, что создавать Web-страницы очень просто, и для этого достаточно простейшего текстового редактора. Также вы познакомились с программами Web-серверов и узнали, как работает клиент-серверная архитектура. В связи с этим возникает вопрос: если язык HTML так прост, а в состав Windows входит Блокнот, прекрасно с ним справляющийся, то зачем нужны еще и Web-редакторы вроде Macromedia Dreamweaver?

А вот зачем...

"Уберите от меня подальше этот проклятый HTML!!! — кричит один читатель. — Я ничего в нем не смыслю! Он слишком сложен для меня, а у меня нет времени ему учиться. Дайте мне нормальный текстовый редактор, тот же Microsoft Word, и я буду работать в нем." И он будет прав.

HTML, конечно, прост. Но для кого-то он может оказаться невероятно сложным, ведь люди все разные. Кроме того, сложные Web-страницы писать "врукопашную" крайне неудобно. Уж поверьте! Поэтому вам понадобятся программы, автоматизирующие ваш труд — эти самые *Web-редакторы*, к славной плеяде которых относится и Macromedia Dreamweaver.

"Но мне нравится HTML! — возразит другой читатель, собаку съевший в Web-дизайне. — Мне проще вводить вручную HTML-теги, чем елозить по тексту мышкой и тыкать кнопки." И он тоже будет прав.

Ведь права старая поговорка "Пуля — дура, а штык — молодец". Опытный мастер вручную или с помощью простейших инструментов может сделать такое, что не под силу механизированным и автоматизированным заводам. Не зря же все шедевры — в любой области искусства — делаются, как правило, вручную. А механизмам оставляют разве только изготовление заготовок...

Выходит, правы и первый, и второй читатели. Так как же помирить два враждующих лагеря: HTML-поклонников и HTML-ненавистников? Похоже, что никак...

Вот поэтому на свете и существуют две разновидности Web-редакторов: визуальные и невизуальные. Или, как еще говорят, WYSIWYG- и не-WYSIWYG-редакторы. (WYSIWYG или What You See Is What You Get — "что ты видишь, то ты и получишь".)

Невизуальные редакторы работают с "чистым" (или "сырым") HTML-кодом. Они предоставляют возможность быстрого ввода тегов, синтаксического подсвечивания, проверки правильности получившегося кода и ссылок. К ним относится, в частности, известная Arachnophilia.

Примечание

Невизуальные Web-редакторы часто называют HTML-редакторами.

Визуальные редакторы позволяют работать с самой Web-страницей "как она есть". Пользователь редактирует и форматирует текст, вставляет рисунки, таблицы, как в обычном текстовом редакторе, а уж сама программа формирует соответствующий HTML-код. К такого рода редакторам относится популярный Microsoft FrontPage и целый ряд программ попроще.

Как вы уже поняли, невизуальные редакторы прекрасно подходят для опытных Web-дизайнеров, съевших собаку на HTML и прекрасно представляющих, как должна выглядеть Web-страница в результате тех или иных изменений в ее коде. Визуальные редакторы — отличная стартовая площадка для начинающих Web-дизайнеров, плохо знакомых с HTML. Но и опытные дизайнеры часто пользуются визуальными редакторами, чтобы быстро "набросать" Web-страничку, а потом довести ее до ума "врукопашную".

Специально для таких пользователей, совмещающих прекрасное знание HTML с пристрастием к визуальным средствам редактирования, существуют так называемые *гибридные редакторы*. Они имеют мощные средства визуального редактирования, одновременно предоставляя доступ к получившемуся HTML-коду. Именно к таким редакторам и относится Macromedia Dreamweaver.

Примечание

Вообще-то сейчас, судя по всему, уже нет чисто визуальных редакторов — все они являются гибридными. Просто у одних доступ к HTML-коду осуществляется

проще (Dreamweaver), а у других — чуть сложнее (FrontPage). Поэтому практически всегда, когда говорят "визуальный Web-редактор", подразумевают как раз гибридные программы.

Вдобавок, как правило, все более-менее мощные редакторы Web-страниц — и визуальные, и невизуальные — имеют в своем составе развитые средства управления сайтом. Они знают, из каких страниц состоит ваш сайт, помогут опубликовать его на Web-сервере, предупредят о гиперссылках, указывающих "в никуда", и даже позволят вам составить список всех действий, которые вы хотите проделать с сайтом. Конечно же, Dreamweaver это тоже может.

Что дальше?

Вот и закончился наш краткий курс интернет-технологий. Конечно, многое здесь не описано. Все, что вам будет необходимо для создания и публикации Web-страниц, будет рассмотрено по ходу дела. А сейчас давайте, не углубляясь в дебри Интернета, перейдем к нашей главной цели — к работе с замечательным пакетом Web-редактора Macromedia Dreamweaver MX.