Самоучитель

Вадим Дунаев

Основы Web-дизайна

2-е издание

HTML 4/5 и XHTML

CSS 2/3

JavaScript

Практические примеры и советы

BHV

УДК 681.3.06
ББК 32.973.26-018.2
Д83

Дунаев В. В.

Д83 Основы Web-дизайна. Самоучитель. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 480 с.: ил.

ISBN 978-5-9775-0110-1

Рассмотрены основные сведения по разработке Web-приложений, необходимые начинающему Web-разработчику. Приведены общие понятия, а также вопросы, связанные с разработкой, публикацией и дальнейшим сопровождением сайта в целом. Подробно рассмотрены разметка гипертекстовых документов (XHTML, HTML 4 и HTML 5) и применение каскадных таблиц стилей (CSS 2/3). Описаны основные средства разметки страниц сайта и форматирования текстов. Показано, как помещать в HTML-документ содержимое из внешних источников (графические изображения, Flash-ролики, звук, видео и т. п.), а также как создавать специальные визуальные эффекты. Уделено большое внимание элементам пользовательского интерфейса, с помощью которых обеспечивается интерактивность Web-страниц. Рассказано о решении различных задач с помощью сценариев, написанных на языке JavaScript. Второе издание переработано и дополнено с учетом современных Web-технологий.

Для начинающих Web-дизайнеров

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Игорь Цырульников*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Подписано в печать 30.09.11.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 38,7.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0110-1

Оглавление

ГЛАВА 1

Что такое Web-дизайн

Суть дизайна — формообразование. Форма — внешняя оболочка, представление и способ существования содержания любой вещи или явления, в том числе и сайтов, созданием которых занимаются Web-дизайнеры.

Вы можете заняться разработкой сайта для себя или стороннего заказчика. В любом случае работа состоит из нескольких этапов. Некоторые из них особенно важны, если заказчик и подрядчик (исполнитель) — разные лица. С заказчиком бывает трудно договориться, по крайней мере на первых этапах, не столько из-за его привередливости или скупости, сколько из-за того, что зачастую он сам не вполне осознает, что конкретно ему нужно. Идеальная ситуация — заказчик выдает вам техническое задание, достаточно ясное и подробное, чтобы немедленно приступить к проектированию. В реальности же техническое задание на проект чаще всего вырабатывается совместными усилиями исполнителя и заказчика итеративно.

В данной главе мы рассмотрим в общих чертах основные этапы создания Web-сайта. Попробуйте представить себя как в роли исполнителя, так и в роли заказчика сайта.

1.1. С чего начать?

Как известно, хорошее начало — более половины всего дела. А начать следует с выяснения цели разработки сайта, т. е. того, для чего он нужен. Например, сайт может предназначаться для рекламы и/или продажи товаров и услуг (интернет-магазин), представления сведений о фирме или персоне (сайт-визитка), размещения информационных статей или медиаресурсов и т. д. Для информационных сайтов также важна тема: культура, наука, искусство, образование, технология, развлечения и т. д. При этом важно знать, какого типа информацию (тексты, графика, видео, аудио) и какие сервисы должен предлагать сайт. Желательно хотя бы приблизительно оценить объем публикуемой информации и частоту ее обновления. Сайты бывают статическими, содержимое которых обновляется время от времени, и динамическими, информация в которых может изменяться чуть ли не ежеминутно (блоги, форумы, гостевые книги). Неплохо также иметь предположения о потенциальной аудитории сайта (социальная и профессиональная группы, образование, возраст). Далее заказчик обычно высказывает свои предположения и пожелания о том, как должна выглядеть главная (домашняя, начальная) страница сайта. Здесь речь может зайти о стиле, цветовой гамме, композиции и т. п. Постарайтесь выяснить, каким должен быть дизайн с эмоциональной точки зрения за-

казчика: спокойным или вызывающим, строгим или легкомысленным, жизнерадостным или мрачным, пестрым или аскетичным, насыщенным деталями или лаконичным. Ваша задача — выяснить вкусы заказчика. Разумеется, они могут не совпадать с вашими. Но о вкусах, как говорят, не спорят. В данной книге я всячески пытался избежать обсуждения тем, связанных с вкусовщиной. Не критикуйте вкус заказчика, но учитывайте его, предлагая свое решение.

Заказчик может показать вам несколько уже существующих сайтов, которые ему нравятся, но при этом пожелать, чтобы его будущий собственный сайт обладал явно выраженной индивидуальностью. Казалось бы, здесь возникает противоречие: с одной стороны, требуется выдержать некий стиль, схожесть с уже существующими дизайнами, а с другой — обеспечить заметное отличие от последних. Это противоречие и должен разрешить дизайнер. В архитектуре, например, оно ведь разрешается. Так, среди сохранившихся соборов готического стиля много очень похожих, как близнецы, друг на друга, но есть и неповторимые, например Нотр-Дам в Реймсе, Нотр-Дам в Париже и собор Святого Павла в Страсбурге.

Не следует забывать, что при всех изысках внешнего вида сайта его потенциальный посетитель не должен испытывать ни малейших затруднений относительно элементов управления (пользовательского интерфейса). Ссылки, кнопки, поля ввода данных и т. п. должны легко узнаваться, не требуя от посетителя экспериментальных исследований с помощью мыши, чтобы выяснить, с чем именно он имеет дело. Главная цель дизайна — сделать сайт как можно более удобным и облегчить доступ к находящейся на нем информации. Удивить и заворожить посетителя — дело третьестепенное, если не десятое.

Хорошо, если заказчик ограничивается лишь общими замечаниями относительно облика главной страницы. Тем не менее часто бывает, что он пытается излишне детализировать задание на дизайн. Например, заказчик может явно указать, что заголовки должны мигать, там-то следует разместить Flash-ролики, а фон страницы должен иметь вид кирпичной стены. Казалось бы, это должно упростить задачу дизайнера. Но, как правило, подобного рода указания существенно ограничивают необходимые для творчества степени свободы дизайнера и, наоборот, усложняют его задачу. Может статься, понравившийся заказчику дизайн вызовет отвращение у самого дизайнера. Тогда, разумеется, можно утешиться тем, что заказчик всегда прав или же просто отказаться от заказа. Однако на практике дизайн, непосредственно воплощающий прямые и слишком детализированные указания заказчика, часто потом не нравится и ему самому. В лучшем случае заказчик получит урок, что "пироги должен печь пирожник, а сапоги тачать — сапожник", а в худшем — он обвинит вас в непрофессионализме. И тогда дизайнер приступает к разработке макета или эскиза сайта сообразно с собственными знаниями, опытом, навыками и вкусом, но, конечно, учитывая общие пожелания и пристрастия заказчика. Впрочем, опытный дизайнер должен прийти к этому, не тратя излишне свои силы и время на бесплатные уроки. Он просто заранее имеет достаточно весомые аргументы, чтобы убедить заказчика довериться профессионалу. Другое дело, если заказчиком и исполнителем является одно и то же лицо, причем новичок. В этом случае редко когда удается сразу, без переделок, создать хороший проект, потому что обычно "первый блин — комом".

Если вы создаете сайт для себя лично и в первый раз в жизни, то при выборе дизайнерского решения критически посмотрите на чужие разработки и мысленно представьте

себе, что вы делаете не первый и не последний сайт в своей жизни. Даже если в дальнейшем вы не будете этим заниматься, данный психологический прием убережет вас от неуместного загромождения своих страниц всякими экзотическими штучками. Как говорил Л. Толстой, "главное свойство во всяком искусстве — чувство меры". А меру мудрецы считали синонимом мудрости. Дело в том, что новички в своем первом проекте стараются продемонстрировать, что они очень много знают и умеют. Аналогично, многие заказчики сайтов для своих фирм нередко хотят, чтобы на их сайте присутствовало все самое модное и "передовое", чуть ли не все такое, что они уже где-то видели, и даже более того. Так что, неопытные дизайнер и заказчик могут совместными усилиями породить чудовище не только по внешнему виду, но и в структурно-функциональном смысле, отталкивающее посетителей и внешним видом, и плохой функциональностью. На мой взгляд, красивое и гармоничное не может быть чрезмерно сложным. По крайней мере, сложность создаваемого объекта не должна быть видна.

1.2. Графический дизайн

Рассмотренный в предыдущем разделе этап Web-дизайна считается предварительным, поскольку он обычно не завершается заключением письменного договора или хотя бы устного соглашения между заказчиком и исполнителем. Перед тем как заключить такое соглашение заказчик, как правило, хочет получить от исполнителя эскиз (макет) главной и, возможно, еще нескольких страниц будущего сайта, поскольку "лучше один раз увидеть, чем сто раз услышать". Так возникает задача графического дизайна сайта.

Заказчик может потребовать графический дизайн не только для того, чтобы после возможной его коррекции принять наконец-то решение о дальнейшей разработке сайта. Графический дизайн может представлять собой и самостоятельный объект договора, т. е. рассматриваться как законченный продукт. Его разработка может быть поручена одному исполнителю, а все остальное (воплощение в кодах, заполнение содержимым и т. д.) — другому. Так что, приступая к графическому дизайну, вы должны отдавать себе отчет о статусе этой разработки: является ли она только промежуточным или же окончательным этапом вашей работы. Нередко разработку и оплату графического дизайна оформляют специальным соглашением.

Если графический дизайн рассматривается как промежуточный этап разработки сайта в целом, то вам следует подготовить несколько (два-три) его вариантов. В результате обсуждения этих вариантов дизайна с заказчиком в идеале должен быть выбран один из них. Однако бывает и так, что заказчик желает сделать некий гибрид из представленных кандидатов. В этом случае возникает опасность разрушения концептуального и художественного единства вашего замысла проекта, о которой уже говорилось в предыдущем разделе. Тем не менее вы должны быть готовы к итеративному процессу рассмотрения вариантов графического дизайна. Окончательный выбор все равно следует пока рассматривать лишь в качестве рабочего, поскольку дальнейшая разработка сайта может потребовать внесения коррективов и в его графическое оформление. Однако если графический дизайн хорош не только с эстетической, но и с технологической (структурно-функциональной) точки зрения, то необходимая впоследствии его коррекция будет и минимальна, и относительно безболезненна. Иначе говоря, создавая эскизы страниц сайта, нужно все время помнить о технологических аспектах.

Теперь вкратце рассмотрим, как делают графический дизайн страниц сайта. Напомню, что заказчику обычно требуется дизайн одной-двух страниц — главной, которая загру-

жается в браузер первой, и той, которая отображается при переходах по ссылкам. Различные страницы сайта могут иметь разную компоновку и средства навигации, но должны быть выполнены в едином стиле. Это хорошо не только в эстетическом смысле, но и с точки зрения поддержки ориентации потенциального посетителя в информационном пространстве. Единство стиля различных страниц создает ненавязчивое ощущение, что мы находимся в пределах одного и того же сайта. Самый простой способ обеспечения единства стиля различных страниц — применять единые цветовые решения, параметры шрифтов и средств навигации.

Конечный результат графического дизайна можно представить в виде:

- графических изображений страниц в целом;
- шаблонов страниц, т. е. набора файлов с HTML- и CSS-кодами и графическими изображениями.

Графическое изображение страницы сайта можно нарисовать на бумаге с различной степенью подробности и тщательности отделки. Этот способ многие используют при создании личного сайта. Однако стороннему заказчику нужна не бумага, а файл графического формата, и тогда дизайн выполняют с помощью графических редакторов, таких как коммерческие Adobe Photoshop, CorelDRAW и свободно распространяемые GIMP, Inkscape.

В графическом редакторе создают изображение страницы целиком, включая элементы пользовательского интерфейса (гиперссылки, кнопки, переключатели, поля ввода данных и т. п.). Если графический дизайн не является предметом отдельного заказа, а служит лишь в качестве эскиза или прототипа страницы для согласования с заказчиком всего проекта создания сайта, то его сохраняют в файле растрового формата, например, PNG или JPEG. В противном случае (специальный заказ) графический дизайн страницы выполняют как многослойное изображение, в котором каждый элемент страницы рисуется в отдельном слое, наложенном на предыдущие. Это делается для того, чтобы элементы графического дизайна можно было в дальнейшем скорректировать или заменить другими, а также использовать при реализации страниц сайта. Элементы графического дизайна имеют самостоятельную ценность, и желательно иметь возможность извлечения их из первоначальной целостной конструкции. В данном случае изображение страницы сохраняют либо в растровом формате с поддержкой слоев (например, PSD, TIFF), либо в векторном формате (например, EPS). Файлы таких форматов можно открыть и изменить во многих графических редакторах, например Adobe Photoshop, Adobe Illustrator, CorelDRAW, GIMP и др.

Графический дизайн в виде картинки в дальнейшем используется как ориентир для разработки структуры и функциональности страницы. Разработчик, взглянув на такое графическое изображение и сопровождающее его словесное описание, должен придумать, какими программными средствами воплотить замысел художника. Например, он решает, сделать ли название сайта как графическое изображение, или вставить его как соответствующим образом отформатированный текст, т. е. применяя HTML и CSS; создать ли обычные гиперссылки или же графическую карту ссылок.

Вместо того чтобы рисовать картинку страницы, можно сразу создать ее шаблон, используя язык разметки HTML, каскадные таблицы стилей (CSS) и, если потребуется, графические изображения. Шаблон — это прототип страницы без информационного

содержимого, с ограниченной функциональностью или вообще без таковой (например, не работают ссылки, меню, формы и т. п.). Преимущество данного способа заключается в том, что не требуется в совершенстве владеть графическим редактором и уметь создавать с его помощью оригинальные изображения, навыки рисования в данном случае не являются необходимыми. Важно также, что шаблон легко модифицировать практически на любом этапе разработки сайта. Кроме того, его в дальнейшем можно использовать (немного скорректировав) и для других сайтов. Иначе говоря, использование шаблона позволяет путем относительно небольших модификаций его параметров получить существенные изменения внешнего вида Web-страниц. Если ваша задача состоит в разработке только лишь графического дизайна, то конечный результат вашей работы можно получить просто как скриншот (снимок экрана) загруженного в браузер шаблона.

Даже если графический дизайн выполнен как картинка, последующая разработка сайта все равно предполагает создание прототипов (шаблонов) его страниц, которые можно загрузить и просмотреть в браузере. Так что создание шаблона — не просто один из методов графического дизайна, но одновременно и этап разработки дизайна сайта вообще.

1.3. Шаблоны страниц

Готовые шаблоны нетрудно найти в Интернете, а можно сделать их и самому. Корректировать имеющиеся и создавать новые шаблоны, наполнять их содержимым (т. е. делать собственно Web-страницы) можно с помощью специальных программ визуальной разработки, таких как Adobe (Macromedia) Dreamweaver, Microsoft FrontPage или Microsoft SharePoint Designer. Кстати, Microsoft SharePoint Designer 2007 предоставляется бесплатно.

Системы визуальной разработки популярны в среде новичков, не желающих возиться с HTML- и CSS-кодами, не говоря уж о скриптах. Тем не менее их используют и профессионалы, особенно в условиях разработки одновременно нескольких сайтов, когда необходимо существенно повысить свою производительность. Однако следует иметь в виду, что системы визуальной разработки автоматически создают соответствующие программные коды, причем далеко не оптимальные ни по объему, ни по структуре, ни по выбору привлеченных HTML-элементов. В автоматически созданных кодах нелегко разобраться, возможно придется потратить на это неделю. А ведь сайт, скорее всего, придется сопровождать (дополнять, модифицировать) после сдачи заказчику. В данной книге мы будем рассматривать вопросы дизайна в плане создания шаблонов вручную, т. е. методом прямого программирования. Такой подход, разумеется, не исключает практическое применение систем визуальной разработки, а наоборот, позволяет увеличить эффективность их использования. Действительно, зная HTML и CSS, вы можете вносить коррективы непосредственно в автоматически сгенерированный код и в то же время пользоваться удобными инструментальными средствами визуального проектирования.

Суть создания шаблона заключается в том, что:

- разрабатывается компоновочная схема или, другими словами, структурная разметка страницы сайта. Такая схема может быть представлена в виде таблицы, ячейки которой определяют места размещения оформительской графики (логотип, элементы

украшения), средств пользовательского интерфейса (меню, кнопки, ссылки и другие элементы навигации и взаимодействия с пользователем) и собственно содержимого (контента, как еще говорят) страницы. Для сайта может потребоваться более одной компоновочной схемы: одна для главной страницы, а другие — для страниц перехода по ссылкам с главной. Очевидно, тривиальная схема представляется одноячеечной таблицей. Простая схема содержит две ячейки. Наиболее часто встречаются схемы из трех-пяти ячеек. Примеры типичных компоновочных схем показаны на рис. 1.1. Выбор конкретной схемы зависит от многих факторов, основные из которых — характер размещаемых в ее ячейках элементов (графические изображения, средства навигации, тексты и т. д.), степень важности, а также тематической и структурной разнородности информации;

- для ячеек компоновочной схемы определяются цвет фона и/или фоновые графические изображения, а также параметры границ;
- для текстовой информации (прежде всего, для шрифтов) определяются стилевые параметры. Причем такие параметры могут задаваться и для всей страницы, и отдельно для каждой ее ячейки с текстом.

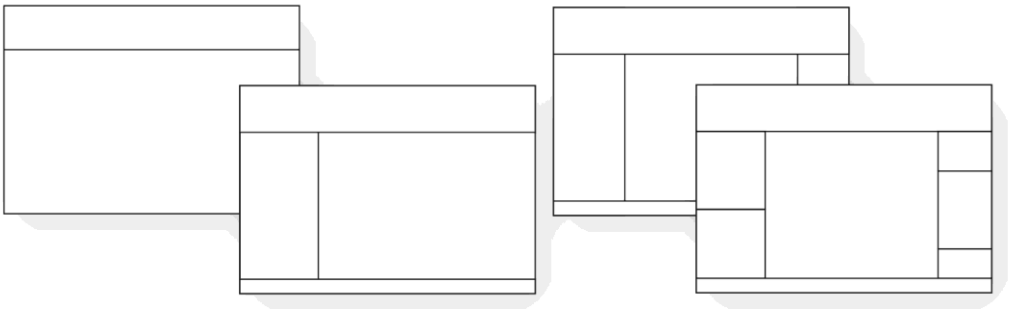

Рис. 1.1. Типовые варианты компоновочных схем страниц сайта

Наконец, определяется, каким образом компоновочная схема будет позиционироваться в окне браузера: должны ли изменяться ее размеры в зависимости от размеров клиентской области браузера или нет, должна ли она центрироваться относительно краев этой области.

1.4. Реализация проекта

После выбора компоновочной схемы страницы сайта приступают к реализации проекта, т. е. программированию с использованием HTML, CSS, а если понадобится, то и языков сценариев, таких как JavaScript, PHP и др. На данном этапе разработки сайта следует учесть несколько важных моментов:

- не исключено, что на какой-то стадии реализации придется вернуться к пересмотру компоновочной схемы, а может быть, скорректировать и графический дизайн. Вероятность такого развития событий тем меньше, чем лучше выполнена работа на предыдущих этапах. Последнее означает, что графический дизайн и структура страницы выполнены с учетом технологических особенностей предстоящей реализации проекта в целом;

- разработанный сайт после публикации в Интернете, скорее всего, придется сопровождать: изменять частично или полностью информационное содержимое страниц, добавлять новые страницы и т. п. Сложные технические устройства, например, должны обладать целым рядом эксплуатационных свойств, в числе которых — ремонтопригодность. Аналогично, хорошо разработанный сайт также должен обладать хорошими эксплуатационными качествами. Нередко заказчик особо требует, чтобы разработчик сайта обеспечил ему простоту и удобство обновления информационного содержимого без помощи разработчика и даже без обращения к сведущему в Web-дизайне специалисту. Поэтому уже на этапе разработки следует предусмотреть такую возможность;
- современный сайт должен быть инвариантен (одинаково выглядеть и функционировать) относительно всех современных ведущих браузеров, таких как Microsoft Internet Explorer, Mozilla Firefox, Opera, Google Chrome, Apple Safari. При этом инвариантность должна достигаться не только для самых свежих, но и для хотя бы самых распространенных старых версий. В настоящее время, пожалуй, только Internet Explorer широко представлен в сети сразу несколькими своими версиями (6—9). Дело в том, что многие используют тот браузер (а это, как правило, интегрированный в Windows Internet Explorer), который они получили с приобретением компьютера. Поэтому следует проверять разрабатываемый сайт по крайней мере для версии 8 — последней для все еще широко распространенной Windows XP и последующих версий браузера, работающих под Windows 7 и более поздними выпусками данной операционной системы. Пользователи других браузеров в подавляющем числе случаев имеют самые последние их версии. Так что при разработке сайта следует обзавестись несколькими браузерами, хотя бы четырьмя: Internet Explorer, Firefox, Opera и Chrome. Ко времени выхода в свет данной книги доля пользователей указанных браузеров составляла более 97%. В настоящее время браузеры распространяются через Интернет свободно.

Теперь рассмотрим, каким минимальным оснащением должен обладать разработчик сайта. Прежде всего, нужен обычный текстовый редактор вроде блокнота Windows. Лучше использовать более специализированный редактор, обеспечивающий подсветку синтаксиса программных кодов на различных языках, например, свободно распространяемый Notepad++ (**http://notepad-plus-plus.org**). В текстовом редакторе создаются коды HTML и CSS, а также сценарии на скриптовых языках, таких как JavaScript и PHP. Само собой разумеется, нужен браузер, чтобы просматривать результаты своей работы. Точнее, нужны несколько ведущих браузеров для проверки, что создаваемый сайт достаточно инвариантен относительно них. Для подготовки графических элементов потребуется графический редактор. Сейчас для этой цели широко применяется коммерческий Adobe Photoshop, но вполне можно использовать достаточно мощный, удобный и бесплатный GIMP (**http://www.gimp.org**). Современные ведущие браузеры оснащены средствами разработчика, которые являются хорошим подспорьем при создании и отладке страниц сайта. Так, вы можете просмотреть HTML-код, параметры CSS, скрипты, сообщения об ошибках, относящиеся к загруженной в браузер странице. Наконец, необходим FTP-клиент — программное обеспечение закачки файлов на сервер по протоколу FTP (File Transfer Protocol). В настоящее время существует множество бесплатных и хороших FTP-клиентов. Кроме того, на сайтах компаний, предоставляющих хостинг, обычно есть и сервис для FTP-обмена файлами.

Если вы создаете сайт без серверной части (т. е. без серверных сценариев), то минимальный набор перечисленных средств достаточен. Чтобы протестировать страницу сайта, вы можете на локальном компьютере открыть в браузере соответствующий HTML-файл и посмотреть, работает ли страница согласно вашим требованиям и ожиданиям и как она выглядит.

Серверная часть сайта не является необходимой, если вам не нужно принимать, обрабатывать и сохранять данные, введенные посетителем, а также вести базы данных. В противном случае, например, при создании блога, форума, гостевой книги, интернет-магазина и т. п. без серверной части не обойтись. Тогда для тестирования и отладки сайта потребуется Web-сервер.

Где взять Web-сервер? Во-первых, вы можете воспользоваться Web-сервером хоста в Интернете, на котором в дальнейшем собираетесь опубликовать свой сайт или же на другом, который доступен вам и подходит для отладочных целей. В данном случае необходимо перенести уже разработанные файлы на удаленный сервер и далее производить отладку в режиме онлайн, заменяя при необходимости устаревшие версии файлов новыми. В настоящее время имеется довольно много как бесплатных, так и платных хостингов. Однако при выборе хостинга следует учитывать состав предоставляемых возможностей. Например, популярный бесплатный хостинг Яндекс.Народ (**http://narod.yandex.ru**) не дает возможности использовать серверные сценарии и ограничивает объем файлов. Хостинг "Веб Сервис" (**http://www.webservis.ru**) предоставляет возможность применять серверные сценарии на PHP бесплатно, но не позволяет публиковать мультимедийные файлы некоторых форматов. Хостинг uCoz (**http://www.ucoz.ru**) предоставляет удобный сервис конструктора в режиме онлайн, обеспечивающий создание динамических сайтов (блогов, форумов, гостевых книг). Однако статические сайты, состоящие только из HTML-документов и не применяющие средства взаимодействия с посетителями, не приветствуются данным хостингом.

Общее ограничение, накладываемое всеми бесплатными хостингами:

- ❒ публикуемый сайт не должен иметь коммерческой направленности (так что интернет-магазины на бесплатных хостингах устраивать не стоит);
- ❒ на страницах вашего сайта хостинг разместит свой баннер. Он может появляться где попало и портить внешний вид страниц, даже если его можно закрыть щелчком мыши. Такова плата за бесплатный хостинг;
- ❒ объем отдельных файлов и общий объем дискового пространства для сайта в целом ограничиваются. Например, под сайт может быть отпущено несколько сот мегабайтов, но отдельный файл не должен превышать 2 Мбайт.

Разумеется, платный хостинг предлагает более широкий набор услуг при меньших ограничениях. Выбираемый для отладки сайта хостинг должен иметь параметры, максимально близкие к параметрам того хостинга, на котором вы собираетесь опубликовать окончательную версию сайта. В идеале это должен быть один и тот же хостинг. Отладка сайта в режиме онлайн хороша тем, что вам не нужно самому устанавливать и, в большинстве случаев, настраивать Web-сервер, а также модули, выполняющие серверные сценарии, системы управления базами данных и т. п. Все это сделает администратор платного хостинга. Если же хостинг бесплатный, то вам нужно лишь адаптироваться к предлагаемому набору уже настроенных средств, получив необходимую информацию на сайте хостинговой компании либо у ее администратора. Однако вам

придется заплатить за трафик. Тем не менее, данный режим широко применяется при создании простых и средних по сложности сайтов.

Во-вторых, вы можете отлаживать разрабатываемый сайт в режиме оффлайн, установив на локальном компьютере Web-сервер самостоятельно, например, Apache или Internet Information Services (IIS). Кроссплатформенный Apache распространяется свободно (**http://www.apache.org**), а IIS входит в дистрибутивы Windows XP Professional, Windows 7 Professional/Enterprise/Ultimate, но не устанавливается как служба автоматически. При необходимости IIS следует найти среди компонентов Windows и установить самостоятельно.

Серверные сценарии пишут на различных языках. Одним из наиболее популярных является PHP, и его поддерживает большинство Web-серверов в Интернете. Если вы установили на локальном компьютере Web-сервер, то вам придется также установить и модуль PHP, распространяемый свободно (**http://www.php.net** или **http://ru.php.net**).

Если ваш сайт должен взаимодействовать посредством серверных сценариев с базами данных, то потребуется еще установить и систему управления базами данных (СУБД), причем такую, которая поддерживается Web-сервером того хостинга, на котором вы собираетесь опубликовать свой сайт. Это может быть, например, Microsoft SQL Server (для платформы Windows) или кроссплатформенная СУБД MySQL, распространяемая бесплатно (**http://www.mysql.ru**). Практически все серверы, поддерживающие PHP, поддерживают также и MySQL.

Самостоятельная установка на локальном компьютере Web-сервера, модулей PHP и СУБД — несложная задача, и начинающему Web-дизайнеру было бы неплохо попытаться ее решить хотя бы в ознакомительных целях, даже если принято решение работать в режиме онлайн. Однако она может быть существенно упрощена, если воспользоваться какой-нибудь системой автоматизации разработки сайтов, например бесплатной системой WebMatrix от Microsoft (**http://www.microsoft.com/web/webmatrix**). WebMatrix — среда разработки на платформе Windows с хорошим пользовательским интерфейсом, содержащая все необходимое для создания сайтов: Web-сервер IIS Express, СУБД SQL Server Compact, среды программирования PHP и ASP.NET, удобные редакторы кодов, средства публикации сайта на хостинге по протоколам FTP, FTPS и WebDeploy, функцию анализа сайта с целью выработки рекомендаций по его оптимизации для поисковых систем (т. е. функцию SEO — Search Engine Optimization) и др.

В данной книге мы не будем рассматривать, как пользоваться системами автоматизации разработки и сопровождения сайтов, а остановимся на выполнении необходимых операций вручную. Это позволит лучше понять, как устроен сайт и как его модифицировать.

При любом варианте (онлайн или оффлайн) отладки разрабатываемого сайта вам следует продумать структуру папок, в которых будут храниться файлы. Имена папок и файлов должны быть в латинском нижнем регистре. Создайте на локальном компьютере корневую папку вашего сайта, например mysite. HTML-файл главной (домашней, начальной) страницы должен иметь имя index.html или index.htm и находиться в корневой папке сайта. Все остальные файлы сайта могут располагаться в той же папке, но это неудобно, особенно если их несколько десятков или сотен. Так, желательно хранить файлы каскадных таблиц стилей, графические изображения, скрипты, документы и т. п.

в отдельных папках. Пример файловой структуры сайта показан на рис. 1.2. Разумеется, иерархия и имена папок могут быть иными. Так, файлы дополнительных страниц сайта могут располагаться в отдельных папках, например page1, page 2 и т. д. Главное — чтобы структура папок была удобна вам как разработчику, а также тем, кто будет в дальнейшем сопровождать сайт. Следует иметь в виду, что имя папки для серверных сценариев нередко предопределяется хостингом. Например, это может быть папка cgi-bin.

Рис. 1.2. Пример файловой структуры сайта

При публикации сайта в сети или при использовании Web-сервера на локальном компьютере *содержимое* корневой папки сайта (а не сама корневая папка) должно быть перенесено в корневую папку (домашний каталог) Web-сервера. Например, если ваш сайт на локальном компьютере имеет файловую структуру как на рис. 1.2, то в корневой каталог Web-сервера следует перенести не папку mysite, а вложенные в нее другие папки и файл index.html. Иногда хостингом предопределяется подпапка корневой папки Web-сервера, в которой вы должны разместить содержимое локальной корневой папки сайта. Замечу, что при работе с локальным Web-сервером можно указать, какую папку следует считать домашним каталогом последнего, чтобы не переносить файлы в каталог, принятый по умолчанию.

Чтобы открыть HTML-файл в браузере при работе с локальным Web-сервером, следует в адресной строке указать URL-адрес, а не просто путь к файлу на диске: **http://localhost/*имя_файла*** или **http://*имя_сервера*/*имя_файла***. Здесь ***имя_сервера*** — сетевое имя или цифровой IP-адрес компьютера, на котором установлен Web-сервер; ***имя_файла*** — имя требуемого файла, содержащее при необходимости путь к нему относительно домашнего каталога Web-сервера. Если сервер установлен на вашем локальном компьютере, то вместо конкретного имени сервера можно указать **localhost**, например, **http://localhost/docs/page1.html**. Очевидно, **http://localhost** — URL-адрес главной страницы сайта, т. е. файла index.html или index.htm. Подробно он записывается либо как **http://localhost/index.html**, либо как **http://localhost/index.htm** в зависимости от того, какое именно расширение имеет файл главной страницы.

1.5. Наполнение информационным содержимым

В идеале, наполнение страниц сайта информационным содержимым (контентом) — отдельная задача, а на практике она оказывается тесно связанной с дизайном, в том числе и графическим. Какая-то часть контента, полученная от заказчика, анализируется и форматируется разработчиком в соответствии с общей концепцией дизайна сайта. Например, оформление названия сайта, заголовков рубрик, гиперссылок, данных о кон-

тактах и т. п. выполняется дизайнером с самого начала так, как это нужно для публикации в Интернете. Однако обычно сайт справедливо рассматривается заказчиком как система представления информации, первоначально созданной и оформленной посредством других весьма разнообразных средств, таких как текстовый редактор (например, Microsoft Office Word), фото и видеокамера, графический редактор (например, CorelDRAW) и т. д. В этом случае требуются мероприятия по преобразованию исходного материала к надлежащему виду.

Вот типичная ситуация. Заказчик сайта имеет файл, созданный в Word и содержащий иллюстрированную хорошо отформатированную статью объемом более десятка страниц. Эта статья вполне хорошо оформлена с точки зрения вывода на печать и неплохо, если ее просматривать на экране компьютера во вьювере, снабженном средствами навигации. Но заказчик хочет разместить эту статью в одном из разделов своего сайта. Для этого проще всего преобразовать (конвертировать) Word-документ в Web-страницу средствами самого редактора Word. Так обычно и поступают на скорую руку, хотя это далеко не лучшее решение.

Так, дизайн статьи в формате Word может не гармонировать с общим дизайном сайта, например, по цвету, шрифту и композиции разделов. Разумеется, можно перед конвертированием изменить стиль документа надлежащим образом средствами самого Word. Но это не снимает всех проблем.

HTML-документ, полученный автоматическим конвертированием Word-документа, обычно оказывается очень избыточным за счет слишком большого объема форматирующей стилевой информации. Содержательная информация в нем перемешана со служебной чуть ли не в одинаковой пропорции; структуру статьи в таком представлении трудно разглядеть, не говоря уже о редактировании ее содержания. В данном случае следует в текстовом редакторе создать содержательно пустой HTML-документ и перенести в него посредством буфера обмена статью из редактора Word, а затем разметить ее HTML-дескрипторами (тегами), указав заголовки, абзацы и места, требующие особого форматирования (например, выделения). Окончательный вид текста статьи следует задать с помощью параметров CSS (шрифты, цвета и др.) для заголовков различных уровней, основного текста и особо выделенных мест.

Статью объемом в несколько страниц не назовешь большой, если она на бумажном носителе. При публикации в Интернете ее объем лучше измерять в экранных страницах. Экранная страница — это область Web-страницы в полностью развернутом окне браузера, отведенная для показа статьи. Если статья занимает более двух-трех экранных страниц, то ее желательно снабдить средствами навигации. Очень большие статьи размещают не в одном, а в нескольких HTML-файлах. В этом случае система навигации по статье становится необходимой, а не только желательной.

Кроме того, следует проанализировать графические изображения (если они есть) с точки зрения их разрешения, а также соответствующие файлы — с точки зрения графических форматов и объема. Как правило, разрешение графических изображений (а значит, и объем файлов) можно уменьшить в несколько раз, сохранив достаточный уровень качества. Формат файлов изображений должен входить в число широко употребимых в Web-публикациях: JPEG, PNG, GIF, SWF и некоторые другие.

Аналогичным образом дело обстоит с видео и аудио, которые заказчик собирается опубликовать на своем сайте. Часто оказывается, что исходные материалы представле-

ны не в том формате, который нужен. Например, файлы видеороликов, полученные с помощью видеокамеры, обычно требуется конвертировать в формат потокового видео (например, FLV), если требуется демонстрировать их на странице сайта, не дожидаясь окончания загрузки, и не во внешнем проигрывателе, которого может и не быть у пользователя.

1.6. Тестирование и опытная эксплуатация

Отдельные страницы и сайт в целом неоднократно тестируются практически на всех этапах процесса их разработки. Тем не менее требуется выполнить тщательное генеральное тестирование сайта в режиме онлайн прежде, чем он будет передан заказчику. Помимо прочего, следует особо проверить следующее:

- внешний вид страниц сайта на дисплеях с различными разрешением и глубиной цвета. Типичные разрешения по горизонтали в настоящее время — 1024 и 1280 пикселов, но неплохо проверить как при меньших (800), так и при больших (1366, 1680) разрешениях. Следует обратить особое внимание на возможные изменения относительного расположения элементов внутри страниц, а также на положение страниц относительно краев окна браузера.

  Типичная глубина цвета — 24 и 32 бита, но неплохо проверить и при 8 битах (256 цветов). Если вы использовали цвета не из Web-палитры, то на дисплеях с малой цветовой глубиной возможны существенные искажения графики и цветов текстовых элементов;

- правильность функционирования всех ссылок, как внутренних (на страницы своего сайта), так и внешних (на страницы других сайтов). Если обнаруживается, что какая-то страница перехода отсутствует, то следует создать ее содержательно пустой аналог с сообщением вроде "Данная страница находится в разработке". Это не самое лучшее решение. Лучше не помещать ссылку на неготовую страницу совсем или отложить публикацию сайта;

- правильность функционирования всех скриптов и поведение сайта в случае, когда выполнение скриптов отключено в настройках браузера;

- скорость загрузки страниц сайта при небольшой и средней пропускной способности канала связи (1—5 Мбит/с). Хорошо, если время загрузки страниц не превышает нескольких секунд. Для оценки времени загрузки и других параметров можно воспользоваться сайтами, предоставляющими сервис по анализу указанных посетителем опубликованных сайтов, например, **http://www.cy-pr.com/analysis**;

- межбраузерную совместимость (инвариантность) внешнего вида и функционирования сайта. Данную проверку нужно выполнить хотя бы для трех-четырех ведущих браузеров — Mozilla Firefox, Opera, Google Chrome и Microsoft Internet Explorer (для версии 8 и самой свежей). Напомню, что версия 8 Internet Explorer — последняя, поддерживаемая Windows XP/Vista, а версия 9 поддерживается Windows 7/Vista. Возможно, что в различных браузерах некоторые части вашего сайта будут выглядеть и/или функционировать не одинаково. Если вы не можете это устранить, то предусмотрите альтернативные замены элементов для "неадекватных" браузеров или хотя бы предупредите об этом посетителей. Подумайте, а не исключить ли неинвариантные элементы совсем или хотя бы до лучших времен: новые версии брау-