

Кирилл Панфилов

По ту сторону веб-страницы

Кирилл Панфилов

По ту сторону веб-страницы



Москва, 2008

УДК 004.4
ББК 32.973.26-018.2

Панфилов К.
П16 По ту сторону веб-страницы. — М.: ДМК Пресс, 2008. — 440 с.: ил.

ISBN 5-94074-392-7

Как создать сайт в одиночку, не упустив ни одной важной детали? Или, по крайней мере, проконтролировать весь процесс? На что нужно обращать внимание, а чего, наоборот, нужно остерегаться? Книга рассказывает о процессе разработки сайтов от замысла и проектирования до создания дизайна и кода. Особое внимание уделено проблемам современного веб-дизайна и нестандартным путям их решения.

Почему 95% дизайнеров рисуют похожие сайты? Потому что не хватает фантазии или знаний? Или потому что так предписывает Нильсен? Но существуют типы верстки, совершенно не похожие на привычные, но не менее удобные.

Что такое «мода в веб-дизайне»? Так ли необходимо ей следовать? И всегда ли нужно использовать инструменты, которые используют все? Не всегда для хранения информации удобны базы данных, и не для всех типов графики идеально подходит Adobe Photoshop.

А знаете ли вы, что языки серверного программирования — это пластилин, из которого можно лепить любые фигуры, были бы умения и фантазия. Создатели этих языков сами не догадывались, какие возможности заложены в их детища. В книге рассказывается о небольшой части этих возможностей. Знания языков HTML, XHTML, CSS, JavaScript, PHP и технологий SSI и RSS (их описания есть в книге) недостаточно: в первую очередь нужно понимать, что вы собираетесь сделать, а инструмент найти — дело второе.

Книга — по большому счету — об одном. Прежде чем сделать сайт, нужно подумать, зачем он нужен и как его задачи соотносятся с его внешним видом и его работоспособностью.

УДК 004.4
ББК 32.973.26-018.2

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 5-94074-392-7

© Панфилов К., 2007
© Оформление, ДМК Пресс, 2008



НАВИГАТОР

HTML 4 128

HTML 5 126

XHTML 124

CSS 147

RSS 242

SSI 279

PHP 283

JavaScript 325

— В книге, помимо правил, рекомендаций, сведений справочного и исторического характера, описания новинок, обзоров технологий, есть краткие, но вполне достаточные описания языков HTML, XHTML, CSS, JavaScript, PHP, а также технологий SSI и RSS с примерами кода.

— Большинство примеров из книги (в том числе примеры кода) будет размещено на странице www.erlang.com.ru/book

— Если по адресу, указанному в книге, какой-либо сайт уже не существует, воспользуйтесь сервисом web.archive.org — он является крупнейшим электронным музеем сайтов. На сайте можно найти разные состояния сайтов за различные годы.

Словарь 404

Ссылки 400

Содержание

0. Вводная часть **8**

- 0.1. Краткая история интернета 9
- 0.2. Инструментарий 11
- 0.3. Процесс 20

1. Проектирование **27**

- 1.1. В сторону смысла: форма и содержание 28
- 1.2. Избыточность и недостаточность 32
- 1.3. Вавилонские сайты 39
- 1.4. Два слова о юзабилити 40
- 1.5. Признаки вечного начинающего 41
- 1.6. Поле деятельности: доменное имя, хостинг и тестовая площадка 44
- 1.7. Страница 404 на реконструкции 59

2. Дизайн **64**

- 2.1. Полюсы дизайна: «у нас» и «у них» 67
- 2.2. Что хорошего в шаблонном решении 69
- 2.3. Как изобрести велосипед: ширина сайта 80
- 2.4. Как изобрести велосипед — 2:
навигационное меню 82
- 2.5. Приятное с полезным: осмысленный подход
к дизайну 85

2.6. Рисунок, живопись, композиция	87
2.7. Флэш против дизайна	101
2.8. Графика против дизайна	106
2.9. Сложный вопрос авторских прав	115
3. Верстка	121
3.1. Техминимум: как сейчас принято	122
3.2. Стопроцентная верстка	161
3.3. Тяжелое полиграфическое наследие	187
3.4. Верстка грамотного текста	197
3.5. Браузер для пользователя и веб-дизайнера	205
3.6. Подписано в печать: версии страниц	217
4. Программирование	232
4.1. Что модно и что можно	233
4.2. Что нужно. Организация структуры	258
4.3. Сайт выполняет черную работу	275
4.4. Сайт следит за вами: JavaScript	324
4.5. Сайт располагается на одной странице	367
4.6. Сайт выходит из-под контроля	368
4.7. Очеловечиваем языки программирования	374
4.8. Филиппика против баз данных	377
4.9. Польза настроечных файлов	387
5. Приложения	390
5.1. Алфавит от Google	390
5.2. Справочник для внутреннего использования	391
5.3. Что почитать	396
5.4. Собрание аннотированных ссылок	400
5.5. Словарь терминов	404

0.

Вводная часть

Если уже есть десятки отличных и сотни других книг по веб-дизайну, то в этой должно быть что-то особенное.

Например, нет смысла переписывать в этой книге все атрибуты тэгов языка HTML, превращая его исключительно в справочник. Рассказывать, как в фотошопе правильно сделать имитацию трехмерного объекта для веб-страницы и как сделать так, чтобы изображение без потери качества и размера стало «весить» в 7 раз меньше.

А стоит рассказать о том, почему в российском веб-дизайне все больше сайтов строится на принципах «тянущейся» верстки, а западные до сих пор предпочитают фиксированные размеры. Рассказать, что плохого в использовании тэга `<div>` и базы данных MySQL при всех их неоспоримых достоинствах. Поведасть, какие достойные альтернативы есть у баз данных и у модной технологии AJAX. И что в целом является модным, что ушло в прошлое, а на чем строится будущее веб-дизайна.

Еще очень важно рассказать о том, что сначала нужно знать, зачем создается сайт, а уж затем создавать его. Эта очевидная истина не принимается во внимание при создании четверти крупных сайтов и девяноста процентов самодельных. А при создании сайта нужно не только представлять, каким он будет, но и отбрасывать большую часть стереотипов и высвобождать свою фантазию. Потому что «так принято» не значит «лучше всего».

Книга — о современном положении дел в веб-дизайне, о том, как решить некоторые его проблемы, и о том, как сделать по-настоящему хороший и необычный сайт: в погоне за стандартами и удобством веб-дизайнеры забывают об эстетической составляющей.

Книга также о том, как сделать сайт не только необычным, но и удобным: в погоне за эстетической составляющей дизайнеры часто забывают о будущем посетителе.

Книга — не только сборник рекомендаций и наблюдений автора, но и краткий справочник. Например, в ней можно найти описания

HTML, XHTML, CSS, PHP и JavaScript, которыми вполне можно пользоваться. Кроме того, описаны технологии SSI, RSS и Atom. Для удобства в книге приводится не только оглавление, но и так называемый «Навигатор», или тематический указатель.

0.1. Краткая история интернета

С течением времени менялась форма передачи данных по сетям интернета, но не менялась суть самого интернета. Сети как таковые были разработаны и созданы для обмена данными между различными компьютерами (логичнее было бы сказать: между людьми, сидящими за разными компьютерами, но в демагогию и казуистику мы ударяться не будем), и с тех пор мало что изменилось. Письма по электронной почте, ночное общение в «аське», торренты, загрузка веб-страниц, закидывание пиратских (и не только) версий программ и аудиофайлов на свой компьютер, проверка лицензионности Windows посредством веб-сайта, оповещение о новых заказах в интернет-магазине, ролик любимого исполнителя на сайте, обновление программы с официального сайта — все это передача данных по сетям. Но это современное положение дел.

В 1991 году был разработан язык HTML, без использования которого или потомков которого не обходится и еще долго не будет обходиться ни одна веб-страница. Даже если в основу дизайна сайта положен флэш, ролики как-то нужно включать на страницы: для этого используется HTML. Какими бы яростными ни были войны между последователями верстки на таблицах и верстки на блоках, оба типа верстки — это HTML. Он является основой для любой страницы, размещенной в веб-пространстве, за исключением страниц, построенных на XHTML, основой для которого все равно послужил HTML. И даже если данные для страницы записаны в формате XML или вообще в базе данных, на страницу они все равно выводятся в формате HTML.

С 1993 года, когда был разработан HTML 1.2, до настоящего времени прошла целая эпоха в том плане, что принципы верстки сильно изменились. Первые версии языка позволяли осуществлять только логическую разметку, едва позволяя касаться внешнего оформления. Обычно внешний вид отдавался на откуп браузерам: с самого начала были выработаны рекомендации по внешнему отображению логических элементов веб-страниц, большинства из которых производители браузеров придерживаются и до сих пор. Но если в самом начале выбора у веб-мастеров не оставалось, и им приходилось довольствоваться

тем, что дают, то в 1995 году ситуация изменилась с разработкой третьей версии языка HTML, в которую было включено большое количество тэгов визуального форматирования. Концепция HTML была нарушена: логический и визуальный код представляли такой сплав, который было трудно разделить. Но именно в это время появляются спецификации CSS и их зачаточная поддержка браузерами. Поскольку поддержка была именно зачаточной, никто о чистоте кода и не думал, и все продолжали верстать текст с помощью тэгов ``, а уж о точном позиционировании элементов мало кто думал (разве что приверженцы Netscape). И только в последние несколько лет началась широкомасштабная, поглощающая ресурсы и истощающая силы борьба за чистый код. Дело в том, что CSS позволяет полностью контролировать внешний вид почти любого элемента (за мелкими исключениями), а значит, больше не нужны громоздкие конструкции вроде `<i><u>Текст</u></i>`, когда то же самое можно записать в стилевом определении один раз, а потом повторять бесконечно, применяя, например, к тэгу ``.

Итак, происходит возврат к логической разметке (правда, к этому пришли еще не все кодеры, да и придут несоро, потому что, чтобы сверстать страницу нестандартно, все равно приходится прибегать к ухищрениям), но уже с применением дополнительной технологии. Но меняется не только это. За более чем 15 лет развития веб-дизайна после долгой и упорной работы производителей браузеров, наконец, прочно вошла в жизнь объектная модель документов и хорошая поддержка активных сценариев, что позволяет управлять внешним видом страницы в режиме реального времени, перемещать фрагменты страницы, скрывать их и создавать «на лету», производить проверки, выводить подсказки и совершать иные действия — проявлять реакцию на действия пользователя.

Получили мощное развитие и популярность благодаря очевидным удобствам серверные сценарии на языках Perl, PHP, ASP и некоторых других. Массу действий, которые приходилось выполнять вручную, теперь можно поручить сценариям. А в последние годы популярность приобрела и смешанная серверно-клиентская технология AJAX.

Буйным цветом цветет на страницах мультимедиа: кроме звука, фонового или полезного, веб-дизайнеры получили возможность включать на веб-страницы видео и анимацию в различных форматах (включая Flash). Что уж говорить об изображениях.

Часть технологий уходит в прошлое, едва появившись, часть остается надолго. Основой любой веб-страницы все же остается текст, и к нему нужно проявлять особое внимание.

Факторы, позволяющие внешней стороне веб-дизайна развиваться, — это люди и браузеры. Как правило, чем больше есть, тем больше хочется. Как только появляется новая возможность, появляются новые