

веб-дизайн

книга дмитрия кирсанова



веб.дизайн

книга дмитрия кирсанова

санкт-петербург



2013

Серия «Библиотека дизайнера»

Дмитрий Кирсанов

Веб-дизайн: книга Дмитрия Кирсанова

Главный редактор
Дизайн книги, иллюстрации
Верстка, корректура, индекс

Александр Галунов
Дмитрий Кирсанов
Алина Кирсанова

К43 *Кирсанов Д.*

Веб-дизайн: книга Дмитрия Кирсанова. – СПб: Символ-Плюс, 2013. – 368 с.: цв. ил.

ISBN-10: 5-93286-003-0

Книга автора бестселлера «Факс-модем: от покупки и подключения до выхода в Интернет» – Дмитрия Кирсанова (www.kirsanov.com) – первый полный курс веб-дизайна на русском языке, написанный профессиональным дизайнером. От теоретических основ визуального дизайна до интернетовских технологий и приемов практической работы над сайтом – все это есть в книге, написанной понятно, подробно и увлекательно. Издание будет полезно не только начинающим создателям сайтов, но и дизайнерам, работающим в более традиционных областях, специалистам по рекламе и маркетингу, художникам, программистам, – и, конечно же, всем творческим и любознательным людям.

ISBN-10: 5-93286-003-0

© Дмитрий Кирсанов, 1999

© Издательство Символ-Плюс, 1999

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, д. 7,
тел. (812) 380-5007, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Подписано в печать 18.12.2012. Формат 70х90^{1/16}.

Печать офсетная. Объем 23 печ. л.

Books.Ru

Предисловие (1) Что здесь есть и чего здесь нет (1) Примеры (2) Соглашени (3) Пара слов о словах (4) **О вас** (5) Сверим часы (6) Как выбрать дизайнера (8) **Обо мне** (9) Почему та тема? (10) Сам себе Гутенберг (11) Книга как технологи (12)

Техминимум. (13)

I.1 Кодировки текста (14) **ASCII** (15) **Однобайтовые кодировки** (15) Так и срослось (17) Семейство 8859 (17) **Двухбайтовые кодировки** (18) ISO 10646 и UTF-8 (18)

I.2 HTML (19) **История** (19) В начале был SGML (19) Прикладна философи (21) Золотой век (22) HTML пл с (23) Идолы рынка (23) Б ки и буки (24) Те же и Microsoft (25) Три, четыре... (26) **Синтаксис** (27) Подстановки (28) Минимальный документ (29) **Текстовая разметка** (29) **Ссылки и привязки** (30) **Формы** (30) **Изображения и объекты** (31) **Таблицы** (32) **Фреймы** (32) **Национальные особенности** (32) Согласование кодировок (33) **Среды и доступность** (34) Доступность: таблицы (35) Как писать alt-тексты (35) Доступность: изображени -карты (37) **Мета-данные и поиск** (38) Мертва зона (38) Сухой остаток (39)

I.3 CSS (40) **Принципы** (41) **Возможности** (42) **Модульный HTML** (43) Режим по живому (44) Сборно-панельный сайт (44) Например (46)

I.4 XML (47) **Синтаксис** (48) DTD (48) Уровни соответстви (50) Конверси (51) **Надстройки** (51) XLL (52) XSL (53)

I.5 Графика (55) **Вектор** (55) Вектор в Интернете (57) 3D (59) **Растр** (59) На все четыре стороны (60) Цвета (60) Палитры (61) Системы представлени цвета (62)

I.6 Программирование (63) **JavaScript** (64) Динамический HTML (65) **Модульные технологии** (67) **Динамические страницы** (71)

Основы дизайна. (73)

Как учиться дизайну (75) Дизайн и «чужое творчество» (76) Материалы и инструменты (77)

II.1 Пространственные отношения (78) **Размер** (78) Относительность размера (79) Форма и размер (79) Искусство выравнивани текста (80) Цвет и размер (81) Текстура и размер (81) **Пропорции** (82) Золото и пластмасса (82) Пропорций простота (83) Микропропорции (84) **Размещение** (84) Одноуровневые элементы (85) Разноуровневые элементы (88) **Плотность** (89) Пол и характер (89) Плотность текста (90) В нужное врем и в нужном месте (90)

II.2 Форма (91) **Прямые** (92) Дизайн Пизанской башни (94) **Прямоугольники** (94) **Круги и закругления** (97) **Кривые Безье** (99) **Бесформенность** (100)

ii ■ оглавление

II.3 **Цвет** (101) **Как устроен цвет** (103) **Прогулка по цветовому кругу** (105)
Ну и что? (108) Главные цвета (108) **Восприятие цвета** (109) **Сочета-
емость цветов** (110) **Текст и фон** (113)

II.4 **Текстуры** (116) **Плоский цвет** (117) **Геометрические текстуры** (118)
Пиксельные текстуры (119) **Фотографические текстуры** (119)
Материальные текстуры (120)

II.5 **Шрифт и текст** (122) **Элементы шрифта** (124) Гарнитура, начертание, шрифт (124) Systema naturae (124) Виды начертаний (125) **Шрифты и время** (126) Классическая антиква (127) Переходные шрифты (129) Новая антиква (129) Кларендон (130) Новые рубленые (131) Матник гуманизма (131) **Подбор шрифтов** (133) Единство противоположностей (134) Слишком декоративные шрифты (134) Развива тему (135) Шрифты и название (137) **Параметры набора** (138) Кегль (138) Прописные и строчные (140) Интервалы (141) Выравнивание и отступы (144) Размещение (145) Чем случайность лучше? (146) Цвет (147) **Текст как текстура** (148)

II.6 **Единство** (149) Content is king (150) Академический стиль (150) В полный рост (153)

II.7 **Баланс** (155) Центр масс системы (156) Правило рычага (157)

II.8 **Контраст** (159) Контраст — то продолжение единства другими средствами (160) **Одномерный контраст** (160) **Многомерный контраст** (161) **Аспекты контраста** (163)

II.9 **Динамика** (166) **Динамика явная** (167) Ветер по Гауссу (168) Динамика совсем вна (169) **Динамика неявная** (170)

II.10 **Нюансировка** (171)

Дизайн веб-сайтов. (175)

Преграды и препоны (176) От общего к частному (178)

III.1 **Типы сайтов** (179) Первый шаг (179) В мире .org (180) Скорей л бите нас (181) Бесплатный сыр (182)

III.2 **Устройство сайта** (184) **Неупорядоченный континуум** (184) **Топология сайта** (185) **Распределение материала** (186) Фреймы и окна (188) **Первая страница** (190) Всплеск и вскрик (191)

III.3 **Формат страницы** (192) **The frame of reference** (193) Заметки о полках (194) Во первых строках (195) У голубого крана (196) Если тебе дадут линованную бумагу — пиши поперек (196) **Ступеньки ригидности** (197)

III.4 **Заглавия и заголовки** (199) **Заглавие** (199) **Заголовки: логика** (200) **Заголовки: дизайн** (201) **Разделители** (203) **Баннеры** (204)

III.5 Навигация (205) **Общий обзор** (206) **Семантика** (207) **Дизайн** (208)
Разновидности одинаковости (209) По ту сторону кнопок (210) Символы
и метафоры (210) Особые случаи (212) **Динамическая навигация**
(213)

III.6 Блоки текста (215) **Кегль** (216) **Шрифт** (218) Коренные шрифты (220)
Все шрифты ношу с собой (221) Осторожно: кириллица (222) **Цвет** (223)
Ссылки (224) **Другие параметры** (225) **When all else fails** (227)
Правила набора текстов (228) Расстановка пробелов (228) Нераз-
рываемые пробелы (229) Текстовые выделения (230) Спецсимволы (230)

III.7 Позиционирование (233) **Таблицы и распорки** (234) Дур-
ная наследственность (235) Двойное дно (236) Белое место (237) Фэйсом
об тэйбл (238) **Позиционирование в CSS** (240)

Веб-графика. (243)

IV.1 Техника (244) **Палитра и диффузия** (245) **Цветовые границы** (247)
Экранный анти-алиасинг (249) Из-под пятницы суббота (249) **Пиксель-
ные эффекты** (251) **Оптимизация** (252) JPEG (252) GIF (253)
Взболтать и настоять (254) Чудное мгновение (255) **Blow up** (256)

IV.2 Функции (257) **Выставочная графика** (257) **Фоны** (259) Как не надо де-
лать фон (259) Прогладить с изнанки (261) Одномерные фоны (262) Од-
ним куском (263) **Логотипы** (264) Диалектика формы (266) На единицу
площади (267) Про что картинка? (267) Символ и слово (268) Аспекты ло-
готипа (269) **Баннеры** (275) Сколько натикало? (275) Форматы (277)
В нужное время и в нужном месте (277) Слева при входе (279) Больше
жизни (279) Пьеса в трех действиях (280) В начале было что-то (282) Соб-
ственная гордость (283) Удар ниже пояса (284) Смена поколений (286) На
килограмм живого веса (286) Acknowledgements (287) **Визуал** (288) Ры-
нок визуалов (288)

IV.3 Приемы (290) **Трехмерность** (290) Император иллюзий (291) Пластмассо-
вый призрак (292) Скульптура или барельеф (293) **Искусство искаже-
ний** (295) Что упало, то пропало (295) Нетекстурирующие эффекты (296)
Текстурирующие эффекты (299)

Примеры. (303)

v.1 www.kirsanov.com (304) **Логотип** (304) **Цветовая схема**
(307) Входные цвета (307) Информационные цвета (309) **Первая стра-
ница** (310) Карты на стол (310) **Навигация** (312)

v.2 www.quiotix.com (314) **Логотип** (315) **Первая страница** (317)
Внутренние страницы (320) **Miscellanea** (321)

v.3 Галерея (323)
Индекс (337)

Предисловие

Предлагаемая вашему вниманию

книга задумывалась не как практическое руководство по веб-дизайну, а скорее как учебник общего дизайна на частном материале веб-сайтов. Иначе говоря, я пишу о том, как сделать что-то (в том числе сайт) *красиво*, а не о том, *как* что-то сделать.

Слово «дизайн» в последние годы необычайно девальвировалось. Лишь постепенно интернетовская публика начинает осознавать, что даже с приставкой «веб-» это понятие отнюдь не является синонимом для языка HTML, сетевого программирования на Java или компьютерной верстки. Поэтому в моей книге вы не найдете ни справочника по HTML, ни описаний последних версий Adobe Photoshop или Microsoft FrontPage. Самое трудное в любом деле — не *как*, а *что*, и вместо составления инструкций по тем или иным эффектам в той или иной программе я рассказываю о том, какие оформительские эффекты в принципе возможны, где они уместны и как воспринимаются зрителем.

Конечно, в чистом виде такой сугубо теоретический подход труднореализуем. Чтобы позволить себе роскошь не учить инструкции, дизайнер должен свободно ориентироваться в существующих дизайнерских технологиях и программах. Кроме того, нужно учитывать, что в веб-дизайне «как» сильно влияет на «что»: компьютер не только великолепно справляется с техникой традиционного дизайна, но и открывает множество новых эффектов, новых возможностей, новых способов комбинирования материала. Поэтому дизайнер, не имеющий представления о специфических возможностях компьютера, вряд ли сможет создать на нем что-нибудь конкурентоспособное.

Что здесь есть и чего здесь нет. Книга начинается с обзора основных технологий, имеющих отношение к веб-дизайну: текстовых кодировок, языков разметки (HTML и XML), способов представления графики (векторные и растровые форматы) и технологий Интернет-программирования (сценарии, модули и динамические страницы).

Большинство упоминаемых технологий должны быть (по крайней мере, по названию) знакомы любому, кого способна заинтересовать тема книги. Поэтому цель гл. I — не в том, чтобы обучить читателя каким-то практическим навыкам; не претендует она и на роль полновесного теоретического введения. Моей целью было лишь освежить

2 ■ предисловие

в вашей памяти важнейшие свойства, возможности и ограничения интернетовских технологий, а самое главное — показать их в сравнении друг с другом и в динамике их развития, стимулируя тем самым целостное восприятие, позволяющее ясно увидеть место и роль каждого отдельного языка или стандарта.

Гораздо систематичнее подается в книге курс собственно визуального дизайна, начинающийся в гл. II с обзора абстрактных *материалов* дизайнера (формы, цвета, текстуры, шрифта) и используемых в дизайне *инструментов* (контраста, баланса, динамики, нюансировки). Все эти понятия исследуются как в теории, так и в приложении к практике компьютерного и, в особенности, веб-дизайна. Вместе с тем веб-сайт как объект дизайна имеет множество особенностей, не выводимых из общих принципов и заслуживающих поэтому отдельной главы (гл. III). Наконец, ряд полудизайнерских, полутехнических вопросов, связанных с созданием компьютерной графики, обсуждается в гл. IV.

Такая структура материала основана не на вычитанных где-то схемах, а исключительно на моем собственном опыте практического освоения, теоретического осмысления и последующего преподавания основ дизайна. Три центральные главы книги во многом построены на материале моих ежемесячных англоязычных статей, публикуемых на www.webreference.com/dlab/, — и, разумеется, на основе читательских откликов, вопросов, поправок и дополнений к этим статьям. Конечно, сюжет этого произведения не пытается повторять все зигзаги истории моего дизайнерского самообразования. Однако я старался, чтобы план книги как можно точнее соответствовал сложившейся у меня системе понятий и оценок.

При всем при том, наверное, мне все же не удастся избежать упреков в непоследовательности и эклектичности. Отчасти я пошел на это намеренно; абсолютно симметричные построения нежизненны, и здоровая доля разбросанности, асимметрии и даже нелогичности помогает усвоению материала — при условии, что нелогичность эта художественная, объясняющаяся композиционными соображениями, а не слабым владением материалом. Вместе с тем обилие перекрестных ссылок и подробный предметный указатель сделают, я надеюсь, нетрудным поиск в книге нужной информации.

Примеры. Как любой учебник, эта книга не могла обойтись без примеров обсуждаемых понятий и приемов дизайна. Возможно, кому-то покажется, что примеров этих маловато для книги такого объема. На это я могу возразить, что большинство примеров используются в тексте по несколько раз, в каждом случае иллюстрируя что-то новое. Это и понятно: ведь любая реальная дизайнерская работа пользуется всем арсеналом средств, доступных ее автору,

поэтому при внимательном анализе из нее можно извлечь очень много поучительного.

Но главная причина все же не в этом. Если я пишу книгу о дизайне, я делаю это в предположении, что даже тщательно подобранный визуальный ряд без текста недостаточен для овладения дизайнерским мастерством. Образцы хорошего дизайна время от времени попадают на глаза любому, но даже если вы способны безошибочно отличать их от посредственности, для перехода «пассивного» знания в «активное» недостаточно рассматривания картинок — необходимо их осмысление, результат которого выразить в чисто визуальной форме уже невозможно. Парадоксальное изобилие текста в книге на абсолютно визуальную тему должно, как мне представляется, эффективно работать в том числе и благодаря своей кажущейся нелогичности.

Не слишком много на страницах этой книги и иллюстраций. Мне бы хотелось, чтобы обсуждаемые в тексте принципы каждый читатель иллюстрировал для себя мысленно — виденными им самим образцами дизайна или даже идеями своих собственных композиций. Я пытаюсь не направлять ваше творческое мышление в то или иное русло, а хочу лишь разбудить его не визуальными (и потому, возможно, по-особому эффективными) стимулами. Поэтому я добавлял иллюстрации в основном там, где они действительно необходимы для понимания, и размещал их не в разрыв текста, а на полях.

Определенную пользу начинающий дизайнер смог бы извлечь и из книги, реализующей противоположный подход, — альбома лучших дизайнерских работ, сгруппированных по авторам или по жанрам, с минимумом текста. Такой сборник, наверное, стал бы хорошим дополнением к настоящей книге, однако вряд ли смог бы заменить ее.

Веб-сайты имеют неприятную привычку периодически менять дизайн, а иногда просто исчезать без следа. Поэтому все примеры снабжены иллюстрациями, показывающими, как выглядела обсуждаемая страница в тот момент, когда она попала мне на глаза (стр. 323). В гл. V обсуждаются также несколько моих собственных работ, — о которых я, естественно, могу рассказать намного подробнее и интереснее, чем о любых других, и которые потому пришлось очень кстати для закрепления всего, что изучалось в теоретических главах.

Соглашения. Кое-какие особенности верстки и композиции книги, очевидно, требуют пояснения. Заголовки и подзаголовки идут в тексте без номеров, но снабжены

4 ■ предисловие

номерами в оглавлении (стр. i). Хотя все перекрестные ссылки в тексте используют номера страниц, а не разделов, для облегчения ориентировки в верхнем колонтитуле каждой страницы приведен номер текущего раздела (в форме «глава.раздел.подраздел»). Абзацы, набранные петитом (мелким шрифтом), обычно содержат второстепенный или более узкоспециальный материал, хотя иногда я пользовался этим приемом выделения просто как средством пунктуации для предотвращения монотонности.

Все ссылки — на страницы, номера глав, URL-адреса и т. п. — **выделены** шрифтом и серым цветом. Ссылки вида «пример такой-то» относятся к одной из образцовых веб-страниц, собранных в «Галерее» в конце книги; если позволяет место, каждая ссылка на пример сопровождается черно-белой копией обсуждаемой страницы на полях, а в «почти натуральную» величину все упомянутые в книге сайты-примеры воспроизведены и пронумерованы на стр. 323—333 (частично на цветной вкладке). На цветную вкладку вынесены также несколько иллюстраций.

Кроме иллюстраций, заголовков и копий примеров, на полях изредка попадаются отдельные слова и фразы мелким шрифтом, — как правило, заготовленные в черновике книги, но не нашедшие места в окончательном тексте и в то же время слишком удачные, чтобы от них отказаться. Вы можете относиться к ним, как вам больше нравится, — как к попыткам автора пошутить, еще одному уровню подзаголовков или просто удобному ориентиру для поиска нужного места в тексте методом «сканирования полей». Если все-таки местами поля будут казаться вам слишком пустыми — не стесняйтесь заполнять их своими собственными мыслями и комментариями!

Пара слов о словах — точнее, об используемых терминах. Не один раз мне приходилось придумывать русские названия для вещей, до сих пор существовавших только по-английски (кстати, все переводные термины обязательно сопровождаются английскими оригиналами в скобках), а иногда и вводить обозначения для своих собственных понятий и принципов. Я старался избегать обеих крайностей — и неудобоваримого, в советском стиле официоза (вроде «программное обеспечение» или «узел системы WEB»), и сетевого сленга («мэйл», «хомепага», «чат»), одинаково раздражающего как не знающих английского вообще, так и знающих его хорошо. Вместе с тем я охотно пользовался словечками вроде «скачать», «промотать» и даже «залинковать», привлекательными краткостью и выразительностью русской морфологии (хотя иногда в сочетании с нерусским корнем). Вообще, способ рассказать о чем угодно на правильном и стилистически нейтральном русском языке существует всегда, нужно лишь не лениться поискать его.

Одно из главных технических понятий в этой книге — компьютерная программа для путешествий с сайта на сайт и просмотра веб-страниц — по-русски пишется двояко: *браузер* либо *броузер*. Я придерживаюсь второго варианта, который хотя и менее правилен с точки зрения произношения английского «browser», все же лидирует по количеству употреблений в русскоязычных интернетовских текстах (статистика получена с помощью поисковых систем, стр. 38). Я также хочу заранее попросить прощения у тех, кому могут не понравиться выражения вроде «оба броузера» или «только в одном из броузеров», — относящиеся, само собой, к программам фирм Netscape и Microsoft. В книге, посвященной не Интернету в целом, а лишь визуальному веб-дизайну для массовой аудитории, я счел возможным ограничиться этими двумя самыми распространенными графическими броузерами, вполне сознавая их неидеальность и не-единственность.

брадилка или бродилка?

Что же касается тех людей, которые броузерами пользуются, то для них (если не считать изредка встречающихся метафор типа «посетители сайта», «жители Интернета» и т. п.) мне пришлось принять диковатую кальку «пользователи», — которая, как мне кажется, при всей своей нерусскости все же элегантнее давно уже ставших привычными этажерок вроде «телезрители» или «радиослушатели».

Кому может быть интересна эта книга? Как мне кажется, **О вас** даже такие противоположные читатели, как профессиональный веб-дизайнер с большим опытом и старшеклассник, только что сделавший свою первую страничку, смогут найти в этой книге что-то ценное (или хотя бы не скучное). Основная же часть моей аудитории находится, видимо, где-то посередине между этими двумя крайностями. Какие-то профессиональные знания или навыки в области веб-дизайна желательны, но не обязательны; гораздо важнее иметь запас терпения, усидчивости, а главное — интереса к предмету изучения.

Я надеюсь, что в какой-то своей части книга будет представлять интерес и для тех, кто не собирается пробовать себя в дизайне, но чья работа так или иначе связана с созданием сайтов, — для менеджеров проектов, программистов, специалистов по рекламе и маркетингу. Имеет смысл также рекомендовать это произведение «бумажным» художникам и дизайнерам, желающим попробовать свои силы в веб-дизайне (но, возможно, еще не приступавшим даже к освоению компьютера).

Может быть, вас интересует вопрос — реально ли с помощью этой книги стать профессиональным дизайнером? Если я отвечу положительно, это не будет нескромностью просто потому, что дизайнером можно стать вообще не читая книг. Для этого нужны, помимо врожденной предрасположенности, всего лишь достаточное упорство и методичность в анализе чужих работ и в самостоятельных упражнениях.

6 ■ предисловие

Дизайнера делают не книги, а его собственный талант и трудолюбие. Однако кое в чем, надеюсь, сможет помочь вам и эта книга — обобщение дизайнерского опыта, самым автором почерпнутого из практики, а не из книг или уроков.

Сверим часы. Есть, впрочем, еще несколько обязательных условий для успешной веб-дизайнерской практики. Поскольку опыт большинства моих читателей, вероятно, ограничивается пока созданием страницы для себя или своей фирмы, я начну с очевидного. Прежде чем приниматься за работу, ответьте честно сами себе: действительно ли у вас есть что сказать миру? Будет ли у вашей страницы достаточно посетителей, которым не придется жалеть о потерянном времени? Бессмысленные и бессодержательные сайты плохи не столько сами по себе (в конце концов, заходить на сайт или нет — личное дело каждого), сколько тем, что они снижают планку стандартов, «развращают» даже вполне талантливых сетевых журналистов, писателей и дизайнеров, которым уже не приходится прилагать больших усилий, чтобы блистать на таком фоне.

Сказанное относится, конечно, в первую очередь к русскоязычной части Интернета, незначительные размеры которой (в сравнении с Интернетом всемирным) делают ее похожей на «большую деревню», где все знают все про всех и где любой хоть сколько-нибудь заслуживающий внимания проект немедленно этого внимания утомляется. В англоязычной вселенной страницы любительские и профессиональные давно живут в разных системах координат, и чтобы перейти из первой во вторую, необходим не только и не столько талант, сколько правильная оценка потребностей рынка и хорошая реклама.

Художественные способности, безусловно необходимые дизайнеру, обычно считаются чем-то сугубо врожденным — «талант либо есть, либо нет». Я же убежден в том, что врожденным является скорее *интерес* к этой сфере творчества — интерес, который вряд ли возможно искусственно стимулировать и без которого даже человек с задатками великого художника не сможет эти задатки развить до рабочего состояния. С другой стороны, искренний интерес и желание попробовать себя уже служат залогом того, что при определенном трудолюбии и методичности в упражнениях вы сможете достичь вполне профессионального уровня.

Бытующее представление о невозможности «научить художеству» происходит, боюсь, от того, что мало кто (не исключая и автора этих строк) отчетливо понимает, как и *чему* нужно учить начинающего дизайнера в первую очередь. Помочь здесь может лишь хорошо проработанная *теория* дизайна, которая позволила бы построить логичный и последовательный учебный

курс. На лавры создателя такой теории я не претендую, но в необходимости ее построения я убедился на собственном опыте. Вот почему в этой книге, задумывавшейся как учебник, встречаются элементы стиля научного исследования: чтобы совладать с материалом, мне приходилось на ходу выдвигать и обосновывать гипотезы, искать подтверждения и аналогии, — и я, честно говоря, рассчитываю, что заразительная радость первооткрывателя восполнит то, что педагогический эффект книги теряет из-за неполноты и непоследовательности изложения.

Уровень компьютерной грамотности у веб-дизайнера должен быть, само собой, несколько выше среднепользовательского, но даже для создания веб-сайта от начала до конца совсем не обязательно быть программистом (если это, конечно, не динамический сайт вроде онлайнного магазина, требующий интенсивного программирования). Достаточно представлять себе, чем отличаются друг от друга основные компьютерные технологии и для чего их лучше всего использовать. Главный же, на мой взгляд, признак компьютерно грамотного человека — понимание, чего в принципе можно потребовать от компьютера, а чего нельзя.

Без знания английского в этой профессии, хотя и с трудом, но все же можно обойтись: если программист, не читающий по-английски, быстро потеряет квалификацию, то дизайнеру даже для освоения нерусифицированных программных продуктов вполне может хватить метода «научного тыка» в сочетании с интуицией. Знание основного языка современного Интернета дает, само собой, множество преимуществ: доступность большинства «бумажных» и сетевых дизайнерских публикаций, причастность к международному профессиональному сообществу дизайнеров, — и, конечно же, возможность наслаждаться лучшими образцами англоязычного веб-дизайна во всей их полноте.

Намного полезнее, однако, для российского дизайнера хорошее владение его *родным* языком. Даже работая с заказчиком, хорошо представляющим, что за сайт ему нужен, не стоит рассчитывать на идеальную подготовку текстов: в большинстве случаев дизайнеру приходится самому редактировать материалы, сочинять заголовки и вводные абзацы, формулировать надписи на кнопках навигации и т. п. Пока что немногие из российских дизайн-студий, не говоря уж о дизайнерах-одиночках, могут позволить себе нанимать для этой работы отдельного человека (по-английски его профессия называется *copy writer*). Поэтому отнюдь не гуманитарная (к сожалению) родословная большинства веб-дизайнеров (чаще всего «переквалифицировавшихся в управдомы» из программистов) не лучшим образом

сказывается на удобопонятности, а зачастую и просто грамотности даже приличных по уровню дизайна сайтов.

Как выбрать дизайнера. Может быть, после ряда попыток вы придете к выводу, что дизайн не приносит вам ожидаемого творческого (или меркантильного) удовлетворения, и решите перепоручить создание своего сайта кому-то другому. Как не ошибиться с выбором дизайнера и на что стоит обращать при этом внимание?

Учитывая высокий спрос на создание сайтов и относительно высокую доступность этого ремесла, разборчивому заказчику не мешает для начала убедиться, что предлагающий ему свои услуги — действительно *дизайнер*, а не программист, системный администратор или вебмастер. Участие представителей этих профессий тоже, конечно, может понадобиться при создании сайта, но попытки совместить в одном лице серьезное программирование с серьезным дизайном — повод усомниться в серьезности как одного, так и другого.

Собственно говоря, статический сайт средних размеров вполне под силу дизайнеру-одиночке, и отсутствие у потенциального подрядчика «команды» не должно служить препятствием к сотрудничеству (к тому же при работе «один на один» все вопросы обычно решаются намного оперативнее). Только если вам нужен действительно динамический сайт, строящийся на основе базы данных или сложных интерактивных функций (например, онлайн-магазин), нужно искать студию, состоящую как минимум из дизайнера и программиста.

Свободное владение не только графическими инструментами, но и HTML и JavaScript входит в обязанности веб-дизайнера, хотя в крупных фирмах разделение труда может пойти еще дальше: один делает набросок дизайна, а другой реализует его в HTML. Не соблазнитесь только слишком дешевыми «дизайнерскими» услугами, предлагаемыми обычно фирмами-провайдерами и состоящими *только* в перегонке предоставленного вами материала в HTML, иногда (и даже за отдельную плату!) с украшением его парой графических заголовков или картинок из коллекции бесплатной веб-графики (стр. 153).

Ограничив же круг своих поисков «настоящими» веб-дизайнерами, вы должны оценивать их только по дизайнерским критериям. В идеале ни географическое положение, ни богатство отделки офиса (и даже наличие такового), ни модное имя того или иного дизайнера не должны влиять на

ваш выбор, так же как они вряд ли повлияют на уровень результата. Вас должно заботить только качество работы дизайнера и его соотношение с запрошенной ценой, — и веб-дизайн уникален тем, что впервые позволяет сделать этот идеализированный критерий выбора основным и даже единственным. Здесь, кстати, посильную помощь может оказать и моя книга: даже если вы не собираетесь учиться на дизайнера, вы сможете почерпнуть из нее немало практических критериев, позволяющих отличить качественный дизайн от никакого.

Хорошее дизайнерское портфолио может быть небольшим, но составляющие его работы должны быть тщательно подобраны, прокомментированы и качественно оформлены: если дизайнер не заботится о том, чтобы красиво подать результаты своего труда, особой тщательности в работе для заказчика от него ожидать также не приходится. Стоит спросить у дизайнера контактные адреса его бывших клиентов и произвести среди них небольшой опрос, чтобы выяснить степень их удовлетворенности сотрудничеством с дизайнером, его ответственностью и пунктуальностью.

Что же можно сказать о второй неизвестной в уравнении — цене? Мне не хочется приводить здесь даже приблизительную стоимость дизайна среднего сайта, так как из-за незрелости рынка разброс цен превосходит всякое разумение, и любая названная мною цифра неизбежно навлечет на меня гнев как тех дизайнеров, кто берет (или хотел бы брать) за свою работу больше, так и тех, кто ценит свои услуги дешевле. Отмечу только один любопытный момент. Вероятность прочесть на сайте западного дизайнера о стоимости его услуг, пусть даже с оговорками о приблизительности и предварительности любых цифр, тем меньше, чем дороже он ценит свои услуги и, как правило, чем выше их качество. У российских веб-дизайнеров можно наблюдать противоположную зависимость: некоторые из самых известных наших студий находят особый шик в том, чтобы чуть ли не на первой странице сайта вывесить «нижнюю планку» своих цен, большинством их потенциальных клиентов воспринимающуюся не иначе как юмористически.

Среди множества причин, заставивших меня взяться за эту книгу, не на последнем месте стояло желание доказать себе и другим, что на почти компьютерную тему и на русском языке все-таки возможно написать хорошую книгу. Книгу, в которой не будет пошаговых инструкций, устаревающих с выходом следующей версии программы. Книгу, которую можно читать — и писать! — не торопясь и с удовольствием. Книгу, автор которой знает больше, чем рассказывает, и потому рассказывает связно и интересно — но который, с другой стороны, *сам* дошел до всего, о чем он пишет, и потому знает, что такое настоящее *понимание*

Обо мне

(один из ранних вариантов названия книги — «Понятный HTML», по аналогии с другой моей книгой — «Понятный Интернет») и чем оно отличается от бессодержательной болтовни на модную тему.

Компьютерные книги — особые и в том, как они пишутся, и в том, как читаются. Странный анахронизм в «почти безбумажную» эпоху, этот жанр существует еще слишком недолго, чтобы можно было сказать с уверенностью — наследуют ли компьютерные книги хоть что-то, кроме чисто внешних черт сходства, у книг «обычных»? Лет через 50, когда все эти торопливые описания быстро созданных и быстро и навсегда забытых программ будут громоздиться пыльными пожелтевшими горами, сохранят ли они — не ценность, о нет, но хотя бы способность вызывать умиление, желание раскрыть их и перелистать? Мне остается лишь гадать — и стараться сделать в меру моего разумения так, чтобы книга эта, если и не достигла полной свободы от преходящего и сиюминутного (это невозможно), все же несла бы в себе максимум вневременных ценностей.

У многих есть сейчас подсознательное ощущение, что kaleidosкопически сменяющие друг друга компьютерные технологии и программы заражают своей суетливой эфемерностью все, с чем они соприкасаются, — не только написанные о них книги, но и созданные с их помощью изображения, музыку или тексты. Несомненно, кое-какие из модных ныне приемов и стилей «компьютерного дизайна» будут вызывать через десятилетия лишь улыбку. Но так было всегда — старомодным по прошествии времени не кажется лишь то, что при своем возникновении не делало никаких попыток выглядеть модным. Сама по себе компьютерная техника не несет в себе никакого «вируса неполноценности»; все, как и всегда, зависит от тех, кто ею пользуется.

Почему эта тема? Конечно, потому, что я — дизайнер и знаю это ремесло лучше других. Однако ответ не настолько прост. Почему я *стал* дизайнером, хотя до знакомства с компьютерами и Интернетом никогда не предполагал, что выберу именно эту профессию? Еще ближе к сути будет другая формулировка этого вопроса — почему я *смог* выбрать и самостоятельно освоить профессию, как оказалось, почти идеально отвечающую моим склонностям и способностям?

Хотя разговоры об Интернете как «новой реальности», «среде обитания», «сфере разума» и т. п. давно стали общим местом, для большинства из нас эти символы лишены реального наполнения. Пользователи Интернета хотя и признают некоторое его отличие от, к примеру, телевизора, вряд ли согласятся с тем, что Интернет сколь-нибудь заметно изменил их жизнь. Я же, наоборот, с полной ответственностью могу сказать, что почти всем, что я представляю собой в профессиональном отношении (и уж во всяком случае — всем, что вошло в эту книгу), я обязан именно Интернету.

Лишь будущие историки компьютерной революции смогут в полной мере оценить значение интернетовского взрыва 90-х годов, без которого компьютер так и остался бы не более чем «средством автоматизации офисных работ». Даже далекие от Интернета продукты и технологии не избежали этого животворного влияния: в «бессетевом» мире почти для всех программ, которыми вы сейчас пользуетесь, выбор был бы значительно уже, качество ниже, а цена выше. И я убежден, что влияние этой первой по-настоящему всемирной информационной системы, парадоксально сочетающей свободу и бесплатность доступа с колоссальным коммерческим потенциалом, не ограничится компьютерной индустрией, а изменит в конечном счете жизнь каждого жителя планеты — так же как оно изменило мою.

Отсюда и мое особое отношение к предмету изложения. Я ощущаю себя не потребителем, пассивным зрителем или «пользователем», но деятелем и делателем, в какой-то мере ответственным за то, каким будет Интернет завтрашнего дня. Чтобы разделять это мое отношение, не обязательно быть дизайнером или писателем — просто делая ежесекундный выбор, щелкая по одной ссылке и игнорируя другую, заинтересовываясь или, наоборот, пропуская мимо внимания мелькающие вокруг нас в киберпространстве тексты, образы и послания, каждый из нас прямо и непосредственно влияет на будущее Сети. Поэтому даже если читатель моей книги не станет настоящим дизайнером, но начнет чуть-чуть лучше разбираться в том, что по-настоящему красиво, а что нет и почему, — я посчитаю свою задачу выполненной.

Сам себе Гутенберг. Остается добавить пару слов о том, как делалась эта книга. «Делалась» — не фигура речи; мне здесь принадлежит не только текст, но и все остальное: идея и план, текст и иллюстрации, дизайн и верстка. Если к тому же учесть, что концепция книги не раз кардинально менялась, отражая мои меняющиеся воззрения и интересы, то неудивительно, что от первых набросков текста до тиража прошло больше двух лет — срок для компьютерной книги огромный.

Какие-то из материалов выходили в виде статей в журналах и в Интернете, кое-что рождалось в частной переписке или в дискуссиях. Многие переводилось с русского на английский или наоборот. Кое-что устаревало, отбрасывалось или опровергалось опытом или размышлениями. Некоторые части пухли как на дрожжах, рост их приходилось даже сдерживать; другие, наоборот, вымучивались медленно и с трудом. Развитие книги еще далеко не закончено, и я рассчитываю на то, что в будущих изданиях смогу

использовать, помимо прочего, критику и отклики моих читателей. Пишите мне по адресу: dmitry@kirsanov.com.

Книга как технология. Как в веб-дизайне, так и в дизайне книжном я всегда отдавал предпочтение тем инструментам, которые обеспечивают максимальный контроль над материалом, даже если за это приходится платить временем, потраченным на их освоение. Так же как я не люблю книг, авторы которых уверены, что читатель глупее их, меня раздражают программы, создатели которых пытаются привлечь пользователей не качеством и богатством возможностей, а кажущейся простотой интерфейса («кажущейся» потому, что создаваемые с помощью этой программы объекты — будь то веб-страницы, графика или тексты — станут от этого не *проще*, а только *хуже*).

Весь текст книги набирался в консольном текстовом редакторе Aurora (www.ntext.net) с одновременным внесением разметки в нотации системы $\text{\TeX}$ (www.tug.org). Кроме автоматического контроля правописания spell-чекером Ispell 3.1.20 (ficus-www.cs.ucla.edu/ficus-members/geoff/ispell.html) и многократной вычитки автором и корректором, текст был с большой пользой обработан программой Fresh Eye 1.21 (www.kirsanov.com/fresheye/). Схематические иллюстрации, обложка и часть цветной вкладки делались в программе CorelXARA 2.0 (www.xara.com) и, как и снимки экрана, сохранялись в формате EPS. Стилиевой $\text{\TeX}$ -файл с макроопределениями был создан в самом начале работы, так что большая часть верстки делалась параллельно с писанием текста и созданием иллюстраций. Для компиляции $\text{\TeX}$ -файлов использовался $\text{\emTeX}$ v. 3.14159 [4b] (www.ctan.org/tex-archive/systems/msdos/emtex/) со слегка модифицированным форматом на основе $\text{\LaTeX}$ 2.09 (www.latex-project.org); dvi-файлы сразу же переводились в PostScript с помощью dvips 5.76 (www.radicaleye.com/dvips.html). Промежуточные гранки просматривались в Aladdin GhostScript 5.10 (www.cs.wisc.edu/~ghost/) с GSView 2.4 (www.cs.wisc.edu/~ghost/gsview/) и печатались на принтере HP LaserJet 5 с разрешением 600 dpi; окончательный макет (размером около 1 Мб в виде $\text{\TeX}$ -файлов и свыше 60 Мб в формате PostScript) был напечатан на принтере Lexmark Optra R+ с разрешением 1200 dpi. Любопытные могут ознакомиться со стилиевым $\text{\TeX}$ -файлом и образцом верстки фрагмента книги по адресу www.kirsanov.com/web.design/.

Тех- МИНИ- МУМ.

К сожалению, веб-дизайн еще не достиг той степени развития, на которой технические средства реализации дизайнерских идей отошли бы на второй план. Веб-дизайнер обязан не только свободно ориентироваться во всем спектре технологий современного Интернета, но и сам следить за тем, чтобы его сайты отвечали духу и букве соответствующих стандартов и спецификаций.

Стандарты, регламентирующие разные аспекты интернетовских технологий, уникальны своим не поддающимся никакому учету количеством реализаций (т. е. веб-страниц, сайтов, сообщений электронной почты и т. д.), — и вследствие этого, увы, чрезвычайно низким в среднем уровнем соблюдения как формальных, так и идеологических требований этих стандартов. Вина за это лежит не только на массовой аудитории, в большинстве своем не подозревающей о том, чем, к примеру, корректное использование

HTML отличается от некорректного, но и на множестве наспех написанных программ, генерирующих зачастую не лезущий ни в какие рамки программный или разметочный код. В этих условиях технологическая грамотность веб-дизайнера приобретает особое значение.

Бурно развивающиеся в последние годы технологии, калейдоскопически сменяющиеся версии программ, напряженная борьба компаний, идей и подходов — все это могло бы дать материал для увлекательного техно-триллера толщиной с хороший кирпич, а не для главы в несколько десятков страниц. Я написал лишь небольшое введение, которое позволит вам свободнее ориентироваться в официальных спецификациях, стандартах и других справочных материалах, без чтения которых вам все равно не обойтись.

Кодировки текста.

Отказавшись от претендующего на всеохватность изложения «от теории к практике» и «от простого к сложному», я сразу же попал во власть встречных течений, то и дело отбрасывавших меня назад: «Об этом писать нельзя, пока не объяснено то, а перед этим обязательно нужно рассказать и о том».

Так, поставив на первое место в этой главе технологии разметки текста (HTML и XML), я был вынужден сделать еще шаг назад и начать со стандартов *кодирования* текста в компьютере. Причина не только в особой актуальности этой темы для пользователей, имеющих дело с кириллическим алфавитом; важно также, что она даст нам возможность обсудить некоторые общие принципы передачи и обработки информации в компьютере.

В общем случае *кодировка* (*encoding*), или *кодовая таблица*, — это однозначное соответствие между подмножеством целых чисел (как правило, идущих подряд) и некоторым набором символов. Ключевым здесь является понятие *символа*. Символ может быть буквой (а может и не быть), может соответствовать звуку речи (а может и не соответствовать) и может быть представлен графическим знаком (но может обходиться и без какого бы то ни было видимого образа). Символ — это *атом смысла*, мельчайшая неделимая частица информации. Так, латинское «А» и кириллическое «А» —

это *разные* символы, потому что они употребляются в разных контекстах и несут в себе разную информацию.

Определяющим для любой кодировки является количество охватываемых ею кодов и, соответственно, символов. Поскольку тексты в компьютере хранятся в виде последовательности байтов, большинство кодировок естественным образом распадаются на *однобайтовые*, или восьмибитные, способные закодировать не больше 256 символов, и *двухбайтовые*, или шестнадцатибитные, чья емкость может достигать 65536 знакомест.

Однако прежде чем переходить к восьмибитным кодировкам, нужно сказать несколько слов о кодировке под названием ASCII (American Standard Code for Information Interchange) — кодировке также восьмибитной, но охватывающей только 128 символов и потому довольствующейся семью значимыми битами (старший, восьмой бит при этом всегда равен нулю). Важность этой кодировки, включающей латинский алфавит, цифры и основные знаки пунктуации, необычайно велика: почти все остальные (большие по размеру) кодировки совместимы с ней, т. е. размещают на своих первых 128 знакоместах те же самые символы в том же порядке.

Первые 32 позиции в кодировке ASCII заняты так называемыми *управляющими символами* (*control characters*), предназначенными не для передачи собственно текстовой информации, а для управления устройством, читающим (или получающим по линии связи) текстовый файл. Лишь немногие из этих символов — возврат каретки, перевод строки, табуляция — до сих пор используются в более-менее общепринятых значениях; остальные, давно уже вышедшие из употребления, в былые времена выполняли для «голового» ASCII-текста те же функции, которые сейчас возложены на разнообразные форматы данных и протоколы связи.

ASCII

Задействовав в кодировке ASCII **Однобайтовые кодировки** старший бит, мы получаем дополнительные 128 знакомест, которых должно хватить для кодирования, например, кириллического алфавита или набора каких-нибудь специальных символов. К сожалению, восьмибитных кодировок на свете существует гораздо больше, чем наборов символов, которые они кодируют. Очень характерна в этой связи ситуация с русским языком — анархия компьютеризации в нашей стране, наложившаяся на всемирную анархию

конкурирующих компьютерных платформ и операционных систем, привела к тому, что для кириллицы существует сразу несколько однобайтовых кодовых таблиц.

Хронологически одним из первых стандартов кодирования русских букв на компьютерах был КОИ8 («Код обмена информацией, 8-битный»). Эта кодировка применялась еще в доисторические советские времена на компьютерах ЕС ЭВМ, и когда в середине 80-х появились первые русифицированные версии операционной системы UNIX, они унаследовали эту кодировку у своих «предков». Сеть Релком, открывшая в начале 90-х эпоху российского Интернета, в те годы состояла в основном из компьютеров с UNIX и потому также приняла кодировку КОИ8 в качестве стандартной. В результате КОИ8 является сейчас единственно допустимой кодировкой в русскоязычной электронной почте и телеконференциях Usenet и одной из кодировок, которые обязательно должна поддерживать любая русская веб-страница.

Вторая по значению в русском Интернете (и, безусловно, первая по употребимости на персональных компьютерах) кодировка — это стандартная кириллическая кодировка Microsoft Windows, обозначаемая аббревиатурой CP1251 («CP» расшифровывается как «Code Page», «кодовая страница»). Все Windows-приложения, работающие с русским языком, обязаны понимать эту кодировку без перевода. Благодаря распространенности Windows кодировка CP1251, вместе с КОИ8, входит в абсолютный минимум кодировок, которые приходится поддерживать русскоязычным сайтам.

Реликтом эпохи MS DOS остается так называемая «альтернативная» кодировка, в терминологии фирмы Microsoft — кодировка CP866. И хотя в Интернете компьютеры с MS DOS — большая редкость, кодировка эта сохраняет определенный авторитет благодаря тому, что она принята в качестве стандартной в операционной системе OS/2 и в некоммерческой сети Фидо. Поэтому изредка можно встретить сайты, предлагающие посетителям в качестве одного из вариантов и альтернативную кодировку MS DOS.

Однако первой фирмой, выпустившей русифицированную операционную систему, была все-таки не Microsoft, а Apple. И разумеется, русифицированные Макинтоши, появившиеся в конце 80-х, имели свою собственную, ни с кем не

совместимую кодировку кириллицы. Хотя в нашей стране компьютеры этой марки так и не приобрели популярности, сравнимой с их популярностью на Западе, в Сети можно встретить и кириллическую кодировку для Макинтошей.

Так и спростось. Зачем, однако, вообще понадобилось создавать разные кодировки для одного и того же алфавита? Помимо соображений конкуренции, вообще способствующих появлению несовместимых решений, изобретателями кодировок в первую очередь двигал чисто практический расчет. Как правило, еще до того, как операционная система русифицируется, за символами из верхней половины таблицы ASCII (с кодами от 128 до 255) уже закреплено то или иное употребление, и чтобы разместить в этом же диапазоне кириллический алфавит, приходится чем-то жертвовать. Разумеется, наименее ценные кодовые позиции, замещение которых нанесет меньше всего ущерба функциональности системы, в каждом случае свои, и чаще всего необходимые для полного набора кириллицы 66 знакомест (включая букву «ё», которая традиционно выносится за пределы основного алфавита) не удастся расположить подряд. Этим и объясняется тот факт, что ни в одной из кодировок, кроме ISO 8859-5 и Unicode, русские буквы не идут сплошным блоком. В некоторых кодировках допускаются даже отклонения от алфавитного порядка — в частности, в КОИ8 расположение русских букв определяется, как это ни смешно, алфавитом тех латинских букв, которые на клавиатурах советского производства располагались на одних клавишах с соответствующими буквами кириллицы: например, буква «Ю» стояла на одной клавише с символом «@», идущим в ASCII перед латинской «A», и потому стала в КОИ8 «первой буквой алфавита».

Семейство 8859. Похожая ситуация с конкурирующими платформами и операционными системами и, как следствие, с конкурирующими несовместимыми кодировками наблюдается и в других языках, пользующихся своим собственным алфавитом или даже латинским алфавитом с расширениями. Международная организация по стандартизации (International Standards Organization, ISO) попыталась навести порядок в восьмибитных кодировках, создав серию кодировок ISO 8859, расширяющих таблицу ASCII для латинских букв с диакритикой и лигатур (кодировка ISO 8859-1), кириллицы (ISO 8859-5), арабского (ISO 8859-6), греческого (ISO 8859-7), иврита (ISO 8859-8) и других алфавитов.

Если кодировка ISO 8859-5 для кириллицы так и не прижилась, первая из этой серии — кодировка ISO 8859-1, известная также под именем Latin-1, — сумела стать общепринятым стандартом для кодирования «расширенной» латиницы. В эту кодировку включены почти все символы, употребляющиеся в письменностях западноевропейских языков — французского, немецкого, испанского и т. д.

По аналогии с ASCII первые 32 позиции во второй половине кодировок серии ISO 8859 (коды со 128 по 159 включительно) объявлены «неиспользуемыми». На сей раз, однако, производители программ решили обойти этот запрет. Так, большинство шрифтов для Windows соответствуют кодировке ISO 8859-1 начиная с позиции 160 до конца таблицы, но в диапазоне 128—159 размещают некоторые дополнительные символы (в частности, длинное тире и символ «торговой марки», стр. 233). Поскольку HTML обязан соответствовать стандарту Latin-1 (а начиная с версии 4 — Unicode), числовые подстановки (стр. 29) не могут ссылаться на коды из этого диапазона.

Двухбайтовые кодировки. Языки с иероглифической письменностью (японский, китайский, корейский) пользуются смешанными кодировками, в которых иероглифы (а их в сотни раз больше, чем букв в алфавите) представлены двухбайтовыми кодами, а вставки на латинице кодируются по однобайтовой таблице (обычно совпадающей с Latin-1). Переключение между двухбайтовым и однобайтовым режимами производится специально зарезервированными управляющими символами.

В 1991 году была предпринята попытка создать единую универсальную двухбайтовую кодировку, охватывающую все алфавиты и иероглифические системы мира. Результатом стал стандарт под названием Unicode, покрывающий не только системы письменности всех живых и большинства мертвых языков мира, но и множество музыкальных, математических, химических и прочих символов. Хотя массовое применение Unicode в документах и программах остается делом будущего, для веб-дизайнера эта кодировка имеет особое значение, так как именно она объявлена «стандартной кодировкой документа» в HTML начиная с версии 4 (стр. 32).

ISO 10646 и UTF-8. Предвидя неизбежное рано или поздно исчерпание и двухбайтового кодового пространства (пока еще до этого далеко, так как около 30% кодов в Unicode до сих пор не заняты), ISO уже застолбила стандарт *четырехбайтовой*, совместимой с Unicode кодировки под названием ISO 10646. Пока что вместо этого обозначения, которое то и дело попадает в стандартах, вы можете с чистой совестью подставлять «Unicode», так как никаких новых символов, выходящих за границы совпадающих с Unicode первых 65536 знакомест, в ISO 10646 еще не определено.

По-видимому, в ближайшее время все более важную роль будет играть особый формат Unicode (и ISO 10646) под названием UTF-8. Эта «производная» кодировка пользуется для записи символов цепочками байтов

различной длины (от одного до шести), которые с помощью несложного алгоритма преобразуются в Unicode-коды, причем более употребительным символам соответствуют более короткие цепочки. Главное достоинство этого формата — совместимость с ASCII не только по значениям кодов, но и по количеству бит на символ, так как для кодирования любого из первых 128 символов в UTF-8 достаточно одного байта (хотя, например, для букв кириллицы нужно уже по два байта).

Вместе с XML, которому посвящен следующий раздел, HTML обычно причисляют к «языкам разметки текста». На самом деле роль этих двух языков, как и самого формата под названием «просто текст» («plain text»), выходит далеко за рамки обработки текстовой информации.

HTML

Текстовая часть любой веб-страницы теснейшим образом переплетена с управляющими конструкциями языка HTML, невидимыми сами по себе, но определяющими внешний вид и размещение всех остальных элементов страницы. Таким образом, в первую очередь HTML выполняет роль «скелета» страницы и сайта в целом — на HTML-разметку нанизываются текст, изображения, ссылки, интерактивные элементы и вообще все, что только может быть отображено в окне браузера. Лишь «по совместительству» HTML-файл содержит в себе еще и собственно текстовую часть страницы.

Давайте для начала проследим короткую, но необычайно насыщенную историю HTML — языка, в котором столкнулись противоположные (и, в идеале, взаимодополняющие) подходы к проблеме представления информации в компьютере и который стал в последние годы аренной сотрудничества и противоборства основных действующих лиц «компьютерного театра».

История

В начале был SGML. Начало истории HTML следует отнести к далекому 1969 году, когда Чарльз Гольдфарб, работавший тогда в компании IBM, создал прототип языка для разметки технической документации, впоследствии названного GML, а с приданием ему в 1986 году статуса международного стандарта — SGML (Standard Generalized Markup Language). Этот обобщенный метаязык предназначен для построения систем логической, *структурной* разметки любых разновидностей текстов. Слово «структурная» означает, что управляющие коды, вносимые в текст при такой

разметке, не несут никакой информации о форматировании документа, а лишь указывают границы и соподчинение его составных частей, т. е. задают его *структуру*.

Создатели SGML стремились полностью абстрагироваться от проблем представления текста в разных программах, на разных компьютерных платформах и устройствах вывода. Хотя формально ничто не мешает записать средствами SGML *любую* информацию об элементах документа — в том числе и параметры его форматирования (например, шрифт Times полужирного начертания кегля 12 пунктов для заголовков), — *идеология* этого языка требует ограничиться указанием на уровень заголовка и его место в иерархической структуре документа. Все остальное должно быть вынесено в так называемые стилевые спецификации — совершенно отдельный и, как принято выражаться, *ортогональный* (т. е. допускающий независимое изменение) по отношению к структурной основе информационный «слой».

Благодаря этим ограничениям размеченный текст сможет без труда интерпретировать любая программа, работающая с любым мыслимым устройством вывода. К примеру, при работе в графическом интерфейсе заголовок может действительно выводиться полужирным шрифтом повышенного кегля; программа, использующая текстовый интерфейс, выделит его пустой строкой сверху и снизу и, возможно, повышенной яркостью символов; синтезатор речи, читающий документ вслух, сможет отметить заголовок паузой и изменением интонации; наконец, «робот», собирающий базу, придаст тексту заголовка больший «вес» при контекстном поиске. Можно сказать, что SGML-разметка обнажает нематериальную «душу» текста, для которой впоследствии любая программа-интерпретатор сможет подобрать подходящее к случаю «тело».

Сам по себе SGML есть не готовая система разметки текста, а лишь удобный метаязык, позволяющий строить такие системы для конкретных обстоятельств. Жизнь многообразна и непредсказуема: сегодня вам требуется выделять в текстах заголовки, а завтра, возможно, понадобится размечать подписи в письмах, математические формулы или имена действующих лиц в пьесе. Стандарт SGML устанавливает лишь синтаксис записи элементов разметки, а также правила определения новых элементов и указания структурных отношений между ними. Для практической же разметки документов нужно *приложение* SGML — набор

определений элементов, представляющий собой, по сути, формальное описание структуры документа.

Прикладная философия. Разделение «содержания» и «представления» как двух независимых аспектов информации — идея не особенно новая. Как и другие абстрактные противопоставления, до недавнего времени она оставалась чисто философской концепцией, не имевшей никакого выхода на практику. Вспомним, однако, что задолго до того, как философия смогла сделать свои первые шаги, способность к абстрактному мышлению и поаспектному анализу вещей и явлений должна была возникнуть и оформиться в языке. Лингвистам известно, что у языков, находящихся на начальных стадиях развития, зачастую отсутствует способность к разделению абстрактных аспектов явлений — такой язык может иметь самостоятельное слово для «падающего снега» при полном отсутствии слов для понятий «падать» и «снег» по отдельности. Очевидно, невозможность *сказать* что-то отражает и невозможность это *помыслить*.

К чему я заговорил о языке? Дело в том, что история развития абстрактного мышления в целом — хороший аналог происходящему на наших глазах медленному и трудному процессу вычленения и очищения аспектов компьютерного представления информации. До сих пор подавляющее большинство текстов создаются и хранятся в «фирменных», ориентированных на визуальное представление форматах вроде MS Word, — которые, как языки первобытных племен, неспособны отделить «существительное» содержимого документа от «прилагательного» его представления в той или иной среде.

Собственно говоря, в докомпьютерную эпоху практическая задача разделения аспектов документа вообще не могла стоять. Единственное «преобразование формата», возможное для бумажного документа, — это прочтение его вслух, и очевидно, что доступ к хранящейся отдельно структуре текста не очень-то помог бы в такой ситуации. Совсем другое дело — текст, хранящийся в компьютере, — компьютере, возможности которого уместно сравнить с мощностью хорошо развитого человеческого языка. Как и язык, компьютер способен оформить, сохранить и сообщить вовне любую абстракцию, если только для этой абстракции разработана соответствующая *нотация*, т.е. синтаксис записи.

Именно таким синтаксисом и является язык SGML. Теперь мы можем мыслить содержание и оформление как две не только потенциально, но и реально отдельные друг от друга сущности именно потому, что компьютер позволяет нам «высказать» их по отдельности. Важно отметить, однако, что SGML не есть инструмент для *разделения* содержания и представления, а всего лишь удобное средство хранения уже возогнанной и очищенной структурированной информации. Само же вычленение структуры — в любом случае задача человека: как и язык, компьютер не может мыслить сам, а лишь помогает мыслить человеку.

Не менее важно и то, что в SGML нет никакой изначальной склонности к «содержанию» в ущерб «оформлению»; единственное требование к информации, сохраняемой средствами SGML, — это ее *структурированность*. В виде иерархической структуры вложенных друг в друга элементов вполне можно представить не только содержимое документа, но и набор относящихся к нему правил и параметров оформления (как это и сделано в языке XSL, стр. 53). Собственно говоря, SGML-документ больше всего похож на базу данных с произвольной длиной поля и возможностью установления иерархических отношений между полями. Как и базе данных, SGML-документу все равно, что хранить в себе, лишь бы данные

соответствовали заранее заданной структуре, — смысл которой придан ей человеком и существует только для него.

Как мы видим, компьютер — не только лучший из созданных до сих пор инструментов для *записи* идей, но и единственное изобретение человека, позволяющее *реализовать* на практике многие идеи, до сих пор казавшиеся чистой абстракцией.

Золотой век. Принципы, на которых строится язык SGML, значительны и интересны; несомненно, идеология языка оказала влияние на многие компьютерные разработки. Однако сам по себе SGML не получил сколько-нибудь заметного распространения до тех пор, пока в 1991 г. сотрудники Европейского института физики частиц (CERN), занятые созданием системы передачи гипертекстовой информации через Интернет, не выбрали SGML в качестве основы для нового языка разметки гипертекстовых документов. Этот язык — самое известное из приложений SGML — был назван HTML (HyperText Markup Language, «язык разметки гипертекста»).

Изначально HTML, как и положено SGML-приложению, разделял все особенности идеологии SGML. Из сорока с небольшим тегов HTML версии 1.2 (датированной июнем 1993 г.) всего три, да к тому же и не рекомендованных к использованию, тега осмеливались намекать на физические параметры представления документа. Вся разметка была чисто логической, и лишь в описательной части стандарта, сопровождающей формальное определение тегов, можно было прочесть что-нибудь вроде «в графических браузерах действие этого тега может передаваться курсивным начертанием».

Первым же (и единственным в те далекие времена) графическим браузером была программа Mosaic, разработанная, как и сам WWW, в научном учреждении — Национальном центре суперкомпьютерных приложений США (NCSA). Так что нет ничего удивительного в том, что в этот «золотой век» никаких противоречий между официальными стандартами и их реализацией в браузерах еще не существовало. HTML неторопливо развивался, оставаясь в рамках парадигмы структурной разметки, и в апреле 1994 г. началась подготовка спецификации следующей версии языка — 2.0. Этим занимался образованный в том же году Консорциум W3 (W3 Consortium, сокращенно W3C), перенявший от CERN верховную власть и авторитет в мире WWW.

В настоящий момент консорциум, имеющий статус «международного и некоммерческого», объединяет свыше 150

организаций-членов, в том числе фирмы Netscape, Microsoft и множество других. Однако в 1994—1995 гг. его членами были почти исключительно университеты и научные учреждения. Столь академический состав W3C сказывался как на самих документах, публикуемых консорциумом, так и на процедуре (и особенно на сроках) их принятия. Достаточно сказать, что спецификация HTML 2.0, единственным серьезным усовершенствованием в которой был механизм форм (стр. 30) для отсылки информации с компьютера пользователя на сервер, была окончательно утверждена лишь в сентябре 1995 г., когда в W3C уже полным ходом шло обсуждение HTML 3, — или, как его называли поначалу, «HTML+».

HTML плюс. Пожалуй, проект HTML 3 — самая яркая и неоднозначная страница в истории языка. Работа над ним началась в марте 1995 г., и первоначальный вариант стандарта включал в себя много интересных нововведений — теги для создания таблиц, разметки математических формул, вставки обтекаемых текстом рисунков, примечаний и др. Но самое главное — HTML 3 был попыткой разрешить уже достаточно очевидное к тому времени противоречие между идеологией структурной разметки и потребностями пользователей, заинтересованных в первую очередь в гибких и богатых возможностях визуального представления.

Противоречие это было разрешено опять-таки в полном соответствии с идеологией SGML: W3C ввел в HTML 3 поддержку так называемых иерархических стилевых спецификаций (CSS, стр. 40). Система CSS формально независима от HTML, имеет совершенно иной синтаксис, не наследует никаких идеологических ограничений и позволяет, уже в совершенно иных терминах, задавать параметры графического (так же как и текстового, звукового и какого угодно другого) представления для любого тега HTML.

Нет сомнения, что CSS — почти идеальный способ избавить HTML от наследственных дефектов и перевести его развитие на принципиально новые рельсы. Тем досаднее то, как сложилась судьба этого замечательного изобретения. Поскольку спецификацию CSS увязали с другими нововведениями HTML 3, W3C долго не утверждал ее в качестве официального стандарта; задерживалось и доведение ее до более или менее завершенного вида, при котором стала бы возможной реализация CSS в коммерческих продуктах.

Идолы рынка. А между тем коммерческое освоение WWW не заставило себя долго ждать. В начале 1994 г. группа разработчиков броузера Mosaic основала корпорацию Netscape Communications и вскоре выпустила первую версию коммерческого броузера Netscape (начиная с версии 2.0 — Netscape Navigator, а с версии 4.0 — Netscape Communicator). С этого момента начался экспоненциальный рост WWW, продолжающийся по сей день. Чтобы закрепить лидерство (на которое, впрочем, тогда еще мало кто покушался) и привлечь новых пользователей, Netscape вводила в HTML все новые и новые усовершенствования, — поддерживаемые, разумеется, *только* броузером Netscape.

Практически все новые теги, без устали изобретаемые Netscape, были направлены на улучшение внешнего вида документа и расширение возможностей его форматирования. Причины понятны: чтобы убедить, скажем, бизнесмена, что ему пора обратить внимание на некую новую технологию, прежде всего нужно показать ему ее в привлекательном, «товарном» виде. Поставив себе целью завоевание корпоративного рынка, разработчики из Netscape не могли (да и не хотели, по-видимому) уделять должное внимание сложившимся традициям развития языка. В результате тот вариант HTML, который поддерживала выпущенная в начале 1996 г. версия Netscape Navigator 2.0, представлял собой довольно странную смесь старых логических тегов с беззастенчиво вломившимися новыми, ориентированными на графическое экранное представление документа и затрудняющими его воспроизведение на других устройствах вывода.

Бяки и буки. Такая политика компании, с одной стороны, принесла ей быстрый и впечатляющий успех (одно время версии Netscape Navigator составляли более 90% всех используемых броузеров), а с другой — вызвала ожесточенное сопротивление наиболее сознательной части HTML-сообщества. Энтузиасты неустанно разъясняли и разъясняют каждому, кто согласен их слушать, что HTML по природе своей не имеет права зависеть от какого-то конкретного броузера и что заявления типа «эту страницу лучше всего смотреть в Netscape Navigator» являются просто насмешкой над здравым смыслом.

Помимо использования «плохих» Netscape-ориентированных тегов, широко распространена также практика «злоупотребления» средствами HTML ради сопутствующих им

визуальных эффектов. Скажем, вместо того чтобы использовать тег `TABLE` только для оформления «настоящих» таблиц, веб-дизайнеры сплошь и рядом применяют таблицы без рамок для построения графического «каркаса» страницы (стр. 234). Нет нужды объяснять, что оформление в виде таблицы того, что по сути своей таблицей совсем не является, есть прямое нарушение идеологии структурной разметки.

С другой стороны, если все-таки признать за HTML (в отсутствие CSS) право совмещать в себе содержательный и оформительский аспекты разметки, то можно, придерживаясь ряда несложных правил (стр. 35), создавать страницы, великолепно выглядящие в графических браузерах и притом вполне доступные в любых неграфических средах.

Те же и Microsoft. В конце 1995 г. ситуация в мире HTML была довольно смутной. Популярность браузера Netscape неуклонно росла; программисты этой фирмы готовили к выпуску версию 2.0, которая должна была утвердить господство Netscape на вечные времена благодаря неслыханному набору новшеств (интерфейс подключаемых модулей, поддержка Java-апплетов, встроенный язык сценариев JavaScript, возможность разбишки окна на фреймы и многое другое). К этому времени W3C окончательно завяз в своем проекте HTML 3, который был слишком сильно оторван от реальности и на завершение которого у консорциума попросту не хватало ни сил, ни средств. HTML 3 по сравнению с HTML 2.0 был важным шагом вперед, однако он развивался по-прежнему в рамках идеологии структурной разметки, а инструмент, дающий возможность выйти за эти рамки, — система CSS — был еще далек от завершения.

В этот переломный момент в игру вступил новый участник — корпорация Microsoft. Долгое время эта компания, привыкшая монопольно владеть своим сектором рынка, недооценивала перспективы Интернета и не собиралась как-либо участвовать в развитии этой информационной среды. Однако невероятный взлет Netscape (число копий браузера Navigator измерялось к этому времени уже десятками миллионов) заставил Microsoft изменить свое мнение.

И именно на браузерном фронте, где господство Netscape оставляло меньше всего шансов конкурентам, корпорация Microsoft нанесла свой главный удар. Поначалу мало кто верил, что браузер Microsoft Internet Explorer, который тогда существовал в версии 2.0 и не представлял собой ничего

выдающегося, сможет составить конкуренцию Netscape. Тем не менее выпущенная летом 1996 г. версия Internet Explorer 3.0, которая поддерживала почти все расширения Netscape, вызвала настоящий бум и очень быстро утвердилась в качестве «второго главного броузера». Сейчас Microsoft и Netscape делят рынок броузеров почти поровну, и окончательный исход их битвы не берется предсказать никто.

Несколькими ловкими ударами поставив свой броузер на один уровень с казавшимся некогда непобедимым Netscape, корпорация Microsoft, очевидно, не собирается останавливаться на достигнутом. Но еще интереснее то, что Microsoft при этом пытается создать для себя новый имидж — компании, поддерживающей независимые организации вроде W3C и заботящейся об авторитете официальных стандартов не меньше, чем о своей выгоде. На этом фоне Netscape, еще недавно имевшая репутацию главного генератора идей и технологического локомотива всей Интернет-индустрии, начинает казаться слишком закрытой, негибкой и эгоистичной в своих намерениях. В действительности же стратегия Microsoft (как и незадолго до этого Netscape) заключается в том, чтобы, объявив официально о поддержке какого-то открытого стандарта, немедленно «улучшить» его расширениями, поддерживаемыми только в продуктах Microsoft, добиться признания этих расширений де-факто частью стандарта — и тем самым установить контроль как над самим стандартом, так и над соответствующим сегментом рынка.

Очевидно, чувствуя потерю инициативы, корпорация Netscape решилась весной 1998 г. на беспрецедентный шаг — опубликовала исходные тексты своего броузера на сайте www.mozilla.org и пригласила всех желающих программистов и тестеров принять на некоммерческой основе участие в подготовке следующей версии. Как это ни странно, именно работающие из чистой «любви к искусству» энтузиасты создали многие свободно распространяемые и пользующиеся притом огромной популярностью программы (в их числе даже целая операционная система — Linux), и Netscape явно не прочь подзарядиться новыми силами и идеями из этого неисчерпаемого и почти бесплатного источника. По некоторым сведениям, не коммерческих конкурентов, а именно «открытые» программы со свободно распространяемым исходным кодом Microsoft считает главной угрозой для своего могущества.

Три, четыре... Одновременно с разработкой конкурентоспособного броузера Microsoft решила «навести порядок» и в мире HTML. Взяв под свою опеку W3C, она наплатила его денежными и людскими ресурсами и тем самым заработала право едва ли не решающего голоса в этой организации. Проект HTML 3 был заморожен, а вместо него в сжатые сроки создан стандарт HTML 3.2, который, по сути, всего лишь описывает большинство расширений Netscape (с тем же успехом их можно назвать теперь «расширениями Microsoft»). Пройдя обычный в W3C процесс обсуждения и внесения поправок, спецификация HTML 3.2 была утверждена в январе 1997 года. Спираль развития

HTML завершила свой первый виток — как и в «золотой век», расхождения между предписаниями стандарта и реализацией HTML в браузерах вновь были сведены к минимуму.

В декабре того же 1997 г., с принятием стандарта HTML 4.0, маятник, похоже, качнулся уже в обратную сторону — наряду с дальнейшим обогащением репертуара визуальных тегов, эта версия ввела немало пусть и не вполне «логических», но очень важных расширений для поддержки многоязычных документов (стр. 32) и обеспечения доступности документа в разных средах (стр. 34). Кроме того, в HTML 4 наконец-то прямо в тексте стандарта четко проведено разделение логических и визуальных тегов (последние объявлены «нерекомендованными», «deprecated»). Кстати, объем спецификации HTML 4 (которую я советую прочесть всем, кто имеет хоть какое-то отношение к веб-дизайну) в несколько раз больше, чем у 3.2, в основном не за счет описания новых тегов, а благодаря гораздо более подробному обоснованию целей и идеологии языка — так, в спецификацию включен даже краткий курс SGML и разбор HTML DTD.

Многие считают, что язык HTML исчерпал потенциал своего развития и что добавление новых тегов вряд ли выведет его на принципиально иной уровень. История HTML, полная борьбы и противоречий, по-видимому, близится к завершению. Точнее, подошла к концу история его *развития*, так как применяться в более или менее неизменном (и, по-видимому, близком к современному) виде он будет еще долго — ведь в мире накоплено огромное количество ресурсов, жестко привязанных к этому языку. Очень хочется надеяться на то, что наследником HTML станет XML (стр. 47) — язык, гораздо более близкий по идеологии к SGML и в то же время достаточно простой для массового применения.

Врожденные и теперь уже вряд ли исправимые недостатки HTML особенно очевидны для тех, кто занимается практическим веб-дизайном: из-за того, что HTML с самого начала не был рассчитан на описание внешнего вида документа, он не в состоянии удовлетворительно выполнить эту задачу даже сейчас, при наличии множества визуально-ориентированных тегов. Прямым следствием этого является огромное количество расхождений в интерпретации тегов браузерами. Как бы строго вы ни следовали стандарту, HTML-файл приходится обязательно тестировать по меньшей мере в графических браузерах фирм Netscape и Microsoft, и чаще всего такое тестирование не обходится без неприятных сюрпризов: отступы, пробелы, размеры элементов оформления и логика их размещения на странице даже для простейших тегов различаются довольно сильно.

Изучение любого компьютерного языка начинается со знакомства с его основными строительными блоками — операторами, выражениями, переменными. С этой точки зрения язык HTML крайне прост, чтобы не сказать —

Синтаксис