



Донецкий национальный
технический университет

А. Я. Аноприенко, С. В. Иваница, Т. В. Завадская

Интернет-технологии для студентов и преподавателей

-  Графическая информация в Интернете
-  Программное обеспечение для просмотра и верстки сайтов
-  В помощь веб-разработчику



Донецкий национальный
технический университет

А. Я. Аноприенко ♦ С. В. Иваница ♦ Т. В. Завадская

ИНТЕРНЕТ-ТЕХНОЛОГИИ ДЛЯ СТУДЕНТОВ И ПРЕПОДАВАТЕЛЕЙ

Учебное пособие

КНИГА ПЕРВАЯ

-  ГРАФИЧЕСКАЯ ИНФОРМАЦИЯ В ИНТЕРНЕТЕ
 -  ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ ПРОСМОТРА И ВЕРСТКИ САЙТОВ
 -  В ПОМОЩЬ ВЕБ-РАЗРАБОТЧИКУ
-

*Издание приурочено к 95-летию
Донецкого национального технического университета*

ООО «Технопарк ДонГТУ «УНИТЕХ»
Донецк ♦ 2015

УДК 004.738.5(07)

ББК 32.973.202

А69

*Рекомендовано Ученым советом
ГВУЗ «Донецкий национальный технический университет» (г. Донецк),
протокол № 9 от 18 декабря 2015 года.*

Рецензенты:

Л. П. Полякова — доктор наук по государственному управлению, профессор, министр образования и науки Донецкой Народной Республики;

В. К. Толстых — доктор физико-математических наук, доктор технических наук, профессор, заведующий кафедрой компьютерных технологий Донецкого национального университета (г. Донецк);

К. Н. Маренич — доктор технических наук, профессор, заведующий кафедрой «Горная электротехника и автоматика имени Р. М. Лейбова», проректор по научной работе Донецкого национального технического университета (г. Донецк).

Аноприенко А. Я.

А69 Интернет-технологии для студентов и преподавателей: учебное пособие: книга первая / А. Я. Аноприенко, С. В. Иваница, Т. В. Завадская. — Донецк: ООО «Технопарк ДонГТУ «УНИТЕХ», 2015. — 260 с.: ил.

Книга написана на основе материала курса «Интернет-технологии» для магистрантов, читаемого в Донецком национальном техническом университете с 2000 года.

Эта книга — первая из серии «Интернет-технологии для студентов и преподавателей», в которую вошел материал о веб-дизайне, веб-типографике, о работе с графической информацией в Интернете и программном обеспечении для работы с сайтами.

Книга предназначена для старших школьников и студентов, магистрантов и аспирантов, молодых ученых, преподавателей, а также для всех кто интересуется интернет-технологиями.

ISBN 978-966-8248-55-9

УДК 004.738.5(07)

ББК 32.973.202

© Аноприенко А. Я., 2015

© Иваница С. В., 2015

© Завадская Т. В., 2015

© ГВУЗ «Донецкий национальный технический университет», 2015

Содержание

Содержание	3
Введение	7
♥ Глава 1. Графическая информация в Интернете.....	11
1.1. Растровая и векторная компьютерная графика	11
1.1.1. Растровая графика	13
1.1.2. Векторная графика	15
Контрольные вопросы.....	18
1.2. Цветовые модели. Работа с цветом	19
1.2.1. Модель RGB	21
1.2.2. Модель CMYK	22
1.2.3. Модель HSB.....	23
1.2.4. Модель HSL	25
1.2.5. Модель Lab	26
Контрольные вопросы.....	29
1.3. Хранение графической информации	30
1.3.1. Глубина цвета	30
1.3.2. Методы сжатия графической информации	31
1.3.3. Основные форматы файлов для хранения графической информации.....	40
Контрольные вопросы.....	50
Список литературы к главе 1	51

♥ Глава 2. Программное обеспечение для просмотра и верстки сайтов	55
2.1. Браузеры.....	55
2.1.1. История развития. Браузерные войны	55
2.1.2. Современные браузеры. Достоинства и недостатки, рыночные доли.....	64
Контрольные вопросы.....	78
2.2. Программы для верстки веб-страниц	79
2.2.1. Визуальные редакторы.....	81
2.2.2. Специализированные текстовые редакторы	86
Контрольные вопросы.....	94
2.3. Программы для обработки графических файлов	95
2.3.1. Растровые графические редакторы.....	96
2.3.2. Векторные графические редакторы	102
2.3.3. Программы для создания анимации	106
Контрольные вопросы.....	108
Список литературы к главе 2	109
♥ Глава 3. В помощь веб-разработчику	115
3.1. Основы веб-дизайна	115
3.1.1. Общие характеристики шрифтов.....	116
3.1.2. Правильный подбор шрифта	121
3.1.3. Компьютерные шрифты	124
3.1.4. Шрифтовые онлайн-сервисы.....	132
3.1.5. Что нужно знать о цвете.....	140
3.1.6. Цветовые сочетания	148
3.1.7. Инструментарий веб-дизайнера для работы с цветом.....	157
Контрольные вопросы.....	168
3.2. Основы веб-типографики	170
3.2.1. Типографское форматирование текста.....	172

3.2.2. Гарнитура, размер и начертание шрифта	175
3.2.3. Капиталь	185
3.2.4. Втяжка и отбивка текстов	186
3.2.5. Заголовки текста	193
3.2.6. Выключка и отступ	196
3.2.7. Линейки	202
3.2.8. Способы выделения элементов текста	206
3.2.9. Сервисы автоматического форматирования веб-текстов	210
Контрольные вопросы	213
3.3. Специальные символы в веб-текстах	215
3.3.1. Использование специальных символов	215
3.3.2. Типографские знаки	217
3.3.3. Пробельные символы	220
3.3.4. Дефис, тире и прочие черточки	222
3.3.5. Кавычки и штрихи	224
3.3.6. Обязательные мнемоники HTML	227
Контрольные вопросы	230
3.4. Основы грамматики	231
3.4.1. Пробелы	232
3.4.2. Знаки препинания	234
3.4.3. Дефис и тире	236
3.4.4. Числа и математические знаки	238
3.4.5. Специальные знаки	242
3.4.6. Сокращения	243
3.4.7. Переносы слов и строк	245
3.4.8. Текстовые выделения	248
Контрольные вопросы	250
Список литературы к главе 3	251
Заключение	255



ВВЕДЕНИЕ

Современное информационное пространство — это все в большей степени интернет-пространство, обладающее целым рядом принципиально новых свойств. В частности, благодаря поисковым системам, кардинально изменяется ситуация с доступностью информации и ее ранжированием по различным критериям.

Принципиально важной является также возрастающая автоматизация оценки эффективности информационных ресурсов с помощью различных рейтинговых систем, что в конечном итоге повышает объективность, эффективность и значимость таких оценок.

Имеет также место все более четкое выделение статической и динамической составляющих показателей эффективности присутствия в информационном пространстве.

При этом **статическая составляющая** в основном связана с наличием, количеством и качеством соответствующих информационных ресурсов, представленностью их в национальных и международных наукометрических и прочих базах данных, доступностью и заметностью, индексами цитирования, возможностью их нахождения при помощи поисковых систем. **Динамическая составляющая** определяется интенсивностью использования соответствующих информационных ресурсов, важным косвенным показателем чего являются индексы цитирования. Но важнейшим прямым показателем динамической составляющей являются показатели посещаемости и времени просмотра соот-

ветствующих информационных ресурсов, что в современных условиях для ресурсов, представленных в интернет-пространстве, может учитываться и анализироваться с высокой степенью точности.

Освоение необходимых теоретических основ для эффективного использования современных интернет-технологий в профессиональной деятельности и приобретение соответствующих практических навыков по созданию профессионально-ориентированных сайтов и их продвижение в интернет-пространстве уже не кажутся «эксклюзивной экзотикой» современности, а, наоборот, приобретают статус мощного и незаменимого инструментария в арсенале специалистов, профессионалов, деловых людей.

Поэтому все большую значимость набирает курс «Интернет-технологии», читаемый преподавателями кафедры компьютерной инженерии Донецкого национального технического университета (ДонНТУ). В 2015 году курс «Интернет-технологии» отпраздновал свое 15-летие!

Содержание дисциплины раскрывается в следующих темах:

1. Использование Интернета как качественно нового источника информации.
2. Разработка тематических электронных библиотек и списков ссылок на документы или информационные ресурсы.
3. Разработка гипертекстовых документов.
4. Работа с графической информацией в Интернете.
5. Программирование и отладка в среде браузера.
6. Методы и технологии оперативного профессионального общения в Интернете.
7. Применение веб-типографики для оформления текстов в Интернете.
8. Изучение и эффективное использование современного программного обеспечения для просмотра и верстки сайтов.
9. Другие вопросы и перспективы развития информационных возможностей Интернет.

В данном пособии рассматривается лишь часть вышеперечисленных тем, однако все серия книг «Интернет-технологии для студентов и преподавателей» полностью охватывает все содержание дисциплины «Интернет-технологии».

Основная задача курса «Интернет-технологии» включает в себя:

- изучение теоретических основ, методологии и методов современных информационных интернет-технологий;
- приобретение навыков использования интернет-технологий в профессиональной деятельности и прикладных исследованиях, разработка тематических сайтов.

Серия взаимосвязанных лабораторных работ направлена на формирование к концу курса **персонального тематического сайта**, основой содержания которого является систематизированные материалы по теме выпускной работы магистра, найденные и обработанные с использованием интернет-технологий в процессе выполнения лабораторных работ по курсу. Разработанные в рамках данного курса сайты размещаются на **портале магистров ДонНТУ (masters.donntu.org)**, который является одним из самых посещаемых научно-образовательных сайтов в Донецком регионе.

Данное учебное пособие «Интернет-технологии для студентов и преподавателей» основано на материалах курса «Интернет-технологии» для магистрантов, читаемого в Донецком национальном техническом университете с 2000 года. Эта книга — одна из книг серии «Интернет-технологии для студентов и преподавателей», в которую вошел материал о графической информации в Интернете, программном обеспечении для верстки и просмотра сайтов, для работы с графикой. Также читатели могут ознакомиться с основами веб-дизайна и веб-типографики. Книга состоит из трех глав:

Глава 1. Графическая информация в Интернете, в которой рассматривается растровая и векторная компьютерная графика, цветовые компьютерные модели, способы кодирования цвета и особенности хранения графической информации (включая алгоритмы сжатия и файлы для хранения графики).

Глава 2. Программное обеспечение для просмотра и верстки сайтов, в которой описывается порядок работы с рекомендуемым авторами набором программного обеспечения как для верстки и просмотра сайтов, так и для обработки графической информации, в том числе и создания анимированных изображений.

Глава 3. В помощь веб-разработчику, в которую вошли материалы книги «Веб-типографика. Искусство оформления текстов в Интер-

нете» (автор книги, С. В. Иваница), рассматривает основы веб-дизайна и веб-типографики, а также основные правила грамматики, необходимые при создании и подготовке к размещению в Сети качественных веб-текстов.

Каждая глава сопровождается списками контрольных вопросов в конце каждого тематического раздела и заканчивается собственным перечнем используемой литературы, в который также включены все ссылки на рассматриваемые в разделе интернет-сервисы, порталы и сайты.

Авторы данного пособия подготовили и изложили материал таким образом, чтобы он был в равной степени понятен и полезен как магистрантам и аспирантам, так и старшим школьникам и студентам. Авторы верят, что книга будет также интересна и полезна как молодым ученым, так и преподавателям всех квалификационных уровней.



ГРАФИЧЕСКАЯ ИНФОРМАЦИЯ В ИНТЕРНЕТЕ

1.1. Растровая и векторная компьютерная графика

Число пользователей Интернета постоянно растет, и многие уже не помнят или не застали первые сайты, появившиеся в Интернете, которые были, как правило, разработаны научными центрами для использования в военных целях (или в ВПК). Тогда основную ценность представляло их содержание — в основном текстовая информация. Теперь все по-другому — пользователей так много, что относительно небольшая их часть ищет научные или технические сведения о технологиях или фундаментальных исследованиях.

Большинство людей путешествуют по Сети для общения с друзьями, в поисках каких-либо товаров или просто рассматривают красивые картинки. Такой аудитории важно не столько содержание сайта, хотя и оно сохраняет свое значение, сколько его оформление — внешний вид с обилием графики. Хорошо оформленный сайт привлекает пользователя, даже если его содержание далеко от идеала, а вот для сайта, оформленного неряшливо, приобрести известность может оказаться непросто. Таким образом, в Интернете существенную роль играет **графическая информация**.

Графическая информация — сведения или данные, представленные в виде схем, эскизов, изображений, графиков, диаграмм, символов.

Наличие графической информации на сайте повышает его информативность, обеспечивает эстетическую привлекательность веб-страниц, а также, несомненно, является показателем профессионального уровня, в том числе и при разработке веб-дизайна.

Конечно же, эффектный веб-дизайн существенно повышает «заметность» сайта, однако принцип подачи информации в нем играет важную роль в завоевании популярности. Перегруженный текстом сайт рискует потерять огромное количество аудитории, а вот гармоничное сочетание графики и текста делают информацию в Интернете вполне востребованной. Информация и ее представление для аудитории теперь неотделимы друг от друга. Особенно это касается рекламы и корпоративных сайтов компаний, которые, как правило, ориентированы на продвижение товаров. Также социальные сети просто немыслимы без графической информации (достаточно вспомнить о фото («аватарках») участников, а также их многочисленные фотоальбомы). И редко какой новостной интернет-ресурс обходится только текстовыми новостями без обязательного тематического изображения или репортажного фото.

В этом разделе речь пойдет о **компьютерной графике**, без которой отсутствует возможность получения и обработки компьютерных изображений:

Компьютерная графика — это наука (один из разделов информатики), изучающая способы формирования и обработки графической информации с помощью компьютера.

Компьютерная графика появилась относительно недавно — уже в 60-ых годах XX века существовали полноценные графические системы. В настоящее время вся графическая информация может быть «компьютеризирована» с помощью **компьютерной графики** и **компьютерной анимации**: компьютерная графика включает все виды работ со статическими изображениями, а компьютерная анимация имеет дело с динамически изменяющимися изображениями.

По способам задания изображений компьютерную графику можно разделить на категории:

- **двухмерная графика:**
 - растровая;
 - векторная;
 - фрактальная.
- **трехмерная графика** — для изображений объемных объектов;
- **графика CGI¹** — неподвижные и движущиеся изображения, сгенерированные при помощи двумерной или трехмерной компьютерной графики.

В этой книге детально рассматриваются **растровая** и **векторная графика**, программы для обработки графических файлов, а также программы для работы с **компьютерной анимацией**.

1.1.1. Растровая графика

Растровое изображение — это прямоугольная (растровая) сетка пикселей на компьютерном мониторе, бумаге и других отображающих устройствах и материалах [1].

При использовании растровой графики важным элементом является размер полотна, тип цветопередачи и количество используемых цветов. **Пиксель** (*pixel*) — это мельчайшая единица цифрового изображения в растровой графике. Он представляет собой неделимый объект прямоугольной (обычно квадратной) формы, обладающий определенным цветом. Любое растровое компьютерное изображение состоит из пикселей, расположенных по строкам и столбцам в границах изображения (двумерный массив пикселей). Таким образом, если в устройстве отображения пиксели расположены достаточно плотно, то человеческому глазу трудно заметить структуру массива точек (собственно, пикселей) — поэтому такой массив воспринимается как единое изображение (рис. 1.1).

За последние годы число пикселей, отображаемых на компьютерных мониторах (разрешение экрана монитора), значительно увеличилось.

¹ CGI — аббревиатура английской фразы «Computer-Generated Imagery», что буквально переводится как «изображение, созданное компьютером».

Не так давно стандартным считалось отображение на экране 640×480 (всего 307 200) пикселей. Сейчас разрешение экрана современных мониторов начинается от 1024×768 (всего 786 432), 1280×1024 (всего 1 310 720) пикселей и постоянно увеличивается. Число пикселей, которое может отобразить компьютерная система, ограничивается объемом видеопамати и возможностями монитора и видеоадаптера (видеокарты).

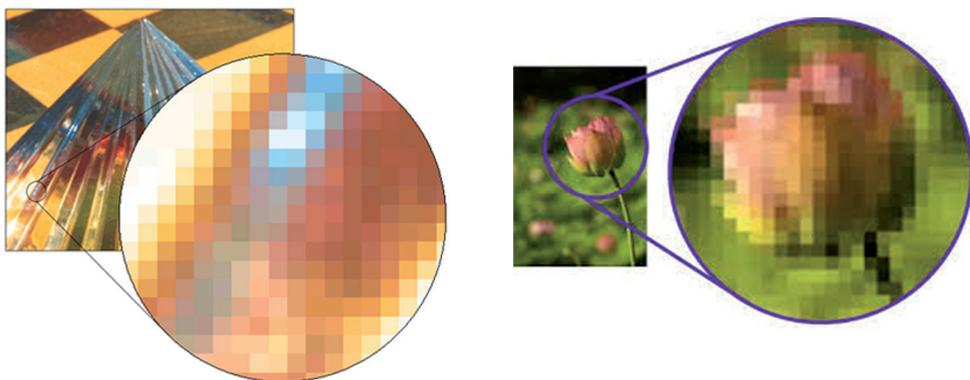


Рис. 1.1. Примеры растровых изображений с увеличенными фрагментами, в которых просматриваются группы пикселей (пикселизация изображений)

Для вывода растровых изображений используются также компьютерные принтеры. Изображение в принтере разбивается на множество пикселей, но с большей плотностью, чем на компьютерных мониторах (при печати на офисном принтере на квадратном дюйме размещается 300 или 600 пикселей). Изображение формируется в памяти принтера, а затем переносится на печатную страницу.

Таким образом, современные мониторы способны отображать, а принтеры — печатать растровые изображения в очень высоком качестве.

Основной недостаток растровых изображений — большой объем данных, необходимых для их отображения. Размер графического изображения в байтах составляет:

$$(\text{ширина} \times \text{высота} \times \text{число_битов_на_пиксель}) / 8.$$

Таким образом, для отображения или хранения изображения размером 600×800 пикселей, с 24 битами на пиксел, необходимо 1 440 000 байт (1,44 Мбайт) памяти или дискового пространства. Поэтому для уменьшения пространства, занимаемого файлом растрового изображения

на диске, обычно применяется сжатие, позволяющее хранить большее число изображений в дисковом пространстве компьютера.

Второй существенный недостаток — значительная потеря качества при изменении размеров растровых изображений. Максимальная детализация растрового изображения задается при его создании и не может быть изменена без дополнительных преобразований в пиксельной матрице. В результате таких преобразований, при уменьшении размеров изображения часть информации утрачивается: например, при укрупнении изображения возникают **эффекты блочности**. Рис. 1.1 иллюстрирует проблему, возникающую при увеличении размеров растрового изображения путем дублирования пикселей. Однако существуют методы сглаживания, способные в значительной степени уменьшать пикселизацию и позволяющие улучшать внешний вид масштабированных изображений.

К **достоинствам растровой графики** можно отнести высокую скорость обработки больших изображений (эффективно при их отображении в маломощных или встроенных системах), а также возможность воспроизведения рисунков любой степени сложности — как схематических, так и фотографического качества, — в отличие, например, от векторной графики, где невозможно точно передать эффект перехода от одного цвета к другому без потерь в размере файла.

Растровые изображения обычно хранятся в сжатом виде. В зависимости от типа сжатия становится возможным или невозможным восстановление изображения в точности таким, каким оно было до сжатия (сжатие без потерь или сжатие с потерями соответственно) [1]. Так же в графическом файле может храниться дополнительная информация (метаданные) в соответствии со стандартом EXIF [2]: об авторе файла, фотокамере и ее настройках, координатах съемки, количестве точек на дюйм при печати и др. Эта информация может использоваться как пользователем, так и различными устройствами, например, принтером.

11.2. Векторная графика

Векторное изображение (рис. 1.2) представляет собой набор объектов (примитивов) — прямых и ломаных линий, простых геометрических фигур (многоугольник, окружность), сплайнов [3] и кривых Безье [4]. Этим объектам присваиваются атрибуты — толщина линий и цвет заполнения. Объекты векторной графики являются графическими

изображениями математических функций. В свою очередь графический формат файла для представления векторного изображения содержит в себе последовательность команд рисования. Кроме того, в файле хранится набор координат, векторов и других чисел, характеризующих объекты рисунка. Таким образом, программа для обработки или отображения векторного рисунка открывает такие файлы и выполняет записанные в них команды с установкой хранимых в файле параметров.



Рис. 1.2. Пример векторного изображения с демонстрацией масштабирования без потери качества отображения

Главным достоинством векторного изображения является отсутствие искажений и потери качества при масштабировании (рис. 1.2, 1.3) и трансформировании, поскольку при данных операциях происходит пересчет математических функций с новыми параметрами, а не перерисовка всего изображения, как это происходит в растровой графике (создание новой пиксельной матрицы). Также следует отметить минимальный размер векторного файла (по сравнению с растровым файлом), поскольку он хранит только параметры и команды для рисования, а не само изображение.

Также к достоинству векторных форматов можно отнести аппаратную независимость (т. е. способность обрабатывать файл на компьютерном устройстве с произвольной конфигурацией и с любой операционной системой), поскольку все параметры векторного изображения обычно указываются в аппаратно-независимых единицах.

К **фундаментальному недостатку векторной графики** можно отнести тот факт, что далеко не каждый объект может быть легко изображен в векторном виде (см. рис. 1.3) — для достижения подобия оригинальному изображению может потребоваться очень большое количество объектов с высокой сложностью, что негативно скажется на количестве

памяти, занимаемой изображением, и на времени для его отрисовки на устройствах вывода [5].

Следующий недостаток связан с тем, что все современные компьютерные видеодисплеи способны отображать информацию только в растровом формате. Для отображения векторного формата на растровом используются преобразователи, аппаратные (встроенные в видеокарту) или программные. При этом, если перевод векторной графики в растр достаточно прост, то обратного пути, как правило, нет — отсутствие эффективного алгоритма преобразования растра в вектор (трассировка растра, см. рис. 1.4) не обеспечивает высокое качество векторного рисунка (при том, что трассировка может требовать значительных вычислительных мощностей и времени). Однако следует отметить, что существует узкий класс устройств, ориентированных исключительно на отображение векторных данных. К ним относятся графопостроители, а также некоторые типы лазерных проекторов [6].

Векторные изображения используются при компьютерном черчении и создании технической иллюстрации, деловой графики, шрифтов, векторной анимации.

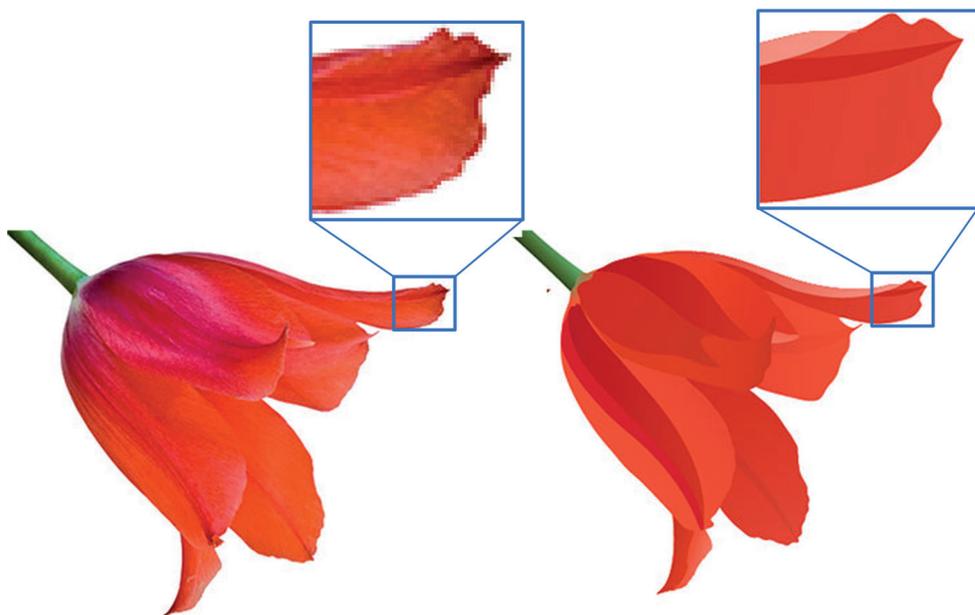


Рис. 1.3. Растровое (слева) и векторное (справа) изображения одного объекта с увеличенными выделенными фрагментами

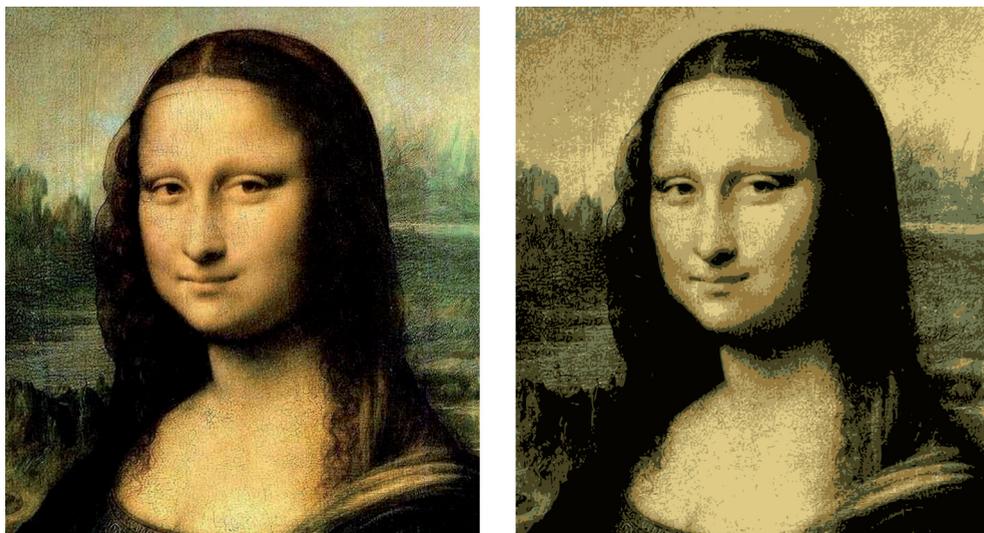


Рис. 1.4. Растровое изображение фрагмента портрета «Мона Лиза» (слева) и полученное из него векторное изображение (справа) путем трассировки с высокой степенью детализации

Контрольные вопросы

1. Дайте определение графической информации и компьютерной графики. Как связаны эти два понятия?
2. На какие категории можно разделить компьютерную графику исходя из способов задания изображений?
3. Что такое растровое изображение? Какая единица цифрового изображения используется в растровой графике?
4. Какими достоинствами и недостатками обладает растровая графика?
5. Дайте определение векторной графике. Из каких объектов (примитивов) состоит векторное изображение?
6. Перечислите основные достоинства и недостатки векторной графики.
7. В каких случаях целесообразно использовать растровые изображения, а в каких — векторные?

1.2. Цветовые модели. Работа с цветом

Описание цвета обязательно опирается на такие «цветовые» понятия, как насыщенность, плотность, тон и температура. Применительно к «компьютерному» цвету, полученному на экране монитора или выведенному на печать, описание цвета также должно учитывать специфику аппаратуры ввода/вывода для отображения цвета. В связи с этим, для воспроизведения цвета на компьютере, были разработаны различные цветовые модели. Таким образом,

Цветовая модель (англ. *color model*) — это формальная или физическая система, служащая для объяснения и предсказания спектральных свойств света.

Цветовая модель должна удовлетворять следующие требования:

- любой цвет должен быть определен стандартным способом, не зависящим от возможностей конкретного устройства;
- модель должна точно определять гамму задаваемых цветов, т. е. весь диапазон воспроизводимых цветов (цветовой охват).

Однако построение адекватной цветовой модели оказалось очень сложной задачей, которая до сих пор не получила исчерпывающего решения, поскольку, несмотря на значительные усилия разработчиков, универсальная теория, дающая полное объяснение феномену цвета в различных его проявлениях, еще не построена.

Существующие на сегодняшний день цветовые модели описывают цветовые оттенки с помощью смешивания нескольких основных цветов. Общий принцип построения цветовой модели основан на разложении основных цветов на оттенки (градации) по яркости, и каждой градации яркости присваивается цифровое значение (например, самой темной — 0, самой светлой — 255). Выбранный диапазон 0 ÷ 255 определен тем, что человек способен воспринимать около 256 оттенков одного цвета. По этому принципу любой цвет можно разложить на оттенки основных цветов и обозначить его набором цифр — цветовых координат.

Все цветовые модели можно разделить на два типа: субтрактивный и аддитивный. В основе субтрактивной и аддитивной цветовых моделей лежат соответствующие принципы субтрактивного и аддитивного и синтеза цвета. В результате такого синтеза получают одноименные цвета:

- **Субтрактивный цвет** (от англ. *subtract* — вычитать) получается при вычитании других цветов из общего луча отраженного света. В субтрактивной цветовой модели белый цвет появляется в результате отсутствия всех цветов, тогда как их присутствие дает черный цвет. При субтрактивном синтезе цвета наблюдается уменьшение яркости (рис. 1.5).
- **Аддитивный цвет** (от англ. *add* — добавлять, складывать) получается при соединении лучей света разных цветов (рис. 1.6). В аддитивной цветовой модели отсутствие всех цветов представляет собой черный цвет, а присутствие всех цветов — белый.

Классификации аддитивного и субтрактивного смешения цветов не существует, хотя часто полагают, что основные цвета, например, для аддитивного синтеза — это красное, зеленое и синее излучение, а для субтрактивного — голубой, пурпурный, желтый и черный цвета.

Далее рассмотрены цветовые модели, нашедшие применение в компьютерной графике как при выведении цветного изображения на печать, так и при его отображении на компьютерном мониторе [7].

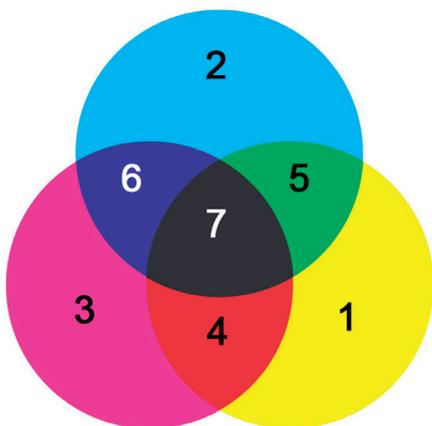


Рис. 1.5. Субтрактивный синтез цвета (1 – желтый, 2 – голубой, 3 – пурпурный, 4 – красный, 5 – зеленый, 6 – синий, 7 – черный)

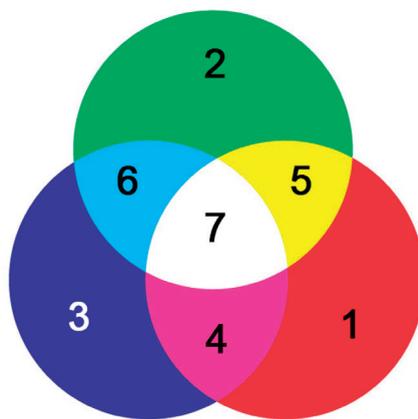


Рис. 1.6. Аддитивный синтез цвета (1 – красный, 2 – зеленый, 3 – синий, 4 – пурпурный, 5 – желтый, 6 – голубой, 7 – белый)

1.2.1. Модель RGB



Модель RGB — аддитивная цветовая модель, в которой любой цвет получается путем смешения трех основных цветов: красного (*Red*), зеленого (*Green*) и синего (*Blue*). Эти цвета также называют *цветовыми каналами*, или *аддитивными цветами* модели RGB.

Каждый из основных цветов может принимать 256 значений интенсивности в диапазоне от 0 до 255, поскольку каждый канал кодируется 8 битами (1 байтом): $2^8 = 256$. Таким образом, в модели RGB для кодирования цвета необходимо 24 бита (3 байта), следовательно, полное количество цветов, представляемых этой моделью равно

$$2^{24} = 256 \times 256 \times 256 = 16\,777\,216 \text{ (более 16,7 млн. цветов!)}$$

Но, несмотря на кажущийся значительный объем цветового охвата, в действительности модель порождает всего лишь небольшую часть видимого цветового спектра.

Исходя из особенностей аддитивного синтеза цвета, черный цвет в модели RGB получается, если интенсивность всех базовых цветов равна нулю, а белый — при их максимальной интенсивности (значения цветовых каналов равны 255).

Свойства модели RGB хорошо описывает так называемый *цветовой куб* (рис. 1.7) — фрагмент трехмерного пространства, координатами которого (x, y, z) являются красный (R), зеленый (G) и синий (B) цвета. Каждая точка внутри куба соответствует некоторому цвету и описывается тремя проекциями — цветовыми координатами: содержанием красного, зеленого и синего цвета. Сложение всех основных цветов максимальной яркости дает белый цвет — точку с координатами (255, 255, 255). Начальная точка куба с координатами (0, 0, 0) означает отсутствие вложенности основных цветов и соответствует черному цвету. Если цветовые координаты смешивать в равных пропорциях, то получится серый цвет различной насыщенности — точки, отвечающие серому цвету, лежат на диагонали куба.

Модель RGB описывает цвета, отображаемые мониторами и сканерами, поскольку именно в них цвет получается путем смешения световых пучков. Эта модель также используется для описания цветов на веб-страницах Интернета в специальном шестнадцатеричном виде.

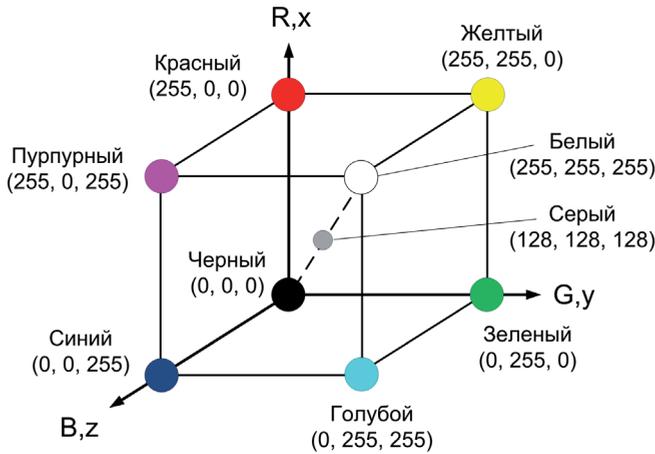
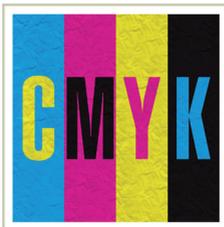


Рис. 1.7. Трехмерное представление цветовой модели RGB

Одним из серьезных недостатков модели RGB (как и любой аддитивной модели) является ее узкий цветовой охват. Также недостатком модели следует считать аппаратную зависимость, которая обусловлена тем, что цвет, воспроизводимый устройством, зависит от множества внешних факторов, часто не поддающихся учету. Например, мониторы (устройства, работающие по принципу сложения базовых цветов) одной модели и одной партии выпуска в процессе работы получают разный износ и эксплуатируются в различных условиях освещенности, поэтому будут отображать один и тот же цвет (который задан значениями интенсивностей базовых цветов RGB) по-разному.

1.2.2. Модель CMYK



Модель CMYK является *субтрактивной цветовой моделью*, которая использует три основных цвета: голубой (*Cyan*), пурпурный (*Magenta*) и желтый (*Yellow*). К этим цветам добавляется черный цвет (*black*), причины использования которого определяются следующими фактами:

1. Поскольку модель CMYK служит теоретической основой цифровой печати, то цветовые координаты рассматриваются как красители, которые наносятся на поверхность бумаги. При этом согласно принципу субтрактивного синтеза цвета, соединение

голубого, пурпурного и желтого цветов дает идеальный черный цвет. Однако на практике смешение голубого, пурпурного и желтого красителей не дает идеальный черный цвет, а цвет «грязно-коричневого» оттенка. Такой эффект происходит ввиду того, что существующие красители по своим химическим свойствам далеки от идеала и часто содержат примеси.

2. Вторая причина добавления кроется в экономичности, поскольку для нанесения черного цвета на бумагу требуется значительный расход трех красителей, а черный пигмент (в качестве которого, как правило, используется сажа) существенно дешевле красителей остальных трех цветов.

Модель СМΥК используют в основном устройства печати изображений — принтеры и печатные машины, которые фактически используют краски, соответствующие основным цветам СМΥК. Интенсивность каждой координаты (красителя) модели СМΥК измеряется в процентах от 0 (отсутствие краски) до 100 (максимальная плотность краски).

К **недостаткам модели** можно отнести плохо предсказуемые результаты получения цвета и очень узкий цветовой охват. К основополагающему достоинству можно отнести тот факт, что вся технология современной печати построена на модели СМΥК.

1.2.3. Модель HSB



Модель HSB (аналогичная аббревиатура — HSV) возникла как попытка преодолеть аппаратную зависимость модели RGB. Модель ориентирована на работу с цветопередающей аппаратурой и опирается на интуитивные понятия тона, насыщенности и яркости. Поэтому в модели HSB (HSV) все цвета определяются тремя координатами: цветовым тоном (*Hue*), насыщенностью (*Saturation*) и яркостью (*Brightness*, или *Value*). Цветовой тон варьируется в пределах от 0 до 359°. Насыщенность варьируется в пределах 0 ÷ 100 %, или 0 ÷ 255 градаций.

Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому. Яркость также задается в процентах в диапазоне 0 ÷ 100 %, или 0 ÷ 255 градаций.

В цветовом пространстве модели HSB используется **цилиндрическая система координат** (рис. 1.8), в которой тон H (координата x) определяется полярным углом, насыщенность S (координата y) — радиус-вектором, а яркость B — z -координатой. Таким образом, цветовой тон (оттенок) изменяется при движении вдоль окружности цилиндра, насыщенность — вдоль радиуса, а яркость — вдоль высоты.

Но и у этой модели есть существенный недостаток: на практике количество различных глазом уровней насыщенности и оттенков уменьшается при приближении яркости (B) к нулю (на оттенках, близких к черному цвету). Также при близких к нулю значениях S и B появляются существенные ошибки округления при переводе из модели RGB в модель HSV и наоборот.

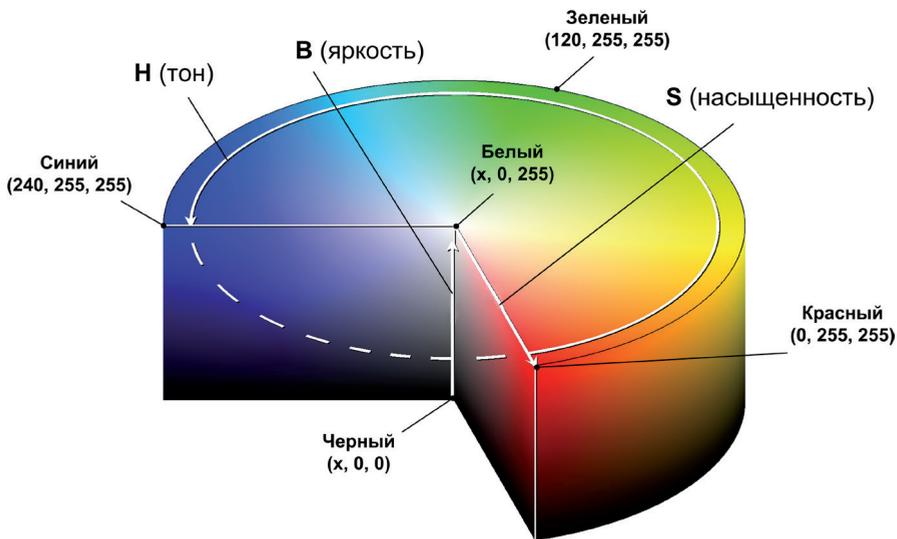


Рис. 1.8. Цилиндрическая система координат цветовой модели HSB

Поэтому чаще используется **коническая визуализация представления цветового пространства модели HSB** (рис. 1.9). Как и в цилиндре, оттенок изменяется по окружности конуса. Насыщенность цвета возрастает с отдалением от оси конуса, а яркость — с приближением к его основанию. Иногда вместо конуса используется шестиугольная правильная пирамида.

Достоинствами данной модели являются, прежде всего, аппаратная независимость, а также интуитивная ясность, удобство применения

и большой цветовой охват¹. **Существенным недостатком** модели HSB является необходимость преобразования в модель RGB для отображения на экране монитора или в модель CMYK для получения полиграфического оттиска. При переходе от аппаратно-независимой модели к аппаратно-зависимой (HSB → RGB и HSB → CMYK) неизбежна потеря цветопередачи.

Неустранимость этого недостатка привела к тому, что в настоящее время HSB используется только как вспомогательная модель.

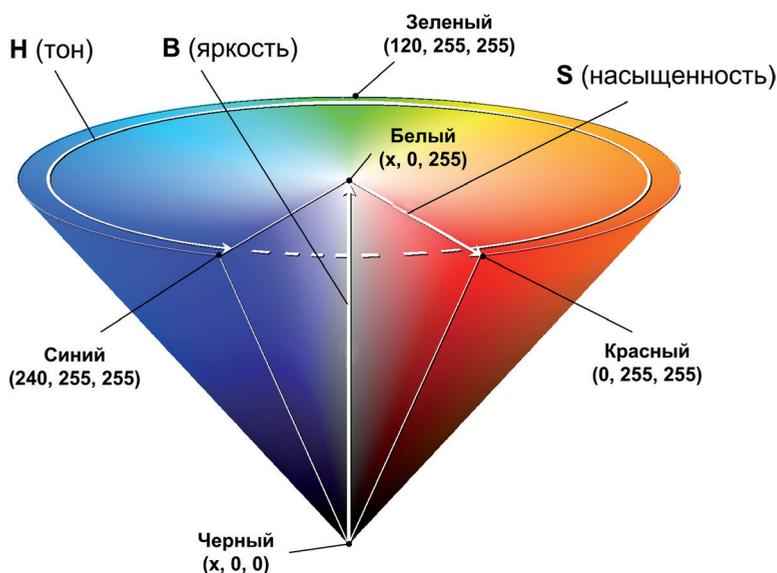


Рис. 1.9. Коническая система координат цветовой модели HSB

1.2.4. Модель HSL

Модель HSL, также известная как HLS или HSI, содержит в себе тон (*Hue*), насыщенность (*Saturation*) и светлоту (*Lightness*). Цветовой тон, или тип цвета варьируется от 0 до 359°. В большинстве приложений, каждое значение соответствует одному цвету: 0° — для красного, 45° — для оттенка оранжевого и 55° — для оттенка желтого (рис. 1.10).

¹ Цветовые модели с подобными свойствами называют перцепционными (интуитивными), в основу которых заложено раздельное определение яркости и цветности.

Насыщенность (или вариация цвета в зависимости от его светлоты) варьируется от 0 до 100 %. Светлота (светимость, сила света) также варьируется от 0 до 100 % (от черного цвета к белому).

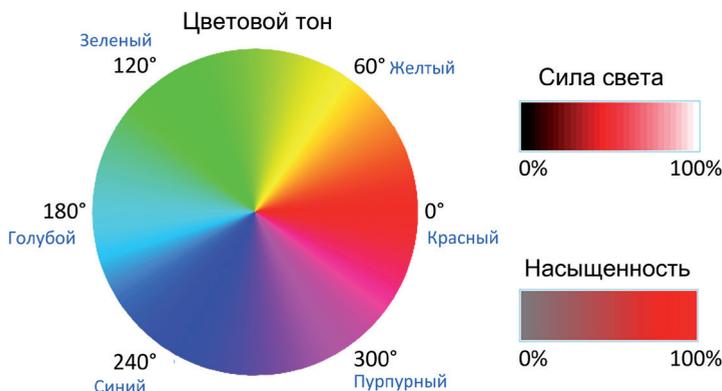


Рис. 1.10. Цветовая модель HSL

Модель HSL так же, как и HSB, является перцепционной моделью и обладает аппаратной независимостью. В цветовом пространстве модели HSL также используется цилиндрическая система координат. На примере цилиндрической системы координат на рис. 1.11 показана разница между моделями HSL HSB.

Для модели HSL также характерна коническая визуализация цветового пространства, причем, в отличие от модели HSB (HSV), коническое трехмерное пространство состоит из двух «склеенных» основаниями цветовых конусов (рис. 1.12).

Преимущества системы HSL включают в себя все достоинства модели HSB, к которым непосредственно добавляются преимущества HSL над HSB: симметричность модели; равноудаленность цветов в цветовой палитре; наличие правильной теории цвета (в центре модели находится нейтральный серый цвет).

1.2.5. Модель Lab

Модель Lab — аппаратно-независимая перцепционная цветовая модель, соответствующая особенностям восприятия цвета глазом человека. Любой цвет данной модели определяется яркостью (*Light*) и двумя хроматическими компонентами: параметром *a*, который изменяется

в диапазоне от зеленого до красного цвета, и параметром *b*, изменяющимся в диапазоне от синего до желтого цвета (рис. 1.13). Таким образом, модель Lab является трехканальной моделью, где канал яркости изменяется в диапазоне 0 ÷ 100 %, а каналы *a*- и *b*-компонентов — в диапазоне от -128 до 127.

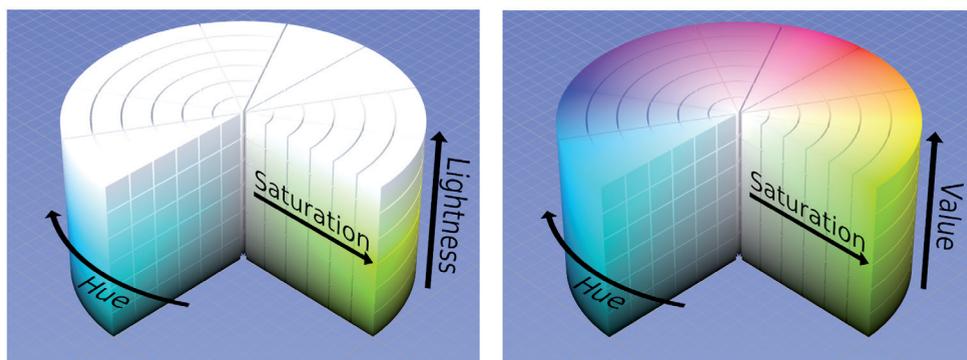


Рис. 1.11. Сравнение цилиндрической трехмерной визуализации цветковых моделей HSL (слева) и HSB (справа) [8]

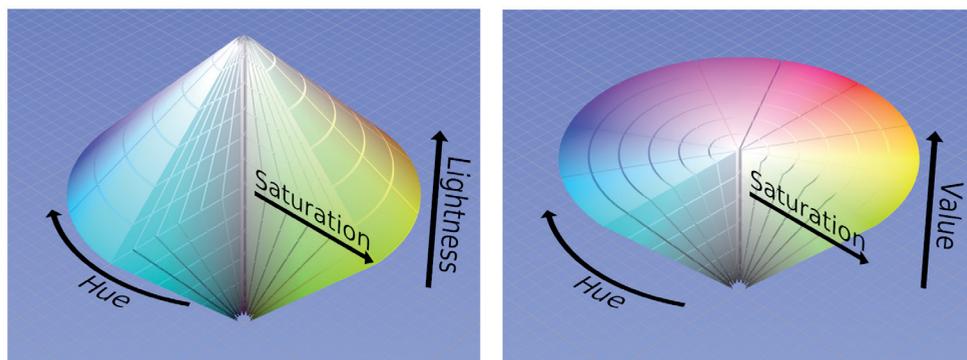


Рис. 1.12. Сравнение конической трехмерной визуализации цветковых моделей HSL (слева) и HSB (справа) [8]

Полное отделение яркости от цвета делает модель Lab удобной для регулирования контраста, резкости и других тоновых характеристик цветов. Цветовой охват модели Lab чрезвычайно широк и включает в себя цветовые охваты всех других цветковых моделей (рис. 1.14). Поэтому модель Lab лишена всех недостатков, связанных с потерей цветопередачи, присущих описанным ранее цветковым моделям, и позволяет осуществлять взаимное преобразование в любую цветковую модель без потерь в параметрах цвета.

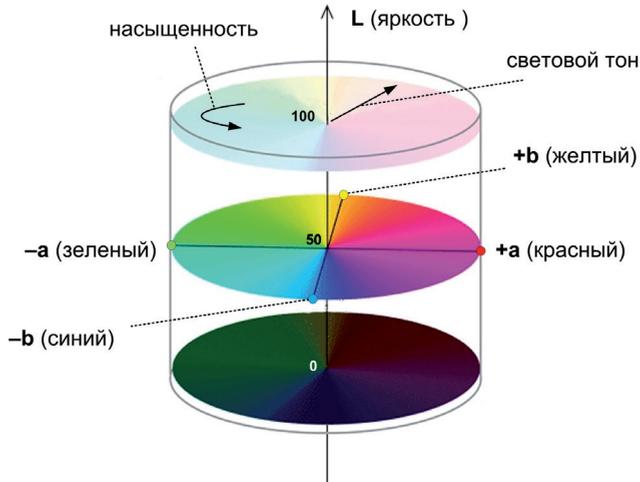
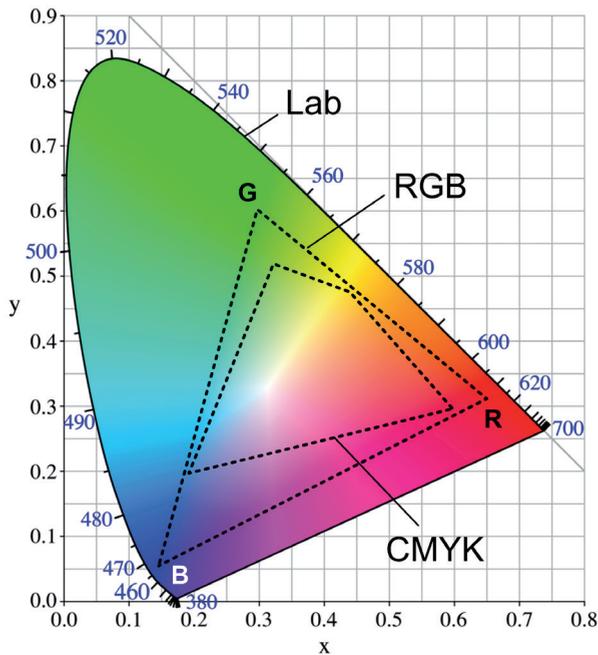


Рис. 1.13. Цветовая модель Lab

Рис. 1.14. Сравнение цветных охватов различных моделей на стандартной цветовой диаграмме CIE¹ XYZ с длинами волн цветов

¹ CIE (фр. Commission Internationale de l'Éclairage) — Международная комиссия по освещению.

Контрольные вопросы

1. Что такое цветовая модель?
2. Каким требованиям должна удовлетворять цветовая модель?
3. Что такое аддитивный и субтрактивный синтез цвета?
4. Какие отличительные особенности имеют аддитивные и субтрактивные цветовые модели?
5. Перечислите основные характеристики модели RGB.
6. Какими параметрами задаются цвета в модели RGB? Какое количество цветов можно представить с помощью этой модели?
7. Каковы причины добавления черного цвета к основным цветам модели CMYK?
8. В каких устройствах и по какой причине используется цветовая модель CMYK?
9. Что такое перцепционная цветовая модель? Какие перцепционные цветовые модели вам известны?
10. Чем отличается цветовая модель HSB от модели RGB?
11. Каковы основные особенности и преимущества модели HSL по отношению к HSB?
12. Какими параметрами задаются цвета в модели Lab? В каких единицах они измеряются?
13. В чем заключается основное преимущество Lab по отношению к остальным цветовым моделям?

1.3. Хранение графической информации

1.3.1. Глубина цвета

Ранее говорилось, что для вывода графической информации используются компьютерные мониторы. При этом важной характеристикой последнего является **разрешающая способность экрана**. Она измеряется как количество пикселей на единицу длины, **dpi** (*dots per inch* — «точка на дюйм»). Для экрана обычно это 72 или 96 dpi, (для сравнения — у лазерного принтера — в среднем 600 dpi). Чем больше dpi, тем меньше «зернистость» монитора и лучше качество изображения.

Не менее важным признаком изображения является количество цветов, обеспечиваемое видеокартой. Его можно менять программно (в пределах возможностей видеокарты) выбирая соответствующий **режим изображения**:

- **черно-белое** или битовое (0 – белый цвет, 1 – черный цвет);
- **16 цветов** (4 бита информации в пикселе, $2^4 = 16$ цветов);
- **256 цветов** (8 бит информации в пикселе, $2^8 = 256$ цветов);
- **Highcolor** (16 бит информации в пикселе, $2^{16} = 65\,536$ цветов);
- **Truecolor** (32 бита на пиксель, $2^{24} = 16\,777\,216$ цветов);
- **Deep Color** (сверх-Truecolor: 36 или 48 бит на пиксель).

Режим изображения Truecolor требует следующих уточнений: **32-битный Truecolor** в реальности является 24-битным с дополнительным 8-битным каналом, который либо не влияет на цвет (заполнен нулями), либо представляет собой канал прозрачности (альфа-канал) для каждого пикселя. Таким образом, 32-битный Truecolor обеспечивает $2^{24} = 16\,777\,216$ оттенков цветов и $2^8 = 256$ градаций прозрачности. Такой режим позволяет создавать на экране изображение, не уступающее по качеству цвета живой природе.

В 2006 году в аппаратный интерфейс была добавлена **поддержка режима Deep Color** для передачи цифрового видеосигнала HDMI. Данный режим позволяет кодировать больше цветов, чем любой другой режим. При этом типичные жидкокристаллические дисплеи не способны отображать пиксели с глубиной цвета более 24 бит (эта ситуация наверняка изменится в недалеком будущем), а 36- и 48-битная глубина цвета являет-

ся избыточной для человеческого глаза (т. е., иными словами, возможное количество цветов больше, чем способен различить глаз человека) [9].

Указанное выше количество бит (1, 4, 8, 16, 32, 36 и 48) для кодирования цвета в одном пикселе определяет **качество цветопередачи** или **глубину цвета** цифрового изображения. Глубина цвета выражается единицей **bpp** — *бит на пиксел* (англ. *bits per pixel*).

Глубина цвета — термин компьютерной графики, означающий количество бит (объем памяти), используемое для хранения и представления цвета при кодировании одного пикселя растровой графики или видеоизображения [10; 11].



Рис. 1.15. Изображение с различной глубиной цвета (слева направо): 4 bpp, 8 bpp и 24 bpp [10]

1.3.2. Методы сжатия графической информации

Графическая информация хранится в памяти компьютера в виде данных соответствующего типа (т. е. фактически в виде файлов с графическим типом данных). Поскольку характерной особенностью большинства типов данных является их избыточность, то, степень избыточности данных при их хранении зависит от самого типа данных. Например, для видеоданных степень избыточности в несколько раз больше чем для графических данных, а степень избыточности графических данных, в свою очередь, больше чем степень избыточности текстовых данных.

Имея достаточно большой объем данных, необходимых для хранения растрового изображения, а также значительную степень избыточности, целесообразно использование различных методов, направленных на преобразование графических данных с целью уменьшения их объема. Такие методы называют **методами сжатия**, ну а сам процесс сжатия (компрессии), прежде всего, основан на частичном или полном устранении избыточности, содержащейся в исходных данных. Методы или алгоритмы сжатия также предполагают возможность реализации процесса восстановления данных (декомпрессии), т. е. процесса обратного сжатия, распаковки — обратного перехода к получению исходных (распакованных) данных.

В целом, все методы сжатия данных делятся на два основных класса:

- Сжатие без потерь** — методы сжатия, при которых возможно полное восстановление исходных данных.
- Сжатие с потерями** — методы сжатия, которые позволяют восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных.

Сжатие без потерь в основном используется для сокращения текстовых документов и компьютерных программ, где требуется абсолютная точность хранимых данных. Сжатие с потерями обычно применяется для сокращения объема аудио- и видеоданных и цифровых фотографий в тех случаях, когда такое сокращение является приоритетным, а полное соответствие исходных и восстановленных данных не требуется.

Для хранения графической информации использование сжатия с потерями оправдано, прежде всего, тем, что зрение человека при анализе изображения оперирует контурами, общим переходом цветов и сравнительно нечувствительно к малым изменениям в изображении. Поэтому на восприятии «картинки» данная потеря не сказывается, в том числе и виду целого ряда других особенностей человеческого зрения (например, распознавание изображения по принципу «что не ухватил глаз, подсказал (дорисовал) мозг»).

Рассмотрим несколько базовых методов сжатия изображений.

Одним из простейших способов сжатия является **метод RLE** (англ. *Run Length Encoding* — кодирование длин серий), который является — **алгоритмом сжатия данных без потерь**, заменяющий повто-

ряющиеся символы (серии) на один символ и число его повторов. Действие метода RLE заключается в поиске одинаковых пикселей в одной строке. Выполнение алгоритма позволяет строку одинаковых символов, составляющих серию, заменить новой строкой, содержащей сам повторяющийся символ и количество его повторов.

Например, если 48-пиксельная строка черно-белого изображения условно состоит из следующей последовательности бит (**Ч** — черный цвет, **Б** — белый)

```
ccccccccccccBBBBBBBBccccccccccccccccccccccccBBBB
```

то, применив RLE, т. е. посчитав количество повторяющихся пикселей, можно получить следующее: 12 пикселей черного цвета, 8 — белого, затем 11 — черного, один — белого, 12 — черного, и, наконец, 4 — белого. Вся полученная последовательность записывается в более короткую строку

```
12ч8Б11ч1Б12ч4Б
```

Если исходная строка состоит из большого количества неповторяющихся цветов, то используются дополнительные преобразования, например деление на части [12], что, в конечном счете, все равно приводит к уменьшению объема полученных данных в результате применения метода.

Алгоритм RLE ориентирован на изображения с небольшим количеством цветов (деловая и научная графика).

☑ Следующий метод сжатия без потерь — **метод LZW**, или **алгоритм Лемпеля — Зива — Велча** (англ. *Lempel-Ziv-Welch*) разработан в 1978 году соответственно Авраамом Лемпелем (англ. *Abraham Lempel*), Яаковом Зивом (англ. *Jacob Ziv*) и Терри Велчем (англ. *Terry Welch*) и доработан в 1984 году. Метод сжимает данные путем поиска одинаковых последовательностей (так называемые фразы) во всем файле. Выявленные последовательности сохраняются в таблице, им присваиваются более короткие маркеры (ключи). Так, если в изображении имеются наборы из розового, оранжевого и зеленого пикселей, повторяющиеся 30 раз, LZW выявляет это, присваивает данному набору отдельное число (например, «6») и затем сохраняет эти данные 30 раз в виде числа 6. В процессе работы алгоритм как бы создает индивидуальный словарь

наборов, поэтому подобные методы и называют **словарными методами сжатия** (такой подход основан на базовом, разработанном в 1952 году, **методе сжатия Хаффмана** (*Huffman*) [13, с. 392–398], который используется как составная часть многих схем сжатия).

Работу метода LZW можно наглядно продемонстрировать на примере сжатия строки текста [14]:

The ¹com⁴pression² and the ¹de²compression³ leave ⁴an⁵ ⁵im⁵pression⁵. ⁵hahahahaha!

Метод LZW, так же, как и RLE, лучше действует на участках однородных цветов, и действует гораздо лучше, чем RLE, при сжатии произвольных графических данных, но процесс кодирования и распаковки происходит медленнее.

☑ Метод **ССИТТ Group 4** — модификация алгоритма сжатия Хаффмана [13], которая в основном используется при сжатии черно-белых изображений (один бит кодирует один пиксел). Данный алгоритм сжатия без потерь был предложен четвертой группой по стандартизации Международного Консультационного Комитета по Телеграфии и Телефонии (англ. *Consultative Committee International Telegraph and Telephone*). Последовательности подряд идущих черных и белых точек в ССИТТ Group 4 заменяются числом, равным их количеству (аналогично методу RLE). А этот ряд, уже в свою очередь, сжимается по методу Хаффмана с фиксированной таблицей.

Алгоритм ССИТТ Group 4 незначительно отличается от предшественника ССИТТ Group 3. В Group 4 исправлены некоторые недочеты Group 3: например, устранение появляющихся в результате работы метода серии горизонтальных черных полос на белом фоне рисунка.

ССИТТ Group 3 и Group 4 первоначально были разработаны для сетей факсимильной связи (поэтому иногда их называют Fax 3, Fax 4). В настоящий момент они также используются в полиграфии, системах цифровой картографии и географических информационных системах. Алгоритм ССИТТ отличается простотой в реализации, является быстрым и легко реализуемым аппаратно.

☑ Метод сжатия **JBIG** — стандарт сжатия изображений, предложенный Группой Экспертов в Сжатии Бинарных Изображений (*Joint Bi-level Image Experts Group*), и стандартизованный Международной Организацией по стандартизации ISO/IEC.

JBIG позволяет управлять такими параметрами, как порядок разбиения изображения на битовые плоскости, ширина полос в изображении, уровни масштабирования. Последняя возможность позволяет легко ориентироваться в базе больших по размерам изображений, просматривая сначала их уменьшенные копии. Настраивая эти параметры, можно использовать эффект «огрубленного изображения» при получении изображения по сети или по любому другому каналу, пропускная способность которого мала по сравнению с возможностями процессора. Распаковываться изображение на экране будет постепенно, как бы медленно «проявляясь». При этом пользователь начинает анализировать картинку задолго до конца процесса разархивации [15].

Алгоритм построен на базе Q-кодировщика [16], патентом на который владеет IBM. Q-кодировщик, так же как и алгоритм Хаффмана, использует для чаще появляющихся символов короткие цепочки, а для реже появляющихся — длинные цепочки, где, в отличие от метода Хаффмана, используются и последовательности символов.

Обновленный стандарт JBIG2, предложенный той же группой, сжимает данные в 2–4 раза сильнее, чем JBIG — предыдущий стандарт сжатия однобитных черно-белых изображений [17]. JBIG2 был опубликован в 2000 году как международный стандарт ITU T.88 и в 2001 году как стандарт ISO/IEC 14492. Он применяется как для сжатия без потерь, так и для сжатия с потерями.

☑ **JPEG** — метод сжатия с потерями, разработан в 1991 году группой экспертов в области фотографии в рамках ISO (*Joint Photographic Expert Group*) специально для сжатия 24-битных изображений.

Алгоритм основан на **дискретном косинусном преобразовании** (англ. *Discrete Cosine Transform*, DCT) [18], применяемом к матрице изображения для получения некоторой новой матрицы коэффициентов [15]. Такой алгоритм в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета. Поэтому наибольшее распространение JPEG получил в цифровой фотографии и для хранения и передачи изображений с использованием сети Интернет.

Являясь методом сжатия с потерями, JPEG также имеет специальную реализацию сжатия без потерь — **lossless JPEG**. Lossless JPEG использует схему предсказания значения пиксела по трем ближайшим

соседям — верхнему, левому и верхнему левому пикселям, а для сжатия разницы между истинным и предсказанным значением пикселя используется **энтропийное кодирование** (т. е. кодирование с возможностью однозначного восстановления, основанное на усреднении вероятностей появления элементов в закодированной последовательности) [19].

Алгоритм сжатия с потерями ищет плавные цветовые переходы. Вместо действительных значений цвета пикселей, JPEG хранит скорость изменения от пиксела к пикселу. Лишнюю, с точки зрения эффективности, цветовую информацию он отбрасывает, усредняя некоторые значения. Перед началом работы алгоритма сжатия можно задать уровень компрессии: чем он выше, тем больше данных отбрасывается (файл становится «легче») и ниже качество сжатого изображения. Метод JPEG, как и другие методы сжатия с потерями, не подходит для сжатия изображений при многократной (многоступенчатой) обработке, так как искажения в изображения будут вноситься каждый раз при сохранении промежуточных результатов обработки. Такой метод не нужно использовать в тех случаях, когда недопустимы даже минимальные потери, например, при сжатии точных изображений, например, использующихся в медицине или космонавтике.

Основным недостатком сжатия по стандарту JPEG является частое появление на восстановленных изображениях «блоковых артефактов» (см. рис. 1.18): при высоких степенях сжатия изображение рассыпается на блоки размером 8×8 пикселей.

☑ **Фрактальный алгоритм** — алгоритм сжатия с потерями, основанный на обнаружении самоподобных участков в изображении. В результате работы фрактального алгоритма изображение представляется в более компактной форме с применением коэффициентов системы итерируемых функций ISF (от англ. *Iterated Function System*).

Сегодня наиболее известны два изображения, полученные с помощью ISF: «Треугольник Серпинского» и «Папоротник Барнсли» (рис. 1.16). При этом «Треугольник Серпинского» задается тремя, а «Папоротник Барнсли» четырьмя преобразованиями — в частности, подобия и различных трансформаций (аффинные преобразования). Каждое такое преобразование кодируется буквально считанными байтами, в то время как изображение, построенное с их помощью, может занимать и несколько мегабайт.

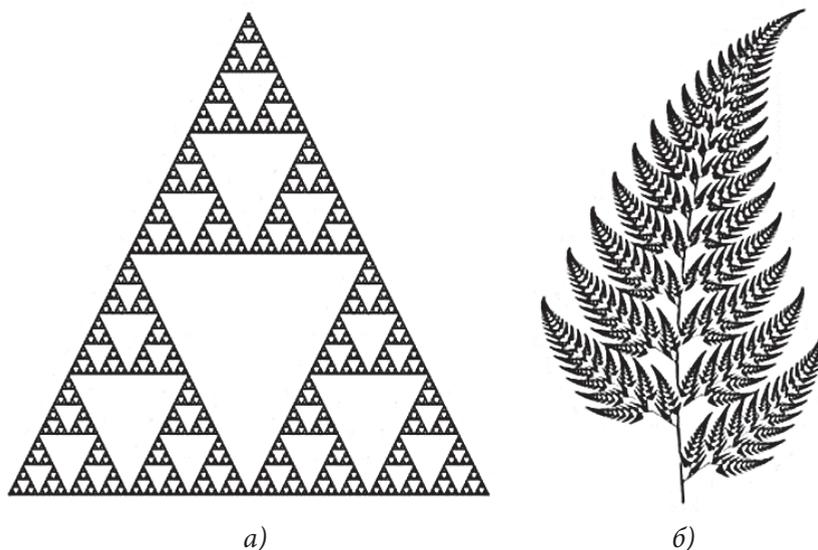


Рис. 1.16. Фракталы «Треугольник Серпинского» (а)
и «Папоротник Барнсли» (б)

Схема кодирования в данном методе основана на следующей последовательности действий (рис. 1.17). Вначале производится разбивка исходного изображения на квадраты, т. е. деление изображения на небольшие неперекрывающиеся квадратные области, называемые ранговыми блоками. Далее строится пул доменных блоков — всех возможных перекрывающихся блоков в четыре раза больших ранговых. Для каждого рангового блока по очереди «примеряются» доменные блоки и ищется такое преобразование, которое делает доменный блок наиболее похожим на текущий ранговый блок. Если «примерка» прошла успешно, т. е. доменный блок ставится в соответствие ранговому блоку, то в закодированное изображение сохраняются коэффициенты преобразования и координаты доменного блока.

Основной недостаток данного метода: алгоритм сжатия сводится к перебору всех доменных блоков и подбору для каждого соответствующего ему рангового блока, что довольно затратно по времени, особенно для больших изображений. В то же время алгоритм декомпрессии (распаковки сжатого изображения) довольно прост: необходимо провести несколько итераций аффинных преобразований, коэффициенты которых были получены на этапе сжатия (компрессии).

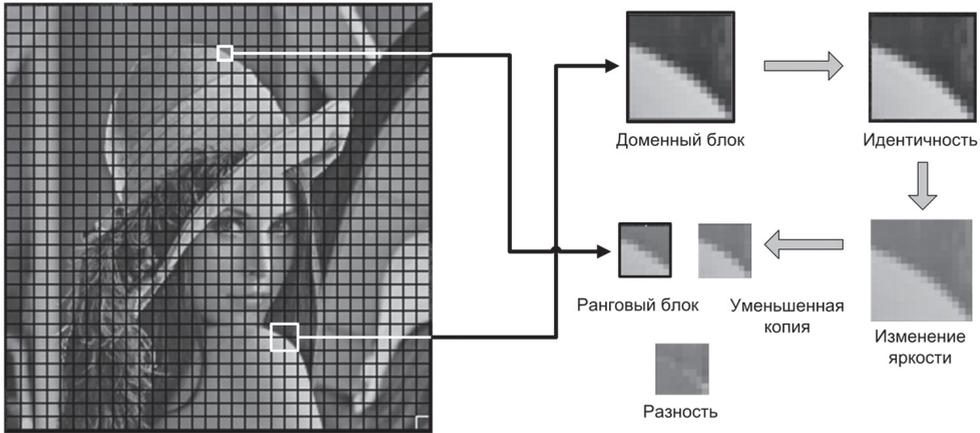


Рис. 1.17. Принцип работы фрактального алгоритма сжатия

☑ **JPEG2000** — относительно новый метод сжатия изображений, который был разработан для замены популярного ранее формата JPEG [20]. В отличие от JPEG, который при компрессии использует дискретное косинусное преобразование, в JPEG2000 в качестве основного механизма компрессии используется волновое преобразование (wavelet) — система фильтров, применяемых ко всему изображению.

В 1997 году были начаты работы над JPEG2000 как универсальной системы кодирования, которая снимала бы все ограничения, накладываемые JPEG, и могла эффективно работать со всеми типами изображений (черно-белые, в градациях серого, полноцветные) и независимо от их содержания (фотографии, мелкий текст, чертеж и пр.). В его разработке принимали участие наряду с международными стандартизирующими организациями такие гранды промышленности, как Agfa, Canon, Fujifilm, Hewlett-Packard, Kodak, LuraTech, Motorola, Ricoh, Sony и др. [21].

Основные возможности JPEG2000:

1. Достижение повышенной степени компрессии по сравнению с методом сжатия JPEG.
2. Обработка монохромных изображений (актуально для сжатия изображений с текстом).
3. Возможность сжатия без потерь (хотя JPEG2000 «по умолчанию» сжимает с потерями).

4. Вывод изображений с постепенным улучшением детализации — так называемое «прогрессивное сжатие», позволяющее по мере загрузки по Сети видеть сначала размытое, но затем все более четкое изображение.
5. Использование в изображении приоритетных областей с более высоким качеством, чем в остальной части изображения.
6. Декодирование в реальном режиме времени (без задержек), что особенно критично в мультимедийных приложениях, например, в интернет-трансляциях.

Сравнивая алгоритмы JPEG и JPEG2000 (рис. 1.18) можно говорить о том, что при высокой степени сжатия качество JPEG2000 существенно выше, чем у JPEG, что позволяет без особых потерь сжимать до 50 раз, а с некоторыми потерями (применимо к изображениям для Интернета) — до 100 и даже до 200 раз. Также при больших степенях компрессии в JPEG2000 изображение не приобретает характерную для JPEG блочную структуру.

К основному недостатку JPEG2000 можно отнести то, что он по сравнению с JPEG потребляет значительно больше системных ресурсов. Также сказывается то, что JPEG2000 пока мало распространен, вследствие чего поддержка файлов, использующих сжатие JPEG2000 (файлы с расширениями .j2k, .j2k, .jpf, .jpx, .jpm или .mj2) осуществляется не всеми современными браузерами (чтобы просматривать такие изображения в браузерах, нужно устанавливать дополнительный программный модуль).



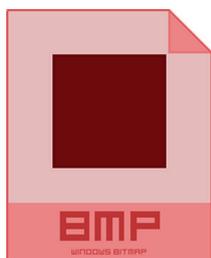
Рис. 1.18. Сравнение работы алгоритмов JPEG (слева) и JPEG2000 (справа) при 100-кратном сжатии изображения

1.3.3. Основные форматы файлов для хранения графической информации

Файлы для хранения графической информации (**графические файлы**) — это файлы, в которых хранятся любые типы графических данных, предназначенных для последующей визуализации. Изображение формируется и хранится в графическом файле в виде цифровых данных, которые могут изменяться в результате операций преобразования файла — эта возможность положена в основу работы всех без исключения графических редакторов. Формирование графического файла обусловлено определенным способом записи графической информации. Такой способ создания графического файла называют **графическим форматом**. Графические форматы делятся на:

- **Растровые форматы**, например: BMP, GIF, ICO, JPEG, JPEG 2000, PCX, PNG, PSD, RAW, TGA, TIFF, XPS.
- **Векторные форматы**, например: SVG, SVGZ, EPS, WMF, EMF, CDR, CMX, AI, XAR, SKP, STL, U3D.
- **Комплексные форматы** — форматы, содержащие в себе комбинацию нескольких графических форматов как векторных, так и растровых. Это форматы DjVu, PDF и CGM.

Далее детально рассматриваются некоторые популярные графические форматы.



BMP (от англ. *BitMap Picture*) — формат для хранения растровых изображений, разработанный компанией Microsoft и являющийся, соответственно, «родным» форматом для операционной системы Windows. Это значит, что формат BMP поддерживается всеми графическими редакторами, работающими под управлением Windows. Файлы этого формата могут иметь расширения **.bmp**, **.dib** и **.rle**. Глубина цвета в формате BMP может быть 1, 4, 8, 16, 24, 32, 48 bpp, а максимальные размеры изображения — 65535×65535 пикселей. При этом для глубины цвета меньше 16 bpp используется палитра (таблица цветов) с полноцветными цветовыми компонентами глубиной 24 bpp. В таких файлах цвет указывается индексом из этой палитры (индексированный цвет), а при цветовой глубине больше или равно 16 bpp — непосредственным значением цветов (как с ис-

пользованием шкалы серого цвета (*Grayscale*, 256 оттенков серого), так и RGB-цветов (около 16 млн цветов).

Фактически BMP — это стандартный, в основном несжатый битовый графический формат, в котором пиксели хранятся в виде относительно простого двумерного массива. BMP также способен хранить графику в аппаратно-независимом формате DIB (от англ. *Device Independent Bitmap* — аппаратно-независимый битовый массив). Такие файлы имеют расширение **.dib**.

В формате BMP изображения могут храниться «как есть» или же с применением некоторых алгоритмов сжатия. В частности, формат BMP поддерживает ранее описанное RLE-сжатие без потери качества. Такие файлы имеют, соответственно, расширение **.rle**, но поддерживаются не всеми графическими редакторами.

Файл BMP состоит из заголовка файла (растровый идентификатор, размер файла, ширина, высота, варианты цвета, и растровые данные начальной точки), заголовка изображения, палитры (которая может отсутствовать) и самого изображения [22].



GIF (от англ. *Graphics Interchange Format* — формат для обмена изображениями) — стандартный и довольно популярный растровый формат [23]. Был разработан в 1987 году фирмой CompuServe как формат для представления изображений во Всемирной паутине (спецификация **GIF87A**), и долгое время оставался одним из наиболее распространенных форматов в Интернете. В 1989 году формат был модифицирован (спецификация **GIF89A**), в который были добавлены поддержка прозрачности и анимации. Файлы формата GIF имеют соответствующее расширение **.gif**.

Для представления цветов в формате GIF используется индексированная 256-цветовая палитра: цвет изображения указывается индексом из этой палитры. Также один из 256 возможных цветов может быть объявлен «прозрачным». В этом случае в программах, отображающих изображение с «прозрачным» цветом, сквозь пиксели с этим цветом будет виден фон. Технология полупрозрачности (например, как в 32-битном *Truecolor* с альфа-каналом) в данном формате не поддерживается.

Изображение в GIF хранится построчно, однако формат поддерживает хранение данных «через строчку» (*Interlaced mode* или просто

Interlacing), благодаря чему, имея только часть файла, можно увидеть изображение целиком, но с меньшим разрешением (к слову, GIF поддерживает разрешение до 66536×65536 пикселей). Иными словами, **при чересстрочном отображении графики**, сначала загружаются (например, по Сети) четные строки изображения — при этом пользователь уже может просматривать условно «размытое» изображение. Затем загружаются нечетные строки изображения — происходит полная «сборка» исходного изображения.

В формате GIF изображение хранится в сжатом виде. GIF использует описанный ранее **метод сжатия без потерь LZW**. Таким образом, хорошо сжимаются изображения, имеющие относительно большие однотонные области с ровными краями. Также, с учетом специфики работы алгоритма LZW, хорошо сжимаются изображения, строки которых имеют повторяющиеся участки, в особенности те, в которых по горизонтали много пикселей одного цвета [24]. Это наглядно демонстрирует следующий пример:

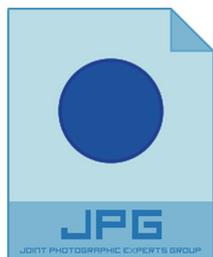
Если взять тестовое изображение (рис. 1.19 а) размером 1,5 килобайт и повернуть его на 90 градусов (т. е. графическая информация остается прежней и ни один пиксель не потерян), то полученное изображение (рис. 1.19 б) будет иметь размер всего 400 байт (почти в 4 раза меньше исходного!). Объясняется это тем, что компрессия в GIF (а компрессия происходит всегда) выполняется горизонтально и построчно. Поэтому, если белый цвет условно обозначить символом «w», а черный соответственно — символом «b», то первая строка исходного изображения будет записана в виде: **8b13w8b13w8b13w8b13w8b** — т. е. фактически «*восемь пикселей черного цвета, тринадцать белого, восемь черного...*», и так далее. В повернутом изображении первая строка будет записана в виде: **92b** — т. е. «*девятью два пикселя черного цвета*», — и все. Таким образом, во втором изображении нужно гораздо меньше места на запись информации.

Таким образом, учитывая особенности сжатия, формат можно использовать для хранения логотипов, надписей, чертежных схем, «мультишных» картинок, элементов для оформления сайтов, а также других изображений с небольшим количеством цветов. Для размещения фотографических и других полноцветных изображений формат GIF использовать не рекомендуется, так как в этом формате все цвета полноцветных изображений будут по умолчанию переведены в 256-цветовую палитру.



Рис. 1.19. Тестовое изображение в формате GIF размером 92×92 пикселя (а) и его точная копия, повернутая на 90° (б), имеют разные размеры: 1,5 килобайт и 400 байт соответственно [24]

Кроме того, файл GIF может поддерживать **анимационные изображения** — изображения, содержащие несколько растровых картинок (статичные кадры), которые при просмотре могут подгружать одну за другой с указанной в файле частотой и временем отображения каждого статичного кадра.



JPEG (от англ. *Joint Photographic Expert Group* — Объединенная группа экспертов по фотографии — название организации-разработчика формата) — один из популярных графических форматов, применяемый для хранения фотографических и подобных им изображений. Файлы, содержащие данные JPEG, обычно имеют расширения **.jpg** (самый популярный), **.jpe** или **.jpeg**.

Строго говоря, JPEG — это в меньшей степени формат, а в большей — алгоритм сжатия, основанный не на поиске одинаковых элементов, как в RLE и LZW, а на разнице между пикселями при поиске плавных цветовых переходов. Поэтому можно сделать следующий вывод: все достоинства и недостатки формата JPEG в точности повторяют описанные ранее достоинства и недостатки одноименного метода сжатия с потерями. Используя JPEG, можно получить файл в 10–500 раз меньше, чем при использовании BMP!

Таким образом напрашивается следующий вывод: формат JPEG идеален для полноцветных фотографических изображений в виду наличия в них многочисленных плавных цветовых переходов и плохо под-

ходит для логотипов или схем — в них больше полутоновых переходов и однотонных фрагментов.



TIFF (англ. Tagged Image File Format — «размеченный формат графического файла») — высококачественный формат хранения растровой графики. Формат был разработан и опубликован в 1986 году компанией Aldus Corporation (в 1994 году была поглощена корпорацией Adobe Systems) в сотрудничестве с Microsoft [25]. Файлы этого формата имеют расширение **.tif** или **.tiff**.

В формате TIFF существует возможность сохранения нескольких изображений, имеющих собственные наборы атрибутов и свойств (тегов) в одном файле. Это свойство делает формат TIFF похожим на GIF, однако возможность создания анимированных изображений у него отсутствует.

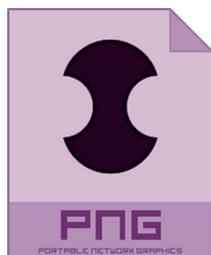
Формат способен хранить изображения, как в сжатом виде, так и без сжатия. Степени сжатия зависят от используемого алгоритма, а сжатие может осуществляться как с потерей, так и без потерь. Формат TIFF использует почти все описанные ранее алгоритмы сжатия: RLE, LZW, LZ77¹, ZIP², JBIG, JPEG, CCITT Group 3 и 4. Также в арсенале TIFF присутствует **поддержка альфа-канала**, который способен объединить переходную прозрачность (от 0 до 100%) с изображением. В отличие от формата GIF, где любой пиксель может быть либо полностью прозрачным, либо абсолютно непрозрачным (т. н. бинарная прозрачность), наличие альфа-канала в TIFF позволяет хранить 254 уровня частичной прозрачности между изображением и фоном.

Формат TIFF широко поддерживается графическими приложениями и используется для хранения изображений с большой палитрой цветов (поддерживает цветовые пространства RGB, CMYK, Lab, бинарные (черно-белые), полутоновые (Grayscale или оттенки серого) и с индексированной палитрой), например, цифровых фотографий или отсканированных документов (все современные сканеры могут сканировать изображения в TIFF). Одно из основных применений формата —

¹ LZ77 — алгоритм сжатия без потерь, наиболее известный вариант в семействе LZ*, которое включает в себя также алгоритм LZW.

² ZIP — популярный формат архивации файлов и сжатия данных без потерь [26].

полиграфия, где необходимо обеспечение максимального качества цифрового изображения. Поэтому TIFF — лучший выбор для использования растровых изображений в издательских системах, а также идеальный формат при импорте растровой графики в векторные программы.



PNG (англ. *Portable Network Graphics* — буквально «переносимая сетевая графика») — растровый формат хранения графической информации [27]. В формате PNG изображения хранятся в сжатом виде, при этом используется **метод сжатия без потерь Deflate** [28], который основан на комбинации работы алгоритмов сжатия LZ77 и Хаффмана. Данный формат был разработан в 1996 году как замена форматов GIF и TIFF и был одобрен консорциумом W3C [29] в качестве стандартного формата для Интернета. Файлы PNG, т. е. файлы с расширением **.png**, поддерживают все основные графические редакторы.

Формат обеспечивает индексацию до 256 цветов, поддержку вплоть до 48-разрядной глубины цвета (Deep Color) и, аналогично формату TIFF, реализацию канала прозрачности (альфа-канал). Как и в формате GIF, в PNG используется режим чересстрочной записи (*Interlacing*), причем не только строк, но и столбцов (**двумерная чересстрочная развертка**).

В файл формата PNG записывается информация о **гамма-коррекции**, которая характеризует зависимость яркости свечения экрана вашего монитора от напряжения на электродах кинескопа. Считанные из файла параметры гамма-коррекции позволяют ввести поправку яркости при отображении на экране. Это позволяет одному и тому же изображению, созданному в одной операционной системе, выглядеть одинаково, находясь на компьютерах с другими операционными системами и отображаясь в мониторах различных производителей. Таким образом, эта особенность помогает реализации основной идеи Всемирной паутины (WWW) — **одинакового отображения информации независимо от аппаратуры пользователя**.

Формат PNG хранит графическую информацию в сжатом виде, и обладает более высокой степенью сжатия для файлов с большим количеством цветов, чем формат GIF, но эта разница незначительная (5–25%), что недостаточно для абсолютного преобладания над GIF.

Формат PNG удобен для редактирования изображений, поскольку при хранении промежуточных стадий редактирования, потерь в качестве не происходит.

Однако, несмотря на то, что формат PNG был создан для замены GIF, в нем не реализована поддержка множественного изображения (анимации), поскольку PNG изначально был предназначен для хранения одного изображения в одном файле. Для анимированных изображений в 1999 году был разработан, близко связанный с PNG, расширенный формат MNG [30], который так и не стал общепринятым для разработчиков браузеров.



SVG (от англ. *Scalable Vector Graphics* — масштабируемая векторная графика) — векторный формат, файлы которого имеет расширение **.svg**. Фактически SVG — это язык разметки входящий в подмножество языка XML [31]. Созданный Консорциумом Всемирной паутины (W3C) [32], язык SVG служит для описания векторной и векторно-растровой графики.

SVG включает в себя три типа объектов (фигуры, изображения и текст) и не поддерживает трехмерную графику. Формат SVG разработан в 1999 году, а с 2011 года включен в рекомендации W3C [33].

SVG часто недооценивается начинающими веб-разработчиками, хотя имеет несколько важных преимуществ:

- Масштабирование.** В отличие от растровой графики, SVG — векторный формат, поэтому изображение не теряет в качестве при масштабировании.
- Доступность.** SVG — открытый стандарт и не является чьей-либо собственностью.
- Уменьшение HTTP-запросов.** При использовании SVG сокращается количество обращений к серверу (по сравнению с загрузкой изображений других форматов), и, соответственно, уменьшается время загрузки веб-страницы сайта.
- Управление.** Параметрами элементов SVG (например, заливками фона, прозрачностью или границами) можно управлять с помощью таблицы стилей CSS, либо напрямую с помощью атрибутов SVG элементов.
- Анимация.** При помощи языка сценариев JavaScript можно анимировать объекты SVG.

- *Редактирование.* Файлы SVG можно читать и редактировать при помощи обычных текстовых редакторов (например, в редакторе NotePad++), а также редактировать в векторных графических редакторах (например, в программе Adobe Illustrator).
- *Малый размер.* Объекты SVG имеют намного меньший объем, чем занимаемый растровыми изображениями.

Так, используя только текстовый редактор, можно создать изображение SVG, которое в дальнейшем можно использовать на страницах сайта и просматривать в браузере. Например, тестовый файл **111.svg** имеет следующее текстовое содержимое:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
3   "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
4 <svg version="1.1"
5   baseProfile="full"
6   xmlns="http://www.w3.org/2000/svg"
7   xmlns:xlink="http://www.w3.org/1999/xlink"
8   xmlns:ev="http://www.w3.org/2001/xml-events"
9   width="100%" height="100%">
10 <rect fill="red" x="5" y="5" width="80%" height="80%" />
11 <rect fill="silver" x="50" y="50" width="80%"
12   height="80%" rx="4em" />
13 <circle cx="200px" cy="200px" r="104px" fill="blue"
14   transform="translate(60,52)" />
15 </svg>
```

При этом в файле 111.svg описаны графические объекты — два прямоугольника «*rect*» (строки 10, 11) и окружность «*circle*» (строка 13), которые выглядят так, как это продемонстрировано на рис. 1.20.

При создании «сложных» изображений, содержащих большое количество объектов, код в файле SVG может быть объемным и занимать много места. Для таких случаев разработан формат **SVGZ** (файлы с расширением **.svgz**), в котором разметка SVG сжимается с помощью утилиты сжатия и восстановления (декомпрессии) файлов **gzip** (сокращение от *GNU Zip*), использующей алгоритм Deflate [28].

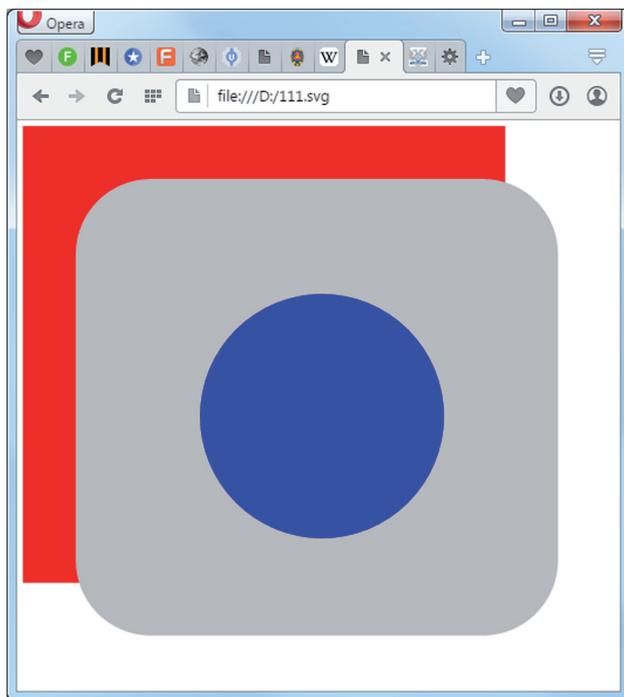
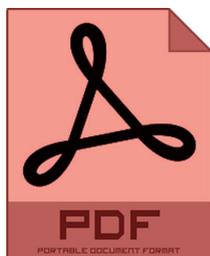


Рис. 1.20. Отображение файла формата SVG (111.svg) в браузере Opera



PDF (англ. *Portable Document Format*) — «яркий» представитель комплексных графических форматов — кроссплатформенный формат электронных документов, созданный фирмой Adobe Systems с использованием ряда возможностей языка PostScript¹. Это значит, что, во-первых, с PDF можно работать в различных операционных системах, а во-вторых, что в PDF внедряется векторная и растровая графика. Файлы этого формата имеют одноименное расширение **.pdf**.

Чаще всего файл PDF является комбинацией текста и графики, реже — текста с формами, скриптами, написанными на языке JavaScript, 3D-графикой и другими типами элементов. В первую очередь PDF

¹ PostScript — разработанный в начале 80-х гг. язык описания страниц, который в основном применяется в настольных издательских системах.

предназначен для представления в электронном виде полиграфической продукции. Распространение этого формата в полиграфическом направлении привело к тому, что сейчас большая часть современного профессионального печатного оборудования имеет аппаратную поддержку формата PDF. Такая поддержка позволяет выполнять печать документов в данном формате без использования какого-либо программного обеспечения. Также PDF используется для просмотра многостраничных документов.

Формат PDF позволяет автоматически внедрять необходимые шрифты, а также векторные и растровые изображения, формы и мультимедиа-вставки [34]. Поддерживает цветовые модели RGB, CMYK, Lab и режим Grayscale. В формате PDF также используется несколько типов сжатия растровой информации, а также реализован механизм электронных подписей для защиты и проверки подлинности документов.

Формат PDF поддерживается всеми современными браузерами, которые способны просматривать PDF-файлы в собственном, встроенном «просмотрщике» (рис. 1.21) с возможностями навигации, масштабирования, сохранения и печати документа.

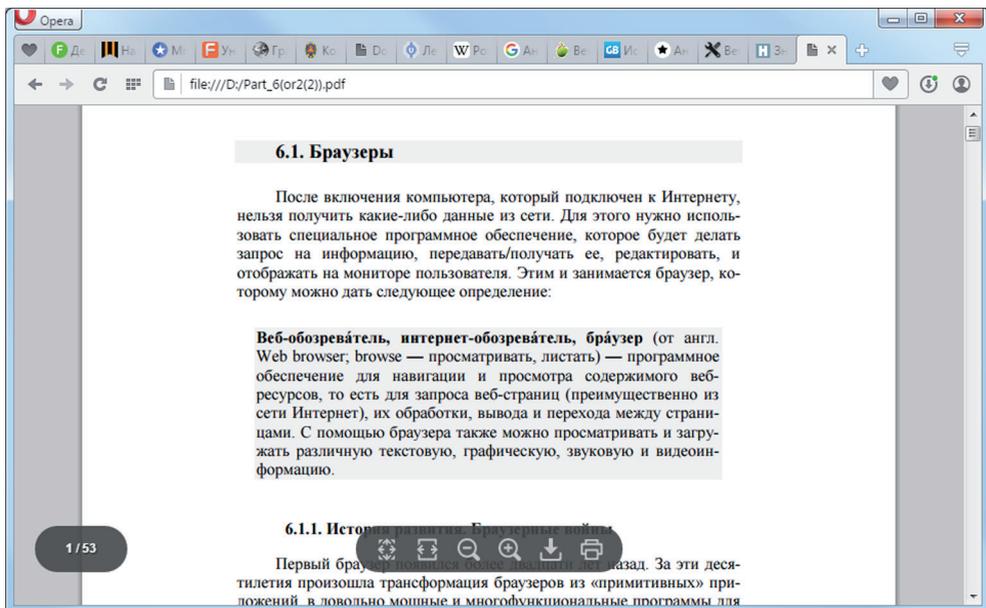


Рис. 1.21. Браузер Opera в режиме просмотра файла PDF

Контрольные вопросы

1. Перечислите режимы изображения. Как рассчитывается количество цветов для каждого режима по известному количеству бит информации на один пиксель?
2. Почему 32-битный режим Truecolor в реальности является 24-битным?
3. Для каких целей был разработан режим Deep Color? В чем заключается избыточность кодирования цвета в Deep Color?
4. Что такое глубина цвета, и в каких единицах она измеряется?
5. Какие методы сжатия графических данных вам известны?
6. В чем разница между алгоритмами с потерей информации и без потери информации?
7. В чем заключается принцип работы алгоритма сжатия RLE?
8. Как работает метод сжатия без потерь LZW?
9. В чем заключается принцип работы алгоритма сжатия CCITT?
10. За счет чего сжимает изображения алгоритм JPEG?
11. В чем заключается идея фрактального алгоритма компрессии?
12. В чем заключается идея сжатия JPEG 2000?
13. Чем отличаются форматы сжатия JPEG и JPEG 2000?
14. На какие типы делятся графические форматы?
15. Что такое комплексный графический формат?
16. В чем заключаются особенности растрового формата GIF?
17. Для каких изображений лучше всего использовать формат JPEG?
18. Почему формат JPEG стал наиболее популярным в среде растровых файлов?
19. Какая максимальная глубина цвета в изображениях форматов GIF и JPEG?

20. Каковы особенности формата TIFF? Для каких целей используется?
21. В чем особенности растрового формата PNG? Чем он отличается от форматов GIF и TIFF?
22. Как хранится изображение в файле векторного формата?
23. Перечислите преимущества векторного формата SVG.
24. Для каких случаев разработан формат SVGZ?
25. Чем обусловлено широкое распространение комплексного формата PDF?
26. Прокомментируйте сферу применения формата PDF.

Список литературы к главе 1

1. Растровая графика [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: https://ru.wikipedia.org/wiki/Растровая_графика — Загл. с экрана.
2. Exif.org. EXIF and related resources [Электронный ресурс]. — Режим доступа: <http://www.exif.org>. — Яз. англ.
3. Интерактивное введение в сплайны = An Interactive Introduction to Splines [Электронный ресурс]. — Режим доступа: <http://ibiblio.org/e-notes/splines/intro.htm>. — Загл. с экрана.
4. Bezier Curve — from Wolfram MathWorld [Электронный ресурс]. — Режим доступа: <http://mathworld.wolfram.com/beziercurve.html> — Загл. с экрана. — Яз. англ.
5. Векторная графика. Фундаментальные недостатки векторной графики [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: https://ru.wikipedia.org/wiki/Векторная_графика — Загл. с экрана.
6. Миано Дж. Форматы и алгоритмы сжатия изображений в действии / Дж. Миано. — М.: Издательство Триумф, 2003. — 336 с.: ил.

7. Иваница С. В. Веб-типографика. Искусство оформления текстов для Интернета. / С. В. Иваница. — Донецк: ООО «Технопарк ДонГТУ «УНИТЕХ, 2013. — 384 с.: ил.
8. File:Color solid comparison HSL HSV RGB cone sphere cube cyl-inder.png. From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/File:Color_solid_comparison_hsl_hsv_rgb_cone_sphere_cube_cylinder.png. — Загл. с экрана. — Яз. англ.
9. Nachman M. HDMI Upgraded To Support «Deep Color» / Mark Nachman [Электронный ресурс] // ExtremeTech — June 12, 2006. — Режим доступа: <http://www.extremetech.com/extreme/76410-hdmi-upgraded-to-support-deep-color>. — Загл. с экрана. — Яз. англ.
10. Глубина цвета [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: https://ru.wikipedia.org/wiki/Глубина_цвета. — Загл. с экрана.
11. Franzen R. Color Spaces (1998–2010) [Электронный ресурс] / R. Franzen. — Режим доступа: <http://r0k.us/graphics/colSpace.html>. — Загл. с экрана. — Яз. англ.
12. Кодирование длин серий [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: https://ru.wikipedia.org/wiki/Кодирование_длин_серий. — Загл. с экрана.
13. Левитин А. В. Алгоритмы: введение в разработку и анализ / А. В. Левитин. — М.: Вильямс, 2006. — 576 с.
14. Быстрянец М. Простейшие алгоритмы сжатия: RLE и LZ77. Хабрахабр [Электронный ресурс]. — Режим доступа: <http://habrahabr.ru/post/141827>. — Загл. с экрана.
15. Ватолин Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин — М.: ДИАЛОГ-МИФИ, 2002. — 384 с.
16. Pennebaker W. B. «Anoverview of the basic principles of the Q-coder adaptive binary arithmetic coder» / W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, R. V. Arps // IBM Journal of research and development, Vol. 32, No. 6, November 1988, pp. 771–726. — Яз. англ.

17. Progressive Bi-level Image Compression, Revision 4.1 // ISO/IEC JTC1/SC2/WG9, CD 11544. — 1991. — September 16. — Яз. англ.
18. Lakhani G. Distribution-Based Restoration of DCT Coefficients / Gopal Lakhani // IEEE Transactions On Circuits And Systems For Video Technology. — 2000. — Vol. 10, No. 5, August — P. 819–823. — Яз. англ.
19. David J. C. Information Theory, Inference, and Learning Algorithms / J. C. David МакКей. — Cambridge University Press, 2003. — 640 p. — Яз. англ.
20. Хашин С. И. Сравнение jpeg2000 и jpeg / С. И. Хашин [Электронный ресурс]. — Режим доступа: <http://math.ivanovo.ac.ru/dalgebra/Khashin/gr/jpeg2000/jpeg2000.html>. — Загл. с экрана.
21. Борисов М. Сжатие изображений: JPEG и JPEG2000 [Электронный ресурс] / М. Борисов // Publish/Дизайн. Верстка. Печать — Режим доступа: http://www.publish.ru/articles/200202_4045367. — Загл. с экрана.
22. Андрианов С. Формат ВМР-файла [Электронный ресурс] / С. Андрианов // Мир ПК. — 2001. — № 10. — Режим доступа: <http://www.osp.ru/pcworld/2001/10/162292>. — Загл. с экрана.
23. GIF™. Graphics Interchange Format. A standard defining a mechanism for the storage and transmission of raster-based graphics information [Электронный ресурс] // CompuServe Incorporated. — 1987. — June 15. — Режим доступа: <http://www.w3.org/Graphics/GIF/spec-gif87.txt>. — Загл. с экрана. — Яз. англ.
24. Лебедев А. § 8. Простой секрет ГИФа. .ру/Ководство (11 сент. 1998) [Электронный ресурс] / А. Лебедев. — Режим доступа: <http://www.artlebedev.ru/kovodstvo/sections/8/>. — Загл. с экрана.
25. TIFF. Revision 6.0 [Электронный ресурс] // Final — 1992. — June 3. — Режим доступа: <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>. — Загл. с экрана. — Яз. англ.
26. ZIP [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: <https://ru.wikipedia.org/wiki/ZIP>. — Загл. с экрана.

27. Roelofs G. A Basic Introduction to PNG Features [Электронный ресурс] / Greg Roelofs. — Режим доступа: <http://www.libpng.org/pub/png/pngintro.html>. — Загл. с экрана. — Яз. англ.
28. Михальчик Е. В. Описание формата сжатия данных Deflate. Все о сжатии данных, изображений и видео. / Е. В. Михальчик [Электронный ресурс]. — Режим доступа: http://compression.ru/download/articles/lz/mihalchik_deflate_decoding.html. — Загл. с экрана.
29. World Wide Web Consortium (W3C): официальный сайт [Электронный ресурс]. — Режим доступа: <http://www.w3.org>. — Загл. с экрана. — Яз. англ.
30. Multiple-image Network Graphics. PNG-like Image Format Supporting Multiple Images, Animation and Transparent JPEG [Электронный ресурс]. — Режим доступа: <http://www.libpng.org/pub/mng>. — Загл. с экрана. — Яз. англ.
31. Extensible Markup Language (XML). W3C SVG Working Group [Электронный ресурс]. — Режим доступа: <http://www.w3.org/XML/>. — Загл. с экрана. — Яз. англ.
32. Scalable Vector Graphics (SVG). W3C SVG Working Group — [Электронный ресурс]. — Режим доступа: <http://www.w3.org/Graphics/SVG/>. — Загл. с экрана. — Яз. англ.
33. Scalable Vector Graphics (SVG) 1.1 (Second Edition). W3C Recommendation [Электронный ресурс]. — 2011. — August 16. — Режим доступа: <http://www.w3.org/TR/SVG11/>. — Загл. с экрана. — Яз. англ.
34. Сырых Ю. А. Современный веб-дизайн. Рисуем сайт, который продает. / Юлия Сырых — М.: ООО «И. Д. Вильямс», 2008. — 304 с.: ил.



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОСМОТРА И ВЕРСТКИ САЙТОВ

2.1. Браузеры

После включения компьютера, который подключен к Интернету, нельзя получить какие-либо данные из сети. Для этого нужно использовать специальное программное обеспечение, которое будет делать запрос на получение и передачу информации, обрабатывать ее и в определенном виде отображать на мониторе пользователя. Этим и занимается браузер, которому можно дать следующее определение:

Веб-обозреватель, интернет-обозреватель, бра́узер (от англ. Web browser; browse — просматривать, листать) — программное обеспечение для навигации и просмотра содержимого веб-ресурсов, то есть для запроса веб-страниц (преимущественно из сети Интернет), их обработки, вывода и перехода между страницами. С помощью браузера также можно просматривать и загружать различную текстовую, графическую, звуковую и видеоинформацию.

2.1.1. История развития. Браузерные войны

Первый браузер появился более двадцати лет назад. За эти десятилетия произошла трансформация браузеров из «примитивных» приложений, в довольно мощные и многофункциональные программы для

работы в Интернете. Мы полагаем, что вам, как активным пользователям браузеров (а тем более слушателям курса «Интернет-технологии»), стоит знать, как это было!

Каждое творение человека имеет собственную историю, наполненную яркими событиями, фактами и даже случайными стечениями обстоятельств. История браузеров также насыщена яркими моментами и позволяет больше узнать как о развитии программного обеспечения (ПО) для веб-ресурсов, так и о развитии сети Интернет в целом. Ведь современные браузеры остаются обыкновенными компьютерными программами лишь с формальной стороны, однако фактически они выполняют функцию связующего звена между Интернетом и человеком и от того, как они справляются с возложенными на них задачами, зависит и наше восприятие виртуального мира.

Историческим началом развития браузеров считается следующее событие: автор концепции Web и создатель первого в мире сайта Тим Бернерс-Ли [1] выпустил текстовый браузер под названием «**WorldWideWeb**» (рис. 2.1). Его разработка пришлась на октябрь-декабрь 1990 года. Браузер работал в популярной в то время операционной системе NextStep, разрабатываемой компанией Next Стива Джобса [2]. Идея Тима Бернеса-Ли была подхвачена многими компаниями, которые начали разрабатывать свои собственные текстовые браузеры. Однако заметную известность браузеры приобрели с момента появления у них **графического интерфейса** [3], что также сказалось и на широком распространении Интернета [4].

Первым распространенным браузером с графическим интерфейсом стал **NCSA Mosaic** [6] (рис. 2.2), который впервые появился 22 апреля 1993 года. Работа по развитию браузера Mosaic была прекращена в 1997 году, однако авторы открыли исходные коды программы, что позволило разработчикам компаний Netscape и Microsoft создать свои браузеры на основе данных исходных кодов. Браузер NCSA Mosaic имел свои недостатки, но почти все они были устранены в браузере **Netscape Navigator** (рис. 2.3). Компания Netscape (некоторые сотрудники компании Netscape были из NCSA и участвовали в разработке Mosaic) выпустила Netscape Navigator под разные операционные системы (UNIX, Windows, Mac OS) и добилась заметного коммерческого успеха. Это побудило компанию Microsoft выпустить свой браузер **Internet Explorer (IE)** (рис. 2.4). Netscape Navigator был самым широко используемым браузером

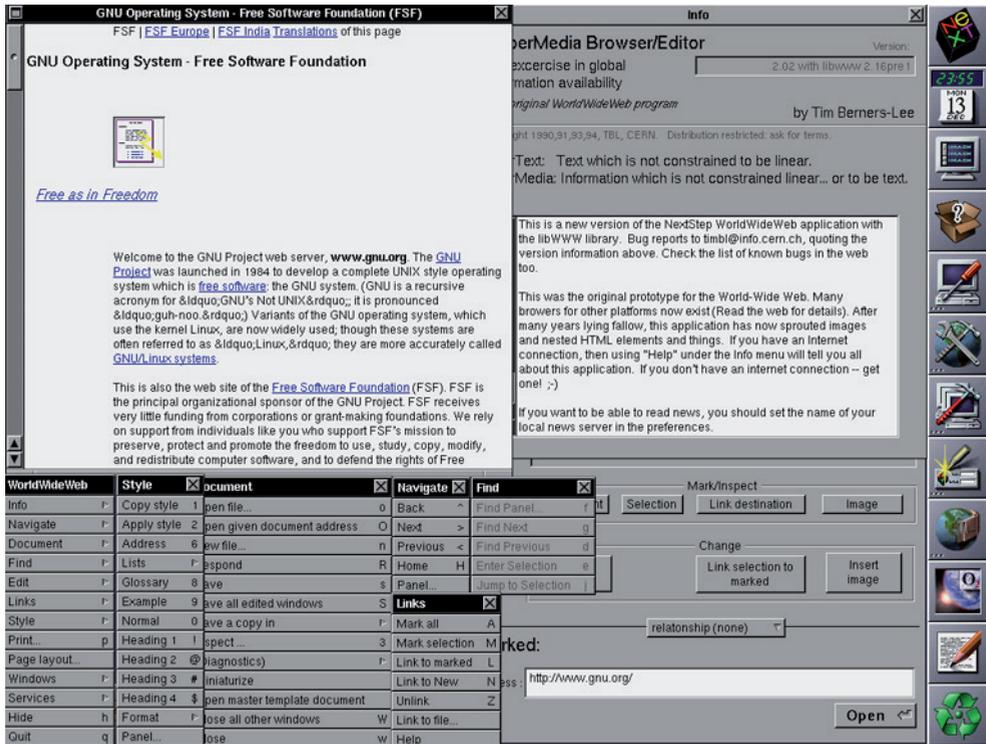


Рис. 2.1. Браузер WorldWideWeb с демонстрацией многих из его функций [5]

ром и Microsoft приобрела лицензию на Mosaic, чтобы создать Internet Explorer 1.0, выпущенный как часть пакета Windows 95 Plus! в августе 1995 года. Спустя три месяца для бесплатной загрузки был представлен браузер Internet Explorer 2.0. В отличие от Netscape Navigator (Netscape Navigator был бесплатным для домашних пользователей, но корпорациям приходилось покупать его за 99 долларов), он был бесплатно доступен всем пользователям Windows, даже коммерческим компаниям. Другие компании последовали примеру Microsoft и сделали свои браузеры бесплатными.

Какое-то время браузеры конкурировали честно, пока все падающая доля Internet Explorer не заставила Microsoft прибегнуть к иным методам завоевания рынка.

Речь идет о включении Internet Explorer в операционную систему Windows 95. Сразу после выпуска в ОС Windows 95 Internet Explorer не был встроен браузер, но он появился позднее, вместе с обновлением,

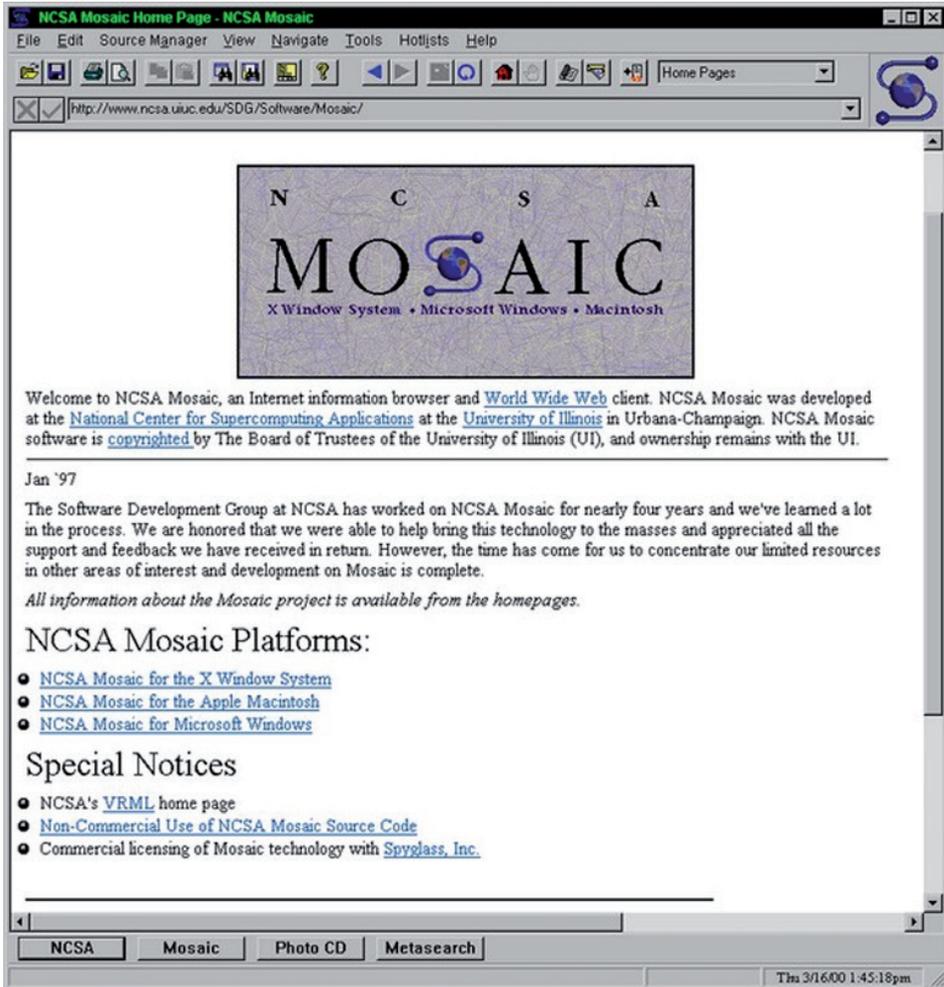


Рис. 2.2. Браузер NCSA Mosaic версии 3.0

получившим название Windows 95 OSR2. Это был Internet Explorer 3.0. Появление браузера, встроенного операционную систему значительно изменило расклад сил. Netscape Navigator стал стремительно терять популярность, поскольку пользователи Windows в основном пользовались встроенным браузером. Поэтому, вместе с ростом популярности Windows, росла популярность Internet Explorer.

Есть и еще один фактор популярности браузера. Когда IE только появился, он не поддерживал фреймы (т. е. возможность разделения окна браузера на отдельные области для загрузки в них информации



Рис. 2.3. Браузер Netscape Navigator версии 4.04

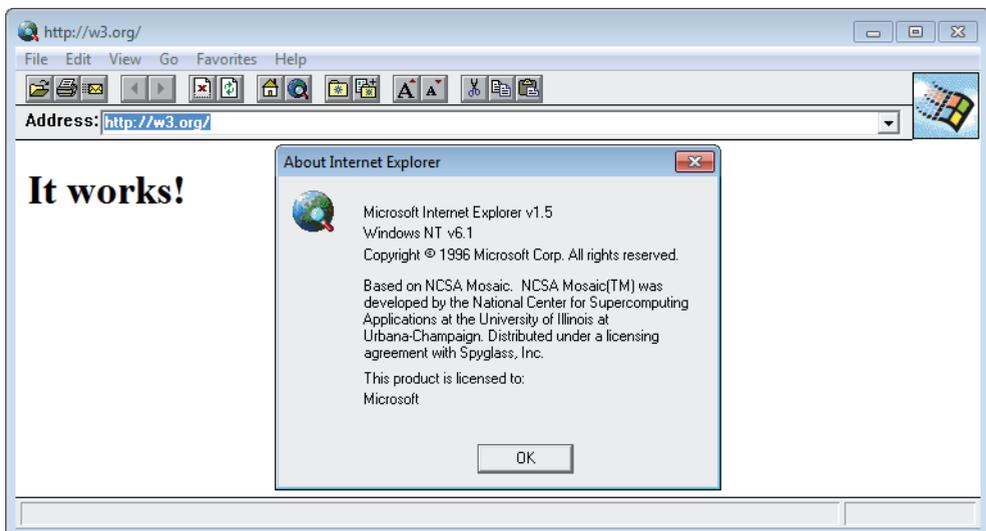


Рис. 2.4. Одна из первых версий браузера Internet Explorer

с различных источников), а потому веб-мастера, создающие сайты, настраивали их таким образом, что при получении User-Agent¹ Internet Explorer, сервер отдавал страницу без фреймов. Потом IE научился работе с фреймами, но веб-мастера не спешили перенастраивать сервера. IE пришлось маскироваться под Netscape Navigator, сообщая, что он совсем не IE, а Mozilla.

Mozilla — это внутреннее имя браузера Netscape Navigator, обозначающее Mosaic Killer (убийца Mosaic). Название показалось сотрудникам фирмы чрезмерно вызывающим, поэтому не стало официальным.

Internet Explorer, став монополистом рынка, фактически заставил всех веб-мастеров создавать страницы специально под него. При этом в Microsoft не слишком заботились о соблюдении веб-стандартов, что привело к ситуации: сайты создавались не по стандартам, а по тому, как решили реализовать ту или иную функцию в Microsoft.

Середину 1995 года можно считать началом **первой войны браузеров (1995–1999)**, закончившейся полным падением Netscape и триумфом Internet Explorer, занявшего более 95 % рынка (рис. 2.5). В таком состоянии браузер Internet Explorer дожил до своего апогея — Internet Explorer 6, вышедшего за несколько недель до выпуска Windows XP. Но тут началась **вторая браузерная война (2004 — настоящее время)**.

Теряющая рынок компания Netscape была куплена корпорацией AOL, которая продолжила традицию NCSA, выпустившей исходные коды Mosaic в свободный доступ, и поступила также с кодами Navigator. Это действие и оказалось той самой «бомбой замедленного действия», которая впоследствии разрушила монополию Microsoft.

В 2000 году был выпущен Netscape Navigator 6, представляющий собой браузер с совершенно новым движком² **Gecko**, который с самого момента своего создания был направлен на полную поддержку общепринятых веб-стандартов. Netscape Navigator 6 основывался на разработках

¹ **User-Agent** — это программа, которая по сети отправляет данные о себе. Термин обычно используется для приложений, осуществляющих доступ к сайтам (браузеры, поисковые роботы, мобильные приложения и пр.). При посещении сайта User-Agent обычно посылает веб-серверу информацию о себе в виде строки текста, являющейся частью HTTP-запроса.

² **Браузерный движок** (англ. layout engine) — программа, преобразующая содержимое веб-страниц (файлы HTML, XML, цифровые изображения и т. д.) и информацию

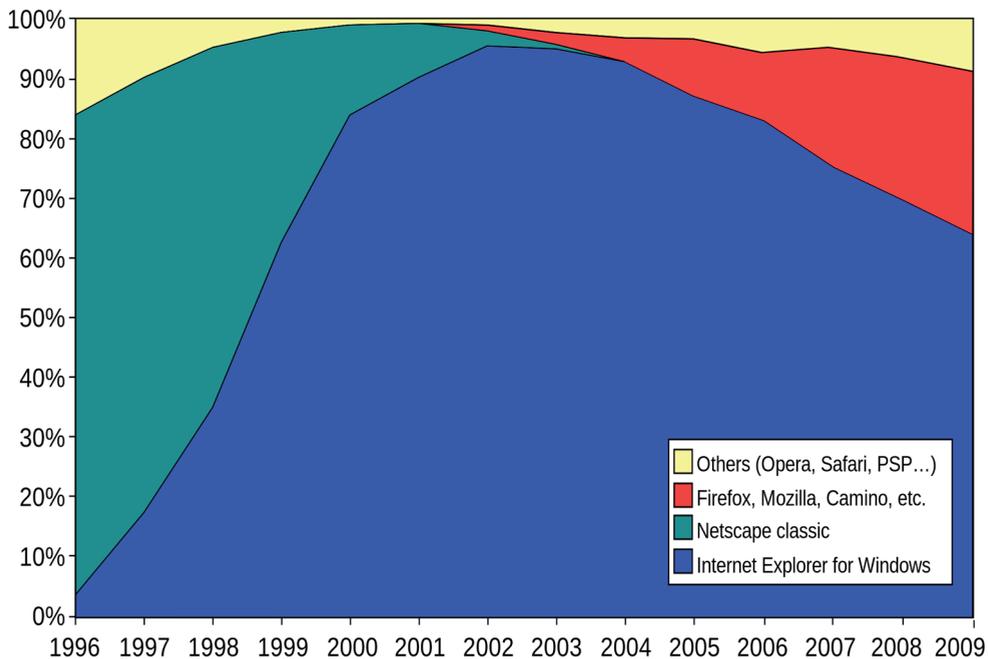


Рис. 2.5. Браузерные войны: конец первой и начало второй

проекта Mozilla Application Suite, многофункционального приложения, включающего в себя кроме собственно браузера, еще и почтовый клиент, календарь, IRC-клиент, HTML-редактор и средства веб-разработки.

Популярность Mozilla Suite оказалась выше, чем у основанной на ней Netscape Navigator, так что спустя несколько лет, AOL отказалась от развития браузера. Исходные коды были переданы проекту Mozilla Foundation.

Mozilla Foundation приняла решение о разделении Mozilla Suite на составляющие части. Получившийся браузер получил название «Phoenix», в честь птицы Феникс, сгорающей и возрождающейся из пепла. Таким образом, разработчики хотели показать преемственность Phoenix по отношению к Netscape Navigator. Вскоре он был переименован на «Firebird» («Жар-птица»), а затем в Firefox («Огненная лиса»). Переименования связаны с тем, что два предыдущих названия уже ис-

о форматировании (CSS, XSL и т. д.) в интерактивное изображение форматированного содержимого на экране. В статье [7] приведены таблицы сравнения движков различных производителей.

пользовались в качестве наименований программных продуктов у других производителей.

На фоне теряющего свои позиции Netscape Navigator, в 1996 году появился браузер **Opera** от норвежской компании с одноименным названием. Многие пользователи в России начали использовать этот браузер, потому что он был быстрее и легче Internet Explorer. Но популярность данного браузера долгое время сдерживала методика его распространения. Дело в том, что с самого своего появления Opera предлагалась своим пользователям по условно-бесплатной модели: в правом верхнем углу браузера находился рекламный блок, а если пользователь хотел его удалить, требовалось заплатить за это. Естественно, что при наличии бесплатных альтернатив мало кто желал использовать браузер с рекламой.

В 2005 году норвежские разработчики браузера поняли бесперспективность использования условно-бесплатной модели и сделали браузер полностью бесплатным.

В 2006 году на рынке браузеров для Windows появился еще один крупный игрок: компания Apple выпустила для ОС Windows свой браузер под названием **Safari**. Safari под Windows, так же как и его версия под Mac OS X работал на революционном движке **WebKit** [8]. В Apple значительно доработали движок и, в соответствии с лицензией, предоставили исходники всем желающим.

Необычность внешнего вида, современные технологии, находки в интерфейсе, бренд Apple — все это позволило браузеру обрести значительное количество поклонников. Вместе с тем, Safari для Windows не поддерживал расширения, что сразу отнимало у него большое количество очков.

Спустя два года, в 2008 году, компания Google выходит на рынок браузеров с продуктом **Chrome**. Это кроссплатформенный браузер с открытым исходным кодом, в котором использовался движок WebKit. По словам разработчиков Google Chrome, в нем впервые в истории браузеров была использована технология изолированных процессов. Кроме того, пользователю впервые предлагался режим «инкогнито», позволяющий не оставлять никаких «следов» в процессе работы с браузером.

И, наконец, в январе 2012 года на технологической конференции Yet another Conference был представлен браузер **Яндекс.Браузер**, создан-

ной компанией «Яндекс». Разработчики постарались использовать не только все лучшее от других браузеров, и добавить свои «фишки». Так, главной особенностью данного браузера стала его интеграция с сервисами самого Яндекса: почтой, переводчиком, «Яндекс.Диском», поиском «Яндекс» и пр.

В наши дни пользователям доступно большое количество самых разнообразных браузеров, но наиболее популярными являются **Google Chrome, Internet Explorer, Mozilla Firefox, Apple Safari и Opera** (см. рис. 2.6). Каждый из рассмотренных браузеров обладает неким базовым функционалом, который способен удовлетворить каждого «рядового» пользователя. В тоже время любой браузер имеет уникальные функции, присущие именно ему. Эти уникальные функции и являются причиной, по которой пользователи выбирают тот или иной браузер.

В настоящее время монополия Internet Explorer полностью исчерпала себя. Это пошло на пользу не только Вебу, который стал развиваться независимо от браузеров и строится по общепринятым стандартам, но и, в конечном счете, самому Internet Explorer. Возросшая конкуренция заставила программистов Microsoft потрудиться над новыми версиями браузера, которые практически не уступают своим конкурентам.

Эпоха Internet Explorer 6 закончилась в 2010 году, когда состоялись символические похороны этого браузера, на которые даже из Microsoft прислали прощальный венок [9].

Впереди ожидается интересная эпоха, в которой разработчики браузеров будут продолжать конкурировать между собой за внимание и благосклонность пользователей. При этом конкуренция будет вестись не за счет монопольного положения или других вне рыночных методов, а за счет улучшения существующих и внедрения новых функций и технологий.

2.1.2. Современные браузеры.

Достоинства и недостатки, рыночные доли

На сегодняшний день существует множество браузеров от различных производителей, но львиную долю на рынке занимает стремительно набравший популярность браузер Chrome, а также не успевший еще окончательно сдать свои позиции Internet Explorer. С небольшим отрывом следует браузер Firefox, дальше — Opera и набирающий популярность в рунете Яндекс.Браузер.

По данным StatCounter [10], на июль 2015 года, популярность браузеров в мире распределяется так, как это представлено на рис. 2.6, а, например, в Российской Федерации (собственно, в рунете) немного другая картина (рис. 2.7).

Но, прежде чем углубиться в обзор браузеров от ведущих компаний, поговорим подробнее о **браузерных движках**.

Браузерный движок является как бы операционной системой браузера, которая способна из принятых сетевых сигналов организовать их обработку и отобразить на экране монитора компьютера в привычном для пользователя виде. Основная цель каждого движка — создание разметки веб-страницы для удобного восприятия информации человеком. По сути, браузерный движок — это сердце браузера.

Браузерные движки создавались в разнообразных вариациях, но острая конкуренция в разгар браузерных войн подвергла их жесточайшему отбору, и к настоящему времени сохранились (в том числе и путем поглощения движков от более «слабых» компаний) и продолжают развиваться только три: **Trident** от компании Microsoft, **Gecko** от Mozilla Corporation, а также **WebKit** от группы разработчиков, в числе которых Apple Inc., Google Inc., Opera Software и др.

Все прочие современные движки — всего лишь вариации или производные от трех базовых движков-корифеев.

Сопоставим движки к рассматриваемым далее браузерам:

- Trident — для браузера Internet Explorer.
- Gecko — для браузера Mozilla Firefox.
- WebKit (и производный от него — Blink) — для браузеров Apple Safari, Google Chrome, Яндекс.Браузер и Opera.

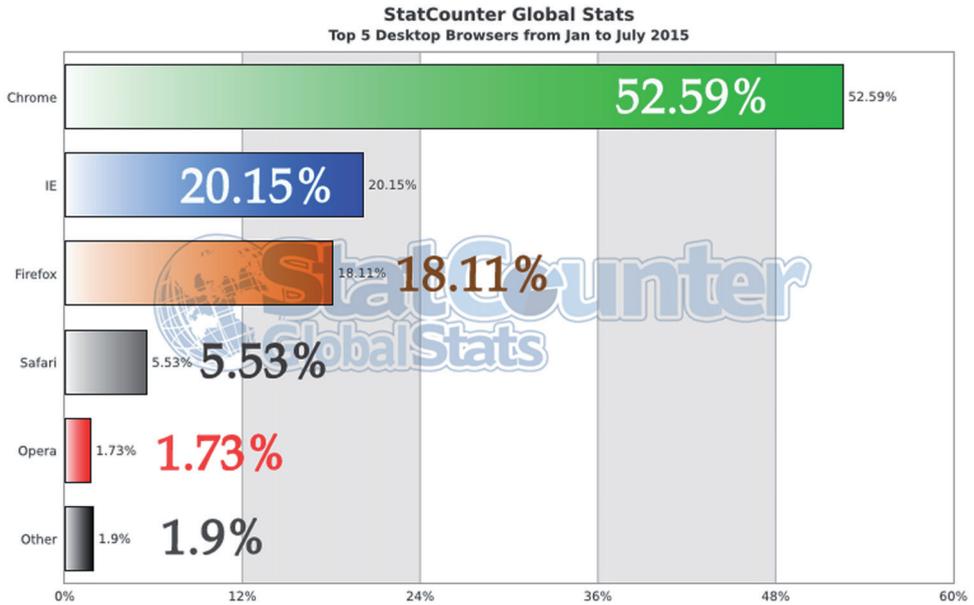


Рис. 2.6. Популярность браузеров в мире по данным StatCounter на июль 2015 года

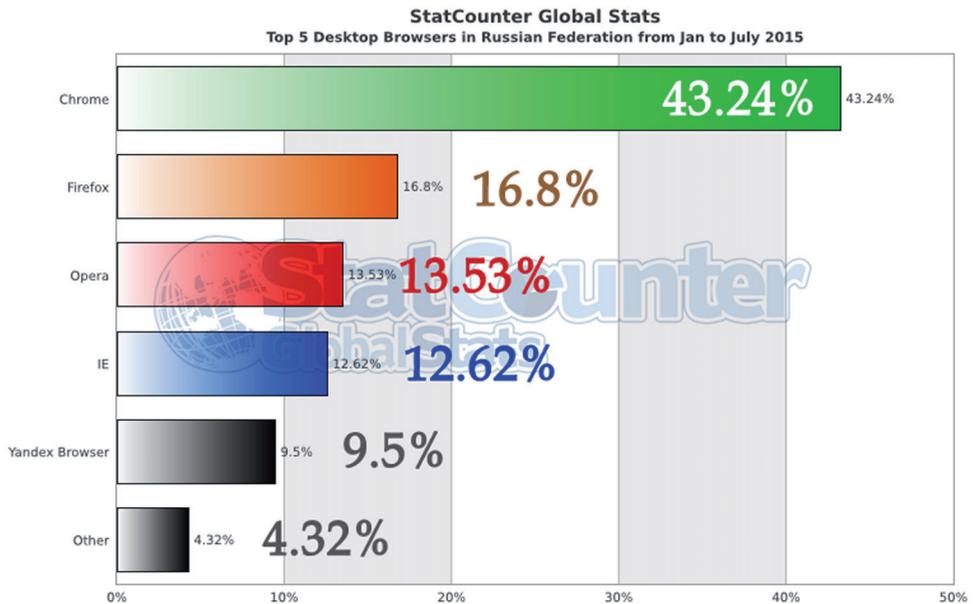


Рис. 2.7. Популярность браузеров в мире по данным StatCounter на июль 2015 года

Далее представлен краткий обзор пяти популярных браузеров (так называемой «пятерки») и примкнувшему к ним браузеру Яндекс.Браузер в порядке убывания их мировой популярности. Попробуем найти у них общие черты, выделим достоинства и недостатки, а в конце раздела подведем итоги и сориентируем читателей на выбор оптимального браузера.



Google Chrome — браузер, разрабатываемый американской корпорацией Google Inc. на основе свободного браузера Chromium [11] и использующий для отображения веб-страниц и движка **Blink** (до апреля 2013 года использовался упомянутый ранее движок WebKit). Первая публичная и стабильная версия этого браузера появилась 11 декабря 2008 года.

Браузер Chrome [12] по дизайну интерфейса очень прост (рис. 2.8), а количество его настроек сведено к минимуму — все это сделано для того, чтобы привлечь широкий круг пользователей, не обремененных тонкой настройкой программного обеспечения для работы в Сети.

Что касается работы браузера, то он отличается высокой стабильностью и наличием уникальных функций (подключаемых отдельно расширений), что обеспечивает стабильную и удобную работу в Интернете. Кроме того, браузер равно адаптирован под основные операционные системы: Windows, Linux и Mac OS X.

И наконец, этот браузер с открытым исходным кодом, который размещен на Google Code [13] и созданный на основе браузера Chromium. Там же есть очень много подробнейшей документации по внутренней архитектуре браузера, советуем почитать и посмотреть, тем более что в нем применяются новаторские подходы.

Перечислим основные достоинства Google Chrome:

1. Простота пользования, приятный интерфейс программы, а главное — скорость и стабильность работы браузера. Скорость — это одно из главнейших преимуществ Google Chrome, которое достигается постоянным усовершенствованием разработчиками движка браузера.

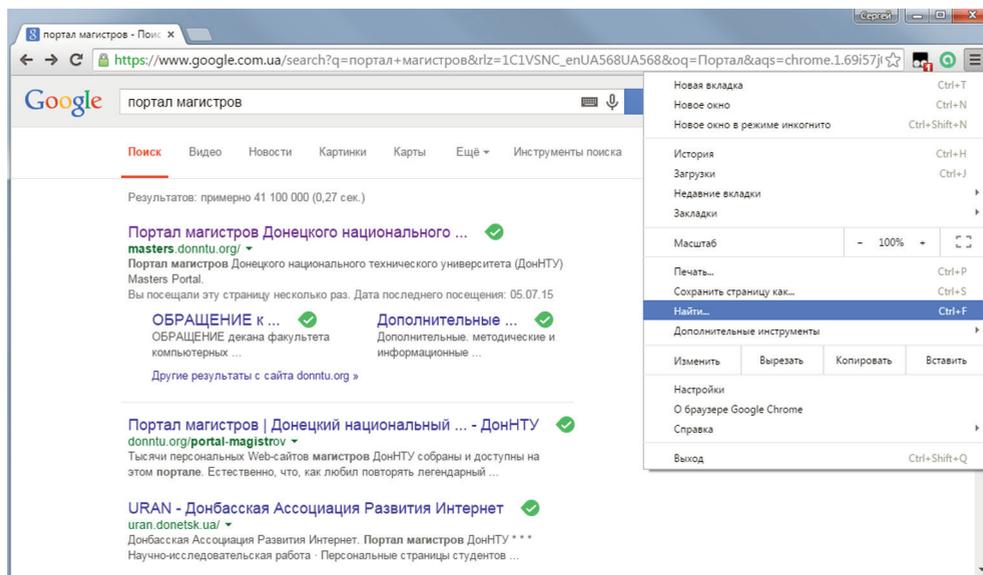


Рис. 2.8. Интерфейс браузера Chrome (версия 44.0.2403) с открытой выпадающей вкладкой «Настройка и управление»

2. Как уже упоминалось ранее, компания Google первая использовала в своем браузере технологию изолированных процессов. Это значит, что для каждой вкладки в Google Chrome выделяется отдельный процесс в компьютере. Такая вкладка способна «жить своей жизнью» и любой сбой в ее работе («зависание») не сказывается на работоспособности браузера в целом.
 3. Встроенный голосовой поиск в поисковике Google позволяет быстрее найти то, что нужно и не печатать нужные фразы. Но голосовой поиск можно реализовать при наличии микрофона на компьютере пользователя.
 4. Встроенный антивирус в браузере в виде двух черных списков позволяет отсеивать сайты с вредоносным программным обеспечением и фишинг-сайтов (сайты, целью которых является получение информации о пользователе).
- Недостатков у Google Chrome не много, но все же:
1. Неудобство работы с несколькими поисковиками. У браузера очень удобный поиск в Google, однако, у программы отсутствует возможность синхронизации с другими поисковыми системами (Яндекс, Rambler и т. д.).

2. Малый набор русскоязычных расширений. Конечно же, некоторые расширения легко находятся и на русском языке, но есть и расширения, большое количество которых на английском языке (например, расширения, связанные с сервером видео Youtube).
3. Значительная нагрузка на оперативную память и приличное расходование энергии аккумулятора не позволяет ноутбукам находиться в сети более 2–3 часов.

Таким образом, авторитет компании Google, а также превосходство значительных достоинств над незначительными недостатками (которые уменьшаются с появлением новых версий) вывели Chrome в ранг самых популярных в мире браузеров. Так, по данным StatCounter [10], по состоянию на июль 2015 года, браузер находится на первом месте по популярности (рис. 2.6, 2.7).

Загрузка текущей версии браузера возможна по ссылке [14] на официальном сайте Google Chrome.



Internet Explorer (IE) — программа-браузер (рис. 2.9), разрабатываемая корпорацией Microsoft с 1995 года. Входит в комплект операционных систем семейства Windows вплоть до Windows 10. IE был разработан для просмотра разных типов веб-страниц и поддержки ряда функций операционной системы, включая обновления Windows. Основанный на браузерном движке **Trident**, Internet Explorer также поддерживает ряд установленных и развивающихся стандартов, таких, как HTML5, CSS3, SVG и др. На сегодняшний день (июль 2015-го) браузер Internet Explorer имеет версию 11, которая была выпущена 8 апреля 2014 года.

К сожалению, в результате монополии IE в 90-х годах компания Windows выбрала путь собственной реализации обработки и отображения гипертекста, в результате чего браузеры IE вплоть до 9-ой версии отображали веб-страницы «по-своему». В результате многие элементы HTML и CSS, стандартизированные консорциумом W3C, или не отображались вовсе (в отличие от браузеров других разработчиков) или отображались так, что сайт буквально «рассыпался» в окне браузера. Это вызвало, в конечном счете, открытую ненависть к Internet Explorer у веб-разработчиков, ведь приходилось создавать сайт со стилями для всех браузеров и отдельно для IE. Такая «популярность» привела к тому,

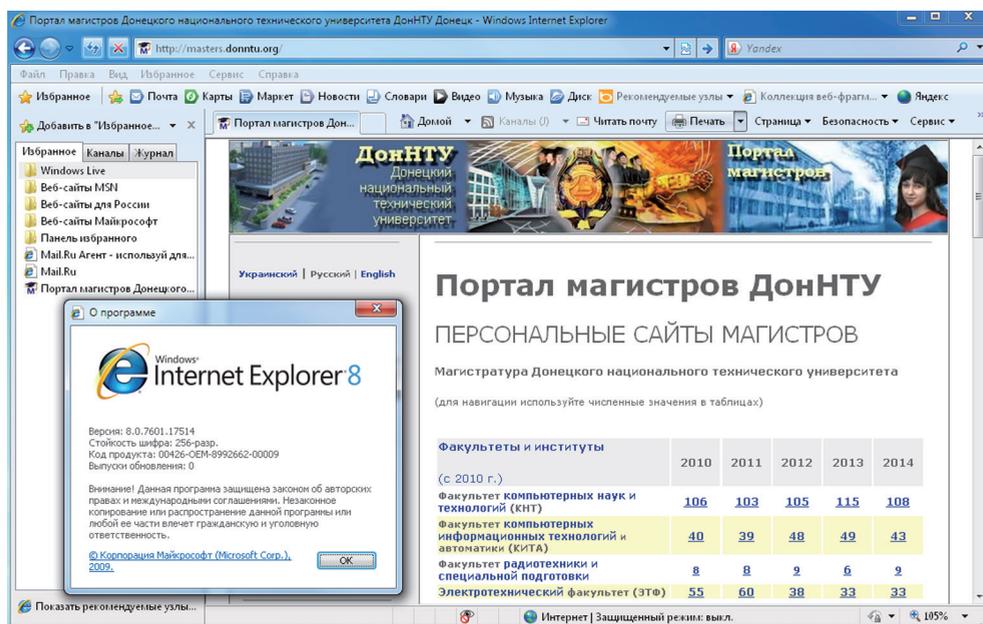


Рис. 2.9. Интерфейс браузера Internet Explorer (версия 8.0.7601)

что в большинстве случаев пользователи (в частности, рунета) стали использовать Internet Explorer только для того, чтобы в первый раз зайти в интернет на новом компьютере и скачать другие браузеры. Так IE снижал прозвище «браузер для скачивания браузеров».

Новые версии IE 9, IE 10 и IE 11 жестко «привязаны» к последним версиям операционной системы Windows и не могут быть установлены на ранние версии, например, на Windows XP (которая укомплектована IE 6) — до сих пор очень популярную систему. Очевидно, это такой маркетинговый ход от компании Microsoft, чтобы пользователи своевременно обзаводились последними выпусками Windows.

На фоне браузеров других производителей, Internet Explorer имеет одно-единственное достоинство — он устанавливается вместе с операционной системой, поэтому после ее установки на компьютер и подключения последнего к Сети, браузер по умолчанию более-менее готов к работе.

Перечислим основные недостатки браузера IE:

1. Довольно медленная обработка и отображение веб-страниц.
2. Неудобный доступ к настройкам и медленное их применение.
3. Отсутствие поддержки расширений.

4. В отличие от остальных браузеров работает только с операционными системами семейства Windows.

Многие пользователи линейки браузеров Internet Explorer все чаще задаются вопросом: почему такой гигант индустрии, как компания Microsoft, имеющая огромные финансовые и людские ресурсы, не может даже после многолетней критики, создать быстрый и удобный браузер? Ответ был получен от самих разработчиков. В декабре 2014 года глава по маркетингу Microsoft сообщил, что Internet Explorer (на тот момент браузеру исполнился почти 21 год) уходит в историю. Заменой ему будет **Microsoft Edge** (кодовое название — Project Spartan), который будет написан целиком с нуля [15].



Mozilla Firefox — браузер от американской компании Mozilla Corporation, работающий на движке **Gecko** (рис. 2.10). Это третий по популярности браузер в мире (см. рис. 2.6). В некоторых странах, например, в Германии, браузер имеет особенный успех. К середине 2015 года в России Firefox занимает второе место (см. рис. 2.7) по популярности среди браузеров для персональных компьютеров с долей 16,8 % пользователей. Первая версия браузера Firefox 1.0 появился на свет 9 ноября 2004 года.

Особенным преимуществом Mozilla Firefox среди браузеров-лидеров является надежность, так как у него самый низкий процент сбоев и наименьший коэффициент ошибок [16]. Firefox небезосновательно предупредит пользователя о возможной опасности при попытке посещения сайтов, содержащих вирусы, а также фишинговых¹ сайтов.

Среди браузеров-лидеров Firefox имеет самый «богатый» выбор дополнений в виде расширений, тем, подборок. Дополнения позволяют добавлять новые функции (например, изменять внешний вид, добавлять игры и развлечения, настраивать оповещения и обновления, управлять фотографиями, музыкой и видео, добавлять языковые инструменты, и т. д.) и фактически изменять способ работы браузера (индивидуализация браузера). Сейчас на официальном сайте дополнений [17] доступно

¹ **Фишинг** — вид мошенничества в Интернете, которое направлено на получение доступа к конфиденциальной информации пользователей, например к именам учетных записей и паролям

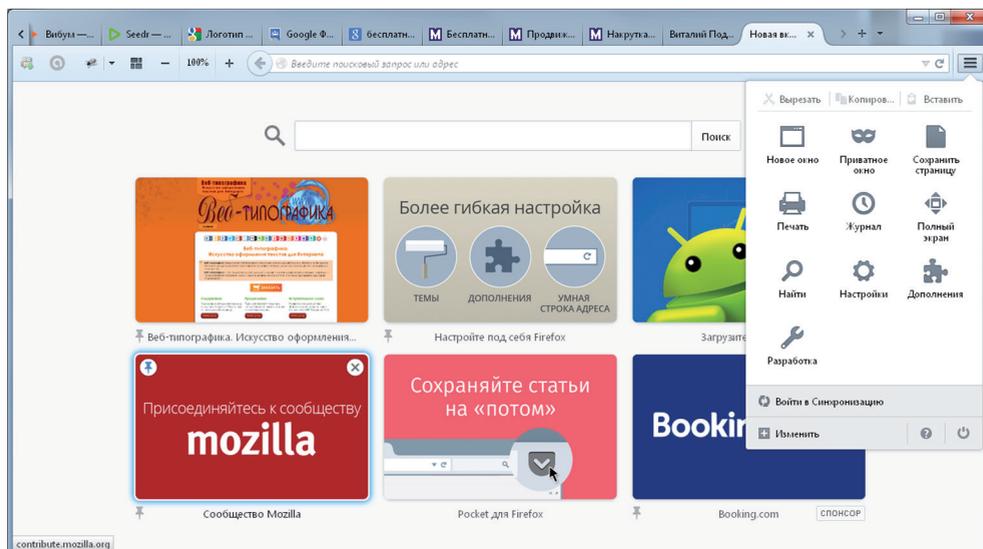


Рис. 2.10. Интерфейс браузера Mozilla Firefox (версия 37.0.1)

для бесплатной установки более тысячи различных дополнений. Разработанные функции безопасности позволяют Firefox максимально защитить компьютер от вредоносных программ (за безопасность отвечают специальные дополнения, например, «NoScript»). Также браузер отличается хорошим блокировщиком всплывающих окон и имеет официальные версии для операционных систем Windows, Mac OS X и Linux.

К незначительным недостаткам браузера Mozilla Firefox можно отнести:

1. Высокую ресурсоемкость и потребление большого объема оперативной памяти.
2. Сброс некоторых установленных дополнений к браузеру после его обновления. Но это скорее недостаток разработчиков того или иного дополнения.
3. Несовместимость с определенными устаревшими видами выполняемых в браузере программ, («скриптов»), с применением которых созданы некоторые сайты. Поэтому отображение их веб-страниц может быть некорректным.

И все же, существенных недостатков Firefox не имеет, зато обладает множеством достоинств. Команда разработчиков оперативно устраняет недочеты и программные ошибки, но пользователю также можно само-

стоятельно оптимизировать браузер под свои нужды с помощью грамотного подобранных дополнений.

На официальном сайте Firefox [18] всегда можно бесплатно загрузить и установить самую последнюю версию браузера.



Apple Safari — браузер, входящий в состав операционных систем OS X и iOS от корпорации Apple. Основан Safari (рис. 2.11) на свободно распространяемом программном коде браузерного движка **WebKit** и в середине 2015 года стабильно занимает четвертое место в мире по числу пользователей (см. рис. 2.6). В

России браузер не столь популярен — по числу пользователей на июль 2015 года он находится на шестом месте и фактически замыкает шестерку браузеров-лидеров.

В июне 2007 года Apple выпустила первую тестовую версию Apple Safari 3.0 для Windows XP и Vista, а в 2012 году разработка Safari для Windows была прекращена (последней версией Safari для Windows была версия 5.1.7, выпущенная 9 мая 2012 года).

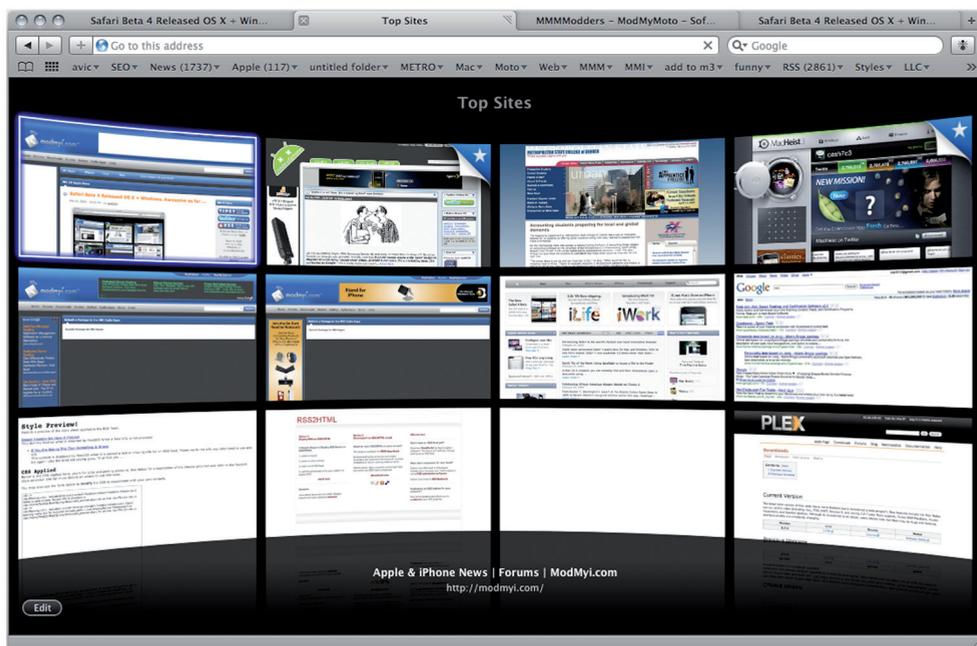


Рис. 2.11. Интерфейс браузера Apple Safari (версия 4.8)

К достоинствам браузера относится его компактный вид, поддержка одновременной работы нескольких вкладок с поиском при помощи встроенных средств (Google в Mac OS X, и Google или Yahoo в Windows), а также предоставление возможности по блокированию всплывающих окон и синхронизации адресных книг Mac OS и Windows. Пользователи, желающие оставаться инкогнито в сети, могут пользоваться функцией частного просмотра, или режима, в котором не ведется истории, а также не запоминаются вводимые данные, включая пароли. В Safari имеется возможность для проверки орфографии в текстах.

К недостаткам браузера можно отнести его «зачточку» только под Mac OS (версий Safari для операционных систем Linux и Android (равно как и для мобильных платформ, кроме iOS) никогда не существовало), медленную скорость при его запуске, а также отсутствие возможностей по копированию всей веб-страницы.



Opera — браузер, выпускаемый компанией Opera Software. Первая версия браузера на браузерном движке **Presto**, разработанного собственными силами разработчиков, была выпущена в 1994 году. Суммарная рыночная доля Opera и Opera Mobile в июле 2015 года составила 1,73 % — пятое место в мире (см. рис. 2.6).

В России процент пользователей браузера гораздо выше мирового — третье место в рунете с долей в 13,53 % (см. рис. 2.7).

Браузер Opera (рис. 2.12) выпускается под основные операционные системы: Windows, Mac OS X, Linux. Существуют также версии браузера для многих мобильных платформ, например для Symbian, Android, Windows Mobile, iOS [19].

Главной достопримечательностью Opera уже много лет была и остается «Экспресс-панель» (панель быстрого доступа, см. рис. 2.12). Позже подобная панель появилась и у других браузеров лидирующей пятерки. Благодаря подобным «изюминкам» (кроме экспресс-панели браузер отличался удобным управлением закладками, историей, наличием своеобразной боковой панели и пр.) браузер Opera долгое время сохранял собственную «индивидуальность» в отличие от остальных браузеров, но в феврале 2013 года разработчики Opera заявили о переходе всей линейки их продукции на **WebKit** в оболочке Chromium [11]. Разработчики также отметили, что в новых версиях будет использован

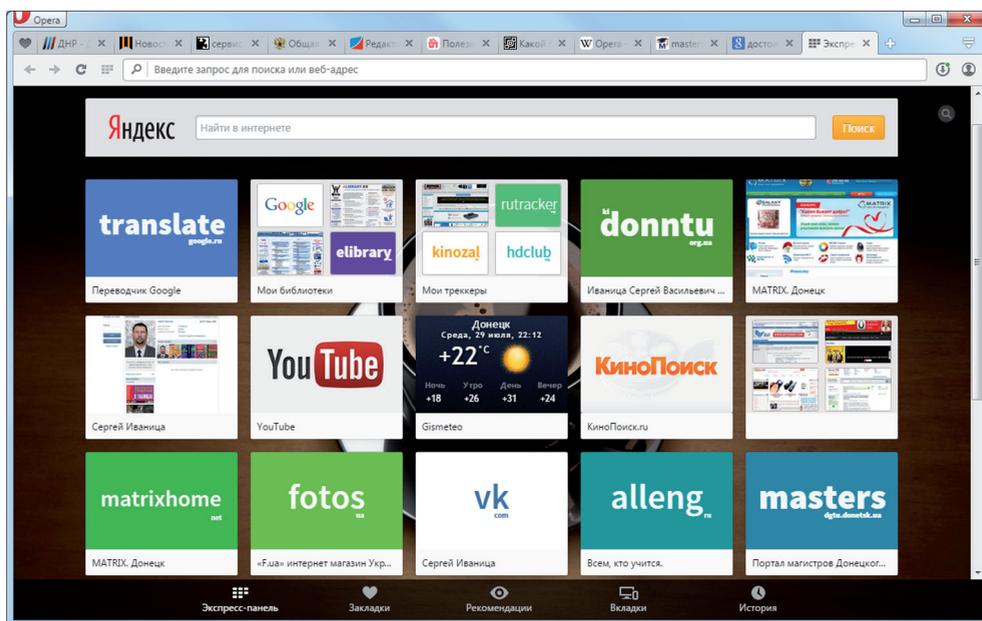


Рис. 2.12. Браузер Opera (версия 30.0.1835) с открытой экспресс-панелью

движок **Blink**, разработанный на основе движка WebKit. Новая ветка браузера (начиная с Opera 15) подверглась серьезной критике со стороны пользователей, главным образом за удаление практически всей прежней функциональности и отсутствие удобства (например, стандартные закладки теперь реализованы в виде особой «копилки» без возможности импорта).

К достоинствам этого браузера можно отнести:

1. Наличие огромного числа всевозможных дополнений.
2. Жесты мышью [20] — управление браузером (обновление, навигация, открытие/закрытие вкладок и пр.) при помощи движений (жестов) мыши, которые преобразуются в команды.
3. Наличие встроенного «компрессора трафика», который уменьшает нагрузку на сеть, что позволяет более эффективно распределять сетевые ресурсы для других программ.
4. Хорошее юзабилити — удобство пользователя при использовании браузера.

Основные недостатки:

1. Частые ошибки, ведущие к перезагрузке браузера и слабая устойчивость к вирусам (версии до Opera 15).

2. Потеря индивидуальности в новых версиях (начиная с Opera 15) и примыкание к Chromium привели к тому, что браузер прозвали очередным клоном Google Chrome.

Текущую версию браузера можно бесплатно загрузить с официального сайта Opera [19].



Яндекс.Браузер — интернет-обозреватель, созданный компанией Яндекс на основе движка **Blink**. Впервые был представлен в октябре 2012 года на ежегодной технологической конференции Yet another Conference, организованной самим «Яндексом». По состоянию на июль 2015 года, доля веб-обозревателя от Яндекса составляет в рунете 9,5 % (см. рис. 2.7).

Поскольку это самый молодой браузер, то вполне очевидно, что разработчики попытались избежать всех ошибок и промахов, которых не удалось избежать другим компаниям с более давней историей создания браузеров. Также разработчики постарались втиснуть в свое детище как можно больше уже откатанных и проверенных на пользователях технологий, перенятых у других браузеров. Среди них, например,

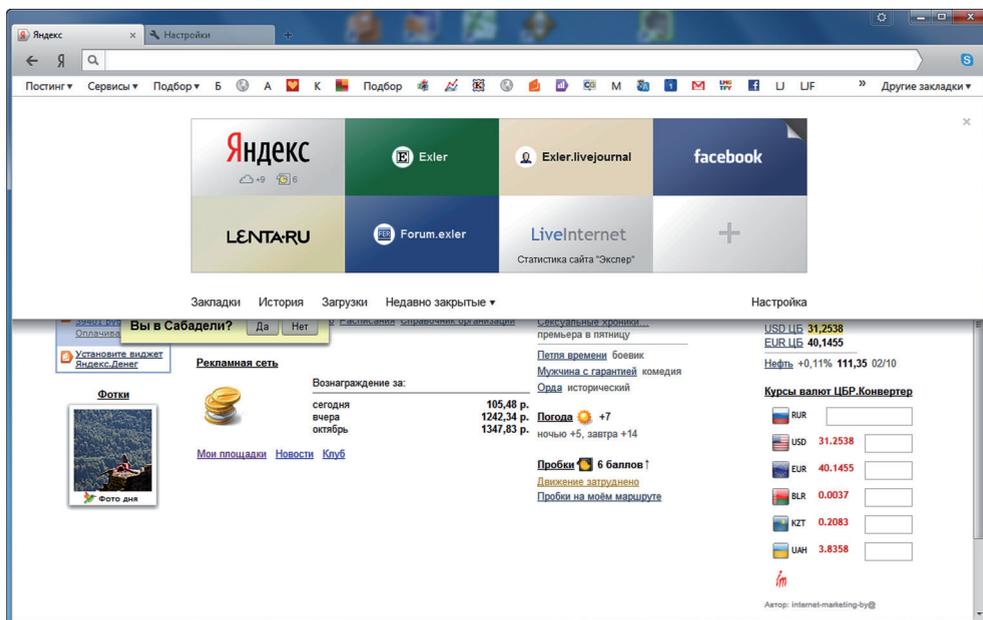


Рис. 2.13. Яндекс.Браузер с открытым Табло поверх текущей страницы [21]

экспресс-панель (у Яндекса это «Табло»), жесты мыши и ускоритель «Турбо» созданные по образу и подобию Орега, «умная» адресная строка, поддержка облачных технологий, отображение doc- и pdf-файлов как у браузера Google Chrome.

Главная особенность этого браузера — интеграция с сервисами самого Яндекса. Кроме того, инновационные технологии антивирусных атак и антиспама в браузере объединены с достижениями Лаборатории Касперского [22]. При загрузке страницы сразу проверяются на наличие вредоносного кода, а файлы — на потенциальную угрозу заражения компьютера. И все же Яндекс.Браузер не лишен недостатков:

1. Перевод веб-страниц на любой из 9 встроенных языков (что само по себе выглядит как достоинство) далеко не идеальный. Сильно уступает тому же переводчику от Google.
2. Слишком упрощенные настройки (недостаток, скорее всего для тех пользователей, которые предпочитают тонкую настройку).
3. Потребление чрезмерных ресурсов операционной системы.

Браузер выпускается в редакциях к различным операционным системам (Windows, Mac OS X, Linux, а также для мобильных Android, iOS) и доступен для бесплатной загрузки на официальном сайте [23].

Рассмотрев шесть самых распространенных браузеров, подведем краткие итоги. Внимательный читатель может справедливо заметить следующее: во-первых, это очень поверхностный обзор браузеров и очень скромные описания достоинств и недостатков каждого браузера; во-вторых, нет сравнительных характеристик основных параметров браузеров (скорость загрузки, объем памяти, настройки и специальные возможности и пр.); и, наконец, в-третьих, до сих пор не понятны принципы выбора идеального браузера.

Давайте разберемся с этим. Каждый браузер имеет свою историю становления и развития, и для каждого из них накопилось достаточное количество технической документации. Всю эту документацию можно найти в Интернете (правда, в основном на английском языке), но она в полном объеме нужна только программистам или «технарям» для узкоспециализированных задач. Далее, любой обзор браузеров или попытка посоветовать лучший — это всего лишь личное пользовательское мнение, которое не всегда подойдет всем без исключения. Ведь на протяжении долгих лет корона «браузерного первенства» переходила к разным

браузерам. Поэтому в каждом веб-обозревателе есть безусловные находки от их разработчиков, равно — как и недостатки, и упущения.

Перед каждым пользователем рано или поздно встает вопрос выбора браузера, ведь браузер является одной из тех программ, которой люди, ежедневно бороздящие просторы Интернета, пользуются чаще всего. Мы — авторы данного пособия, — в силу специфики нашей работы (в том числе и для курса «Интернет-технологии»), пользуемся одновременно всей шестеркой рассмотренных браузеров. И к нашему огромному разочарованию, после длительного перебора целой череды браузеров разных производителей или даже разных версий одного производителя, мы так и не смогли остановить свой выбор на каком-то одном браузере. Это еще и связано с похожестью браузеров. Если еще раз взглянуть на рис. 2.8–2.13, можно заметить много общего в интерфейсе браузеров. Первое, что бросается в глаза, это минимизация элементов управления — все место отдано под окно отображения веб-страниц. Далее — теперь все браузеры имеют «вкладочный» интерфейс (к слову, так было не всегда) — каждая страница может открываться в новой вкладке. Каждый браузер прост в использовании и после установки сразу готов к работе, не требуя особых настроек. Также в каждом браузере есть возможность изменения его внешнего вида и поведения (как правило, это результат манипуляций расширенными настройками). Каждый из них поддерживает современные технологии HTML и CSS, «понимает» Flash, отображает графику и воспроизводит звук. Кроме того, все они способны сохранять веб-страницы, создавать закладки, вести историю и загружать файлы по FTP- или HTTP-протоколам передачи данных с возможностью возобновления загрузки («докачки») при сбоях. Многие браузеры могут открывать doc- и pdf-документы, хранить пароли, создавать многопользовательские структуры, синхронизировать пользовательские настройки и работать с облачными технологиями и сервисами.

Пробуйте, экспериментируйте, и вы быстро найдете для себя оптимальный браузер!

Контрольные вопросы

1. Какое событие считается историческим началом развития браузеров?
2. Как назывался первый распространенный браузер с графическим интерфейсом? В каком году он появился?
3. Когда появилась первая версия браузера Internet Explorer?
4. Чем объясняется быстрая популярность браузера Internet Explorer сразу после его появления?
5. Что такое User-Agent и почему Internet Explorer приходилось «маскироваться» под Netscape Navigator?
6. Что такое браузерная война? Назовите периоды первой и второй браузерных войн.
7. Что такое браузерный движок? Перечислите современные браузерные движки. В каких браузерах они используются?
8. В чем особенности браузера Google Chrome? Перечислите его основные достоинства и недостатки.
9. Какие недостатки у браузера Internet Explorer? В чем заключается неудобство использования ранних версий IE?
10. Какие основные преимущества и недостатки у браузера Mozilla Firefox?
11. В чем особенность браузера Apple Safari? Какие достоинства и недостатки он имеет?
12. Перечислите основные достоинства и недостатки браузера Opera?
13. Что общего у Яндекс.Браузера с браузерами других компаний?
14. Какими общими свойствами обладают все современные браузеры?

2.2. Программы для верстки веб-страниц

Прежде всего давайте рассмотрим основные понятия, связанные с версткой.

Веб-разработка — процесс создания сайта или веб-приложения, включающий в себя следующие основные этапы: веб-дизайн, верстка веб-страниц, создание клиентских и серверных веб-приложений, а также конфигурирование веб-сервера.

О веб-дизайне мы подробно поговорим в следующих разделах, а веб-программирование, к сожалению, вышло за рамки данной книги. Остается определиться с понятием верстки веб-страниц.

Верстка веб-страниц — процесс формирования веб-страниц в текстовом либо визуальном (WYSIWYG-) редакторе.

Далее мы рассмотрим программы-редакторы, специально адаптированные под верстку веб-страниц. Сравним и проанализируем, выделим достоинства и недостатки, сформируем рекомендации к использованию. Но помните, что окончательное решение по выбору того или иного программного обеспечения в процессе веб-разработки вы примете сами (навязывать ничего не будем!). Мы же постараемся подтолкнуть вас к правильному выбору (который у каждого индивидуальный), чтобы сделать процесс верстки приятным и комфортным.

Одним из важнейших инструментов для работы над отдельными веб-страницами и сайтом в целом, является редактор HTML- и CSS-кода. К счастью времена создания страниц сайта в обычном Блокноте Windows (рис. 2.14) бесповоротно прошли, и теперь веб-мастера активно используют различное программное обеспечение, призванное упростить работы над разработкой и версткой веб-страниц.

Если у вас ранее не было опыта веб-разработки и до сих пор вы имели дело только с простейшими операциями, может возникнуть закономерный вопрос: а зачем вообще нужен редактор html /css? Для этого необходимо сфокусировать взгляд на подобные программные продукты с позиции рассмотрения преимуществ, которые дает их применение.



```
Безымянный — Блокнот
Файл Правка Формат Вид Справка
family=Roboto:400,700&subset=latin,cyrillic-ext' rel='stylesheet' type='text/css'>
<link rel="stylesheet" type="text/css"
href="http://www.imagecms.net/templates/image2013/css/style.css?1438455922"
media="all">
<link rel="stylesheet" type="text/css"
href="http://www.imagecms.net/templates/image2013/css/jquery.fancybox-1.3.4.css"
media="all">
<link rel="stylesheet" type="text/css"
href="http://www.imagecms.net/templates/image2013/css/animate.css" media="all">
<!--[if lte IE 8]><link rel="stylesheet" type="text/css"
href="http://www.imagecms.net/templates/image2013/css/lte_ie_8.css" /><![endif]-->
<!--[if IE 7]><link rel="stylesheet" type="text/css"
href="http://www.imagecms.net/templates/image2013/css/ie_7.css" /><![endif]-->
<!--[if lt IE 9]><script type="text/javascript"
src="http://htm15shiv.googlecode.com/svn/trunk/htm15.js"></script><![endif]-->
```

Рис. 2.14. Код HTML в стандартном Блокноте Windows. Берегите время и нервы — не пользуйтесь этой программой для верстки сайтов!

Тем более что значительную часть работы по верстке веб-страницы можно автоматизировать и таким образом сэкономить довольно значительные ресурсы времени.

Используя даже простой редактор для верстки веб-страниц, не говоря уж о сложных программных пакетах, веб-разработчик получает следующие преимущества:

- выделение синтаксиса различных языков программирования и верстки (как правило, цветовое выделение);
- автоматическая вставка часто используемых конструкций (например, автозакрывание тегов — при наборе открывающего тега, автоматически добавляется закрывающий);
- в случае с визуальными редакторами, можно работать даже не обладая знаниями соответствующих языков;
- возможность добавления макросов и плагинов;
- проверка кода на корректность и исправление ошибок;
- одновременная работа сразу с несколькими файлами;
- множество других полезных возможностей, в зависимости от каждой конкретной программы.

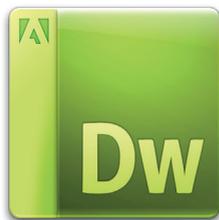
Перед тем как принять решение о выборе в пользу того или иного редактора для верстки необходимо определиться, какой тип редактора вам нужен — визуальный или текстовый? Несмотря на одинаковое предназначение, они совершенно разные в плане подхода к работе. Новичкам, имеющим поверхностное представление о HTML и CSS,

мы рекомендуем использовать визуальный редактор. Но в процессе работы над сайтом, по накоплению опыта верстки и знания кода, нужно непременно переходить к специализированному текстовому редактору, позволяющему полностью контролировать и частично автоматизировать процесс верстки.

2.2.1. Визуальные редакторы

Особенность редакторов WYSIWYG¹ (которые также называют визуальными) состоит в том, что в процессе редактирования содержимое отображается и выглядит максимально близко похожим на конечную продукцию, в нашем случае — на веб-страницу.

Эффективность и удобство использования WYSIWYG-редактора в основном достигается путем взаимосвязанности визуализации и написания разметки (кода). Пока вы работаете с визуальным интерфейсом и буквально «рисуете» ваш сайт, программа по выполненным вами действиям генерирует код файлов сайта. Вы также можете подойти и с «обратной стороны» — писать код и тут же в программе видеть результат своих творений.



В отличие от браузеров, где непросто выбрать идеальный, в WYSIWYG-редакторах есть явный лидер. Таким лидером, несомненно, является программа **Adobe Dreamweaver** [24] — визуальный HTML-редактор компании Adobe. На сегодняшний момент последней версией является Dreamweaver CC (версия 13).

Несмотря на то, что Dreamweaver ориентирован на работу преимущественно в режиме WYSIWYG, по желанию его можно переключать в текстовый режим или разделить интерфейс на две части — визуальную и текстовую (рис. 2.15).

Основная функциональность программы сосредоточена на создании HTML и CSS кода, а также работе с файлами сайта. Визуальный

¹ **WYSIWYG** — аббревиатура от англ. What You See Is What You Get, что буквально переводится как «что видишь, то и получишь».

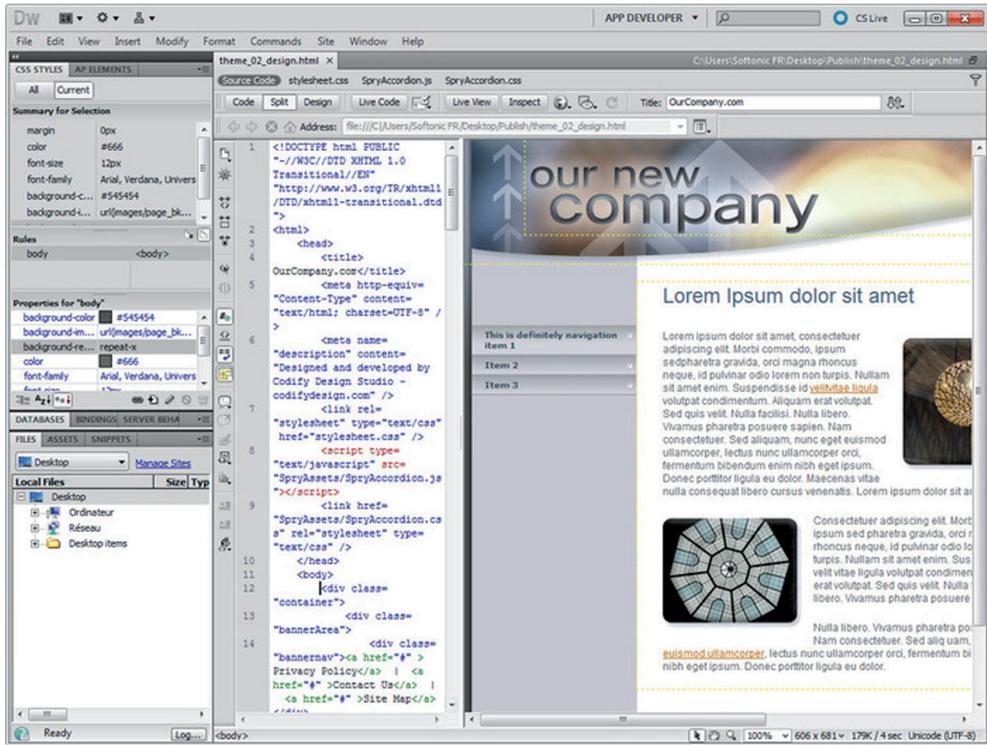


Рис. 2.15. Внешний вид Adobe Dreamweaver с отображением визуальной и текстовой частей, а также панели свойств (слева)

интерфейс WYSIWYG-редактора программы позволяет работать с этими вещами быстро и интуитивно понятно.

Однако давайте взвесим основные достоинства и недостатки программы Adobe Dreamweaver.

Перечислим основные достоинства, позволившие вывести программу в лидеры среди WYSIWYG-редакторов:

1. **Автоматическое дописывание кода** — очень полезная функция (особенно для начинающих веб-разработчиков), которая позволяет экономить время. Когда вы начинаете писать какие-нибудь html-теги или css-свойства, перед вами открывается список возможных вариантов на основе введенных первых букв. Вам остается лишь выбрать подходящий вариант (рис. 2.16). Это позволит новичкам не ошибаться в написании каких-то деталей, а опытным — писать код быстрее.

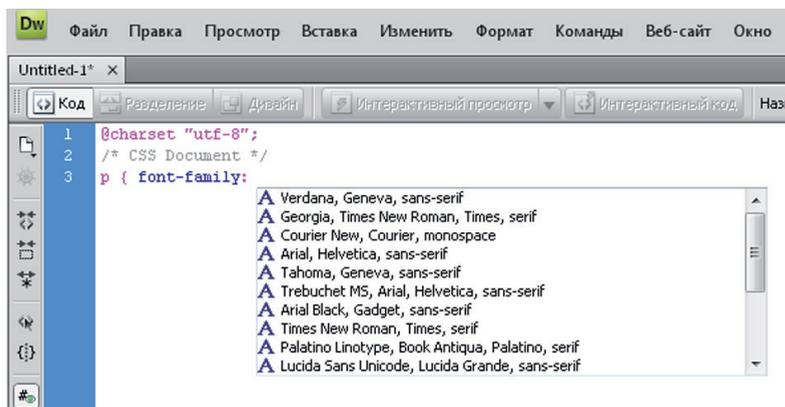


Рис. 2.16. Выпадающее окно для подстановки кода в Adobe Dreamweaver

2. **Проверка кода** — инструмент, который не позволит ни одному веб-разработчику писать код, не соответствующий стандартам. Такая проверка быстро определит ошибки и проблемы совместимости буквально «на лету» — в процессе написания кода.
3. **Файловый менеджер** по праву является той особенностью, благодаря которой Dreamweaver представляет собой нечто большее, что просто редактор кода.
4. **Шаблоны.** Когда создана веб-страница, дизайн которой будет использоваться для всего сайта, ее можно использовать как шаблон, выбирая в ней те области, которые будут редактируемыми и те области, которые будут неизменяемыми на всем сайте.
5. **Интеграция с различными сервисами и службами.** Например, с **BusinessCatalyst** [25] (сервис для размещения сайтов).

Как и любой программный продукт, Adobe Dreamweaver не лишен недостатков:

1. *Запутанный интерфейс* — в окне программы одновременно находятся более 35 элементов, с помощью которых можно выполнять те или иные функции. Очень сложно помнить значение всех этих элементов.
2. *Непродуманная встроенная система обучения*, в которой легко могут запутаться даже профессионалы.
3. *Погрешности визуального редактора.* Нельзя на 100 % доверять отображению сайта в Dreamweaver, т. к. это все-таки не браузер.
4. *Действительно полезными являются только 10 % функционала.*

Таким образом, достоинств у данной программы, несомненно, больше чем недостатков, которые можно считать условными и с которыми можно легко мириться.

И напоследок (что также можно условно считать недостатком) — программа платная и стоит более 600 долларов, а поскольку средний пользователь использует только 10 % заложенного функционала, то покупка программы будет не самым удачным приобретением. Однако если вы все же купили Dreamweaver, то он способен удовлетворить даже самого придирчивого веб-разработчика.



Следующий, заслуживающий внимания визуальный редактор — **KompoZer** [26]. Это бесплатный WYSIWYG HTML-редактор, основанный на исходных кодах редактора **Nvu** [27]. KompoZer по праву считается лучшим из бесплатных редакторов WYSIWYG, доступных на сегодняшний день и лучшей бесплатной альтернативой рассмотренного ранее Adobe Dreamweaver.

Работая в KompoZer, мы можем редактировать страницы в трех режимах:

1. **Режим дизайна** для визуального редактирования страницы и элементов разметки.
2. **Режим кода**, в котором можно работать с исходным кодом страницы как в текстовом редакторе. Однако сразу можно просмотреть страницу в редакторе, как она будет выглядеть в браузере.
3. **Объединенный режим** (рис. 2.17). Очень удобный режим, когда необходимо отработать разметку определенного элемента. В этом случае нам доступны оба вышеперечисленных режима.

KompoZer имеет вкладочный интерфейс (возможна работа с несколькими документами с помощью вкладок), удобное расположение элементов в главном окне программы и выпускается для различных операционных систем (Windows, Mac OS, Linux), а также более чем 20 языковых версий (включая русскую и украинскую). Редактор поддерживает все элементы HTML и в нем также есть встроенный редактор CSS для модифицирования стилей, менеджер для управления файлами сайта, возможность проверки исходного кода веб-страницы (встроенная валидация HTML от W3C) и возможность публикации сайта на сервере (встроенный клиент FTP).



Рис. 2.17. Интерфейс WYSIWYG-редактора KompoZer

К основному недостатку KompoZer относится отсутствие автоматизации ввода кода. В нем нет свойства автозавершения тегов HTML или других способов ускорения ввода тегов. Кроме того, KompoZer выглядит не так презентабельно, как коммерческие редакторы (например, Dreamweaver), но его очень просто освоить и обеспечить качественное визуальное редактирование веб-страниц.

Таким образом, в данном разделе мы рассмотрели два лучших WYSIWYG-редактора — коммерческий и бесплатный. Также заслуживают внимания еще три коммерческих WYSIWYG-редактора: **Artisteer** (редактор шаблонов) [28], **Web Builder** [29] и **Microsoft SharePoint Designer** (замена для Microsoft Office FrontPage) [30].

Но следует помнить, что использование визуального редактора — это самый простой путь потерять контроль над кодом. Когда мы используем только визуальный режим (режим дизайна), то сгенерированный редактором код непременно перегрузится тегами, которые неподконтрольно прописываются и часто являются избыточными. Поэтому визуальные редакторы полезны только для начинающих веб-разработчиков. Но если вы хотите познать все тонкости верстки и досконально освоить

HTML и CSS, то вам придется переходить к верстке веб-страниц в специализированных текстовых редакторах.

2.2.2. Специализированные текстовые редакторы

Специализированные текстовые редакторы мы условно разделим на три группы:

- интегрированные среды разработки (англ. IDE, Integrated Development Environment), включающие в себя возможности редактирования кода HTML и CSS;
- текстовые редакторы, ориентированные на работу с файлами для Веба;
- онлайн-сервисы для создания веб-страниц.

В настоящее время в просторах Всемирной Паутины можно отыскать огромное количество специализированных текстовых редакторов (и их количество продолжает увеличиваться), однако все они обладают достаточной похожестью в рамках одной из наших условных категорий. Поэтому достаточно рассмотреть по одному из представителей каждой, определенной нами группы, чтобы можно было получить достаточное представление о специализированных текстовых редакторах в целом и облегчить выбор подходящего инструмента для верстки сайтов.



Aptana Studio [31] — это полноценная интегрированная среда разработки веб-приложений с большим набором дополнительных программных модулей (плагинов). Дистрибутив программы довольно объемный (более 130 Мб), но это объясняется обширным функционалом. Непрофессионала такое большое количество различных функций и настроек может отпугнуть, но использовать программу в качестве редактора HTML и CSS кода, в принципе, ничто не мешает. Aptana Studio полностью поддерживает CSS3 и HTML5.

Кроме того, эта программа обеспечивает полный спектр сервисов для профессиональной разработки сайта:

- встроенный веб-сервер, который позволяет просматривать динамические веб-страницы с кодом, выполняющимся на стороне сервера, например для php-файлов;
- менеджер проектов с возможностью работы по протоколу FTP;

- встроенный отладчик (debugger) для инспектирования кода HTML, CSS, JavaScript, Rubby on Rails¹;
- встроенный терминал и просмотрщик веб-страниц;
- возможность подключения множества расширений и модулей;
- поддержка Git (распределенное управление версиями), которая крайне полезна в командной работе профессиональных программистов.

Интерфейс Aptana Studio многооконный и обладает возможностью гибкой настройки элементов управления (рис. 2.18). Это значительно облегчает задачу поиска нужных элементов управления, т. к. неиспользуемые элементы можно скрыть, но доступ к ним можно будет получить через главное меню программы.

Еще одним важным достоинством Aptana является расширенная и хорошо структурированная справочная система (на английском языке), которая несет в себе подробно изложенную информацию о работе с программным продуктом и различными расширениями.

К Aptana Studio можно подключать различные расширения, например, такие как **Zen-coding** [33]. При этом открывается возможность ускоренного написания HTML и CSS кода. Zen-coding работает следующим образом: написав в редакторе кода, например, такую предиктивную строковую конструкцию (на манеру CSS записи)

```
div#name>p.one+p.two
```

и нажав определенную комбинацию клавиш, строковая конструкция заменяется текстовым 4-строчным блоком HTML:

```
1 <div id="name">
2   <p class="one"></p>
3   <p class="two"></p>
4 </div>
```

Подобные конструкции определяют синтаксис Zen-coding, инструкция к которому доступна на сайте [33].

¹ **Ruby on Rails** — программная платформа (framework), написанная на языке программирования Ruby. Ruby on Rails предоставляет архитектурный образец «модель-представление-контроллер» (Model-View-Controller) для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером базы данных [32].

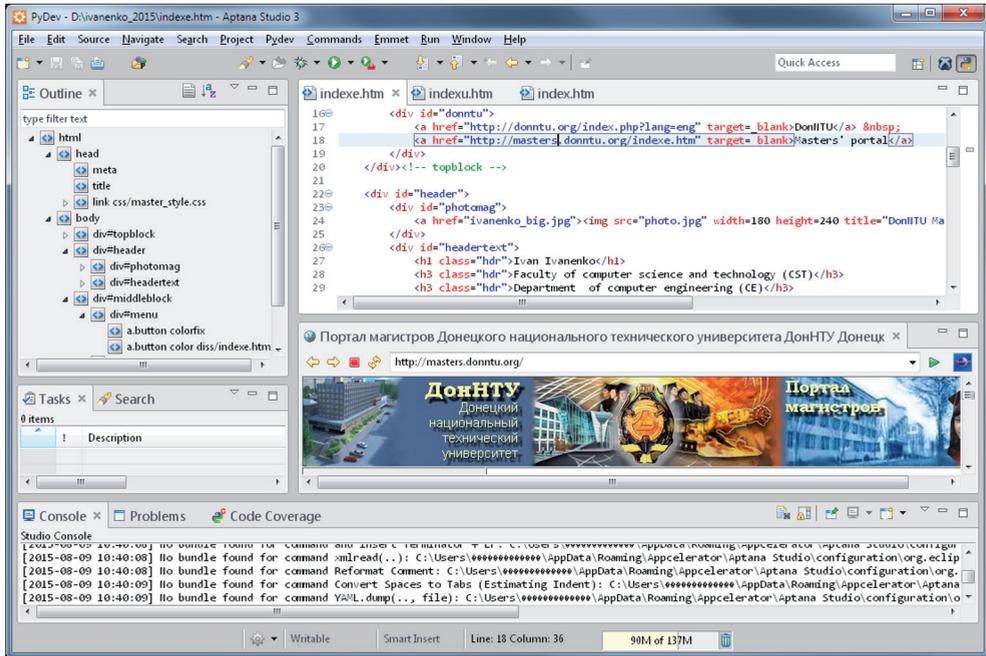


Рис. 2.18. Интерфейс Aptana Studio 3

Помимо самостоятельных версий для Windows NT, Mac OS и Linux, Aptana доступна в качестве плагина для платформы **Eclipse** [34] — свободной интегрированной среды разработки модульных кроссплатформенных приложений.



Notepad++ [35] — свободный текстовый редактор с открытым исходным кодом для Windows (рис. 2.19), который является отличной заменой Windows Notepad. Он поддерживает вкладки и подсветку синтаксиса HTML и CSS. Notepad++ имеет также систему плагинов, позволяющую расширять возможности редактора дополнительными опциями, например, загрузкой через FTP. Он имеет отличную систему поиска и замены, которая включает поддержку регулярных выражений и поиск по списку файлов (групповой поиск).

Базовые возможности Notepad++, которые заявлены на официальном сайте:

- **подсветка синтаксиса** — выделение синтаксических конструкций текста с использованием различных цветов, шрифтов и начертаний;

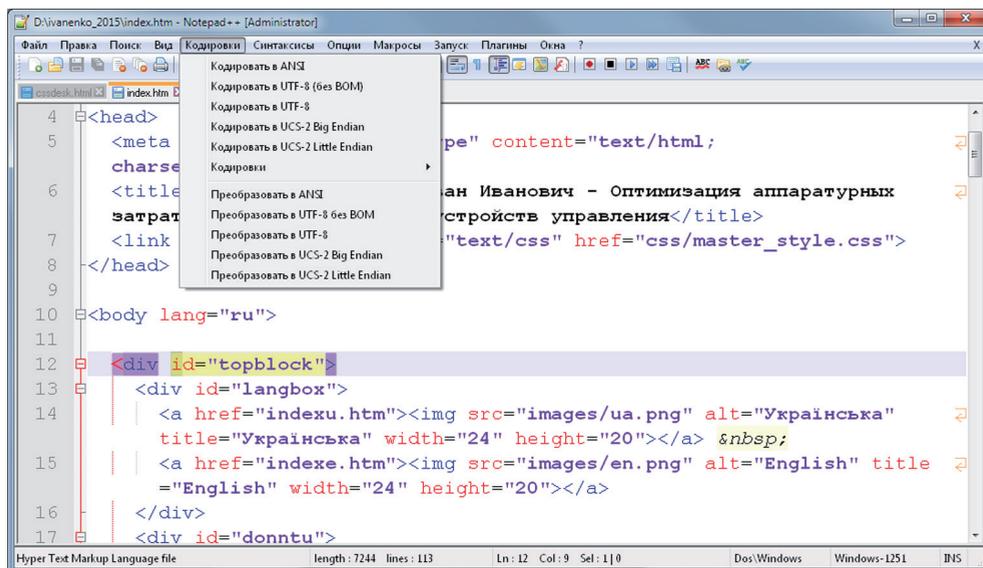


Рис. 2.19. Notepad++ (версия 6.6.9) с активной вкладкой «Кодировки»

- **сворачивание кода** — функция, позволяющая скрывать определенный фрагмент редактируемого кода или текста, оставляя лишь одну строку;
- **автозаполнение** и **выпадающий список подстановок** и параметров функций;
- **закладки** на выбранное место (позицию) в тексте;
- поддержка **регулярных выражений** (формальный язык поиска, основанный на использовании метасимволов или шаблонов) для поиска и замены;
- запись и воспроизведение программных алгоритмов действий, записанных пользователем (**макросы**);
- возможность **сравнения файлов**;
- **менеджер проектов** и **карта документа**;
- переопределение любых **горячих клавиш**;
- **резервное копирование** сохраняемых файлов;
- **блоковое выделение текста** — одновременное выделение текста в нескольких разных местах документа;
- **мультистрочное редактирование** — одновременное редактирование повторяющихся фрагментов текста;
- **шаблоны текста** (сниппеты), вводимые с помощью сокращений (плагин SnippetPlus);

- **менеджер для загрузки данных** на сервер по протоколу FTP (плагин NppFTP);
- **НEX-редактор** — шестнадцатеричный редактор для редактирования данных, в котором данные представлены как последовательность байтов;
- **проверка орфографии** (плагин GNU Aspell);
- симметричное и асимметричное **шифрование текста** (плагин NppDarkCrypt);
- **поддержка Zen Coding** [33];
- поддержка работы с **облачными хранилищами** Microsoft OneDrive [36] и Dropbox [37].

Таким образом, простая с виду программа Notepad++, благодаря обилию встроенных и устанавливаемых отдельно сервисных функций, крайне удобна в использовании для подготовки простых текстовых документов, а также для создания и корректирования кода веб-страниц сайта. Единственный недостаток редактора Notepad++ в том, что он реализован только для операционной системы Windows. Однако в Интернете можно найти подробные инструкции, например [38], рассказывающие об установке Notepad++ на операционную систему Linux.

Программа постоянно развивается: появляются новые функции; своевременно устраняются мелкие программные ошибки и недочеты; часто выходят новые версии. Текущая версия программы бесплатно доступна на официальном сайте проекта Notepad++ [35].

* * * * *

В качестве **онлайн сервиса для редактирования кода HTML и CSS** в браузере (такие сервисы еще называют онлайн HTML/CSS песочницами) рассмотрим сервис **CSSDesk** [39].

Онлайн сервис CSSDesk удобен в случаях когда нужно поэкспериментировать с кодом CSS или HTML, но при этом не хочется создавать новые файлы и прописывать связи между HTML и CSS, а также все необходимые элементы для веб-файлов. Достаточно открыть сайт CSSDesk (рис. 2.20) и сразу начинать набирать код HTML и CSS в соответствующих полях (левая часть окна на рис. 2.20). При этом в процессе создания/правки в коде можно видеть, как меняется итоговый результат на реальной странице сайта (правая часть окна на рис. 2.20).

При желании также можно сохранить набранный код в файл html. В этом случае все необходимые элементы HTML будут добавлены авто-

матически. Например, сохраненный код из рис. 2.20 выглядит следующим образом:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSSDesk</title>
5     <style type="text/css">
6
7       body { background-color: #2574b0; }
8
9       body {
10        font-family: 'Lucida Grande', 'Helvetica Neue',
11                Helvetica, Arial, sans-serif;
12        padding: 80px;
13        font-size: 40px;
14      }
15
16      div {
17        background: #fff;
18        margin: 0 auto;
19        width: 100px;
20        padding: 100px;
21        text-align: center;
22        /* border-radius */
23        -webkit-border-radius: 5px;
24        -moz-border-radius: 5px;
25        border-radius: 5px;
26        /* box-shadow */
27        -webkit-box-shadow: rgba(0,0,0,0.2) 0px 1px 3px;
28        -moz-box-shadow: rgba(0,0,0,0.2) 0px 1px 3px;
29        box-shadow: rgba(0,0,0,0.2) 0px 1px 3px;
30      }
31    </style>
32  </head>
33
34  <body>
35    <div>hello, world!</div>
36  </body>
37 </html>
```

Основные плюсы онлайн редактора CSSDesk:

- раздельное написание кодов HTML и CSS в разных полях;
- визуализация результата в режиме реального времени;
- возможность «поделиться» кодом с постоянной ссылкой на хранилище;
- сохранение кода на локальный компьютер в файлах .html и .css;

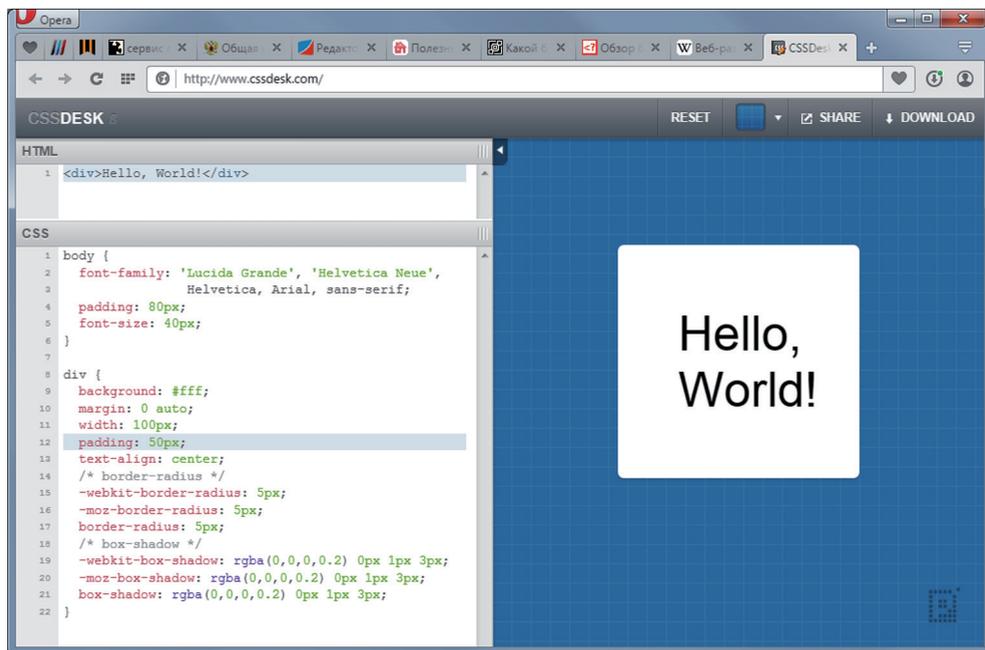


Рис. 2.20. Онлайн сервис CSSDesk

- подсветка синтаксиса;
- смена фона рабочей области.

В настоящее время онлайн-сервисы набирают все большую популярность, постепенно усложняя функционал и улучшая интерфейсы. Также стали появляться облачные онлайн-сервисы для верстки сайтов с возможностью интегрированной работой с файлами веб-страниц используя облачные технологии.

* * * * *

Итак, мы рассмотрели по одному представителю каждой определенной нами группы специализированных текстовых редакторов.

Чтобы принять верное решение и выбрать оптимальный для себя редактор HTML, обращайте внимание на такие вещи как поддержка современных стандартов, наличие функций автоматизации, режим работы (визуальный/текстовый), подсветка и проверка синтаксиса, и т. п. Таким образом, имея четкое представление, что именно вам требуется, выбрать необходимый инструмент будет гораздо проще.

Для облегчения выбора оптимального специализированного текстового редактора, приведем небольшой список **бесплатных программ**

для каждой из категорий, которые были опробованы авторами при создании курса «Интернет-технологии»:

1) Интегрированные среды разработки:

☑ **NetBeans IDE** [40] (Windows, Mac OS, Linux) — интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения. Среда предоставляет все средства, необходимые для создания профессиональных корпоративных и мобильных сайтов и веб-приложений;

☑ **Komodo Edit** [41] (Windows NT, Mac OS X, GNU/Linux) — редактор с открытым исходным кодом, основанный на коммерческом продукте Komodo IDE.

2) Текстовые редакторы:

☑ **Bluefish Editor** [42] (UNIX-подобные, Mac OS, Windows) — открытый текстовый редактор с подсветкой синтаксиса, который предназначен в первую очередь для веб-разработчиков и дизайнеров;

☑ **PsPad** [43] (Windows) — специализированный текстовый редактор (очень похож на Notepad++) и редактор исходных текстов программ для операционной системы Windows;

☑ **jEdit** [44] — кроссплатформенный редактор с открытым исходным кодом, написанный на языке Java (т.е. при установке потребует установить виртуальную машину Java версии 1.5.0 или выше).

3) Онлайн-сервисы:

☑ **JSFiddle** [45] — один из самых популярных инструментов для работы с JavaScript/HTML/CSS: прост, удобен и умеет подключать дополнительные библиотеки;

☑ **CodePen** [46] — инструмент для веб-разработки, позволяющий легко оперировать с CSS-, HTML- и JavaScript-конструкциями (позволяет подключать JS-библиотеки);

☑ **Thimble** [47] — разработка от Mozilla, код и его результат отображается на одной странице в режиме реального времени;

☑ **Liveweave** [48] — очень простая в использовании онлайн-площадка для веб-дизайнеров и разработчиков с поддержкой HTML5 и CSS3.

Контрольные вопросы

1. Что такое веб-разработка? Дайте определение верстки веб-страницы.
2. Что такое WYSIWYG? Расшифруйте аббревиатуру.
3. Какими свойствами должен обладать простой редактор для верстки веб-страниц?
4. Чем достигается эффективность и удобство использования WYSIWYG-редакторов?
5. Перечислите основные достоинства и недостатки визуального редактора Adobe Dreamweaver.
6. Перечислите и поясните режимы работы редактора KompoZer.
7. В чем заключается опасность использования исключительно WYSIWYG-редакторов для верстки сайтов?
8. Перечислите основные группы специализированных текстовых редакторов.
9. В чем заключены основные особенности интегрированной среды разработки Aptana Studio?
10. Что такое Zen-coding? Какое преимущество Zen-coding предоставляет веб-разработчику?
11. Перечислите базовые возможности текстового редактора Notepad++.
12. Определите порядок работы с HTML/CSS песочницей CSSDesk. В чем заключается удобство работы с подобными онлайн-сервисами?

2.3. Программы для обработки графических файлов

Для обработки растровых и векторных изображений для сайта (включая анимированные) нам потребуются соответственно растровые и векторные графические редакторы (а также программы для создания анимации). Особенности растровых и векторных графических изображений и их способы кодирования и хранения в соответствующих компьютерных файлах были рассмотрены в предыдущих главах. Здесь же мы акцентируем внимание на программном обеспечении, необходимом для обработки графических файлов с прицелом на дальнейшее размещение полученных изображений на наших сайтах (способы подключения изображений к веб-страницам также были рассмотрены в ранних главах).

Прежде чем изучать программы для обработки графических файлов, давайте определим следующие понятия:

Растровый графический редактор — специализированная программа, предназначенная для создания и обработки растровых изображений. Такие редакторы нашли широкое применение при подготовке изображений при публикации графики в Интернете, а также печати типографским способом или на фотобумаге.

Векторный графический редактор — специализированное программное обеспечение, позволяющее создавать и редактировать векторные изображения непосредственно на экране компьютера, а также сохранять их в различных векторных форматах для дальнейшей публикации в Интернете или печати типографским способом.

Редактор для создания анимации — специализированная программа для создания компьютерной анимации — мультипликации, создаваемой с помощью компьютерной графики. Компьютерная анимация может быть основана на растровой или векторной графике, а также на фрактальной или трехмерной (3D) графике.

2.3.1. Растровые графические редакторы

Растровые графические редакторы позволяют пользователю рисовать и редактировать изображения на экране компьютера, а также сохранять их в различных растровых форматах (JPEG и TIFF — хранение растровой графики с незначительным снижением качества за счет использования алгоритмов сжатия с потерями; PNG, GIF — сжатие без потерь; BMP (Bit Map) — несжатое «попиксельное» описание изображения).

Большинство современных растровых редакторов содержат векторные инструменты редактирования в качестве вспомогательных, но сохраняют изображения только в растровых форматах.



Ярким представителем и несомненным лидером графических редакторов является коммерческая программа **Adobe Photoshop** [49] — многофункциональный редактор, разработанный и распространяемый фирмой Adobe Systems. Adobe Photoshop (рис. 2.21) соответствует ведущим отраслевым стандартам и позволяет создавать профессиональные изображения.

С Photoshop можно выделять и маскировать сложные элементы изображения, например волосы, удалять любые элементы изображения с мгновенным интеллектуальным заполнением пустой области, создавать великолепные изображения расширяемого динамического диапазона (HDR), раскрашивать созданные объекты, используя кисти с натуральной текстурой и реалистичные наложения цветов, устранять шумы, добавлять зернистость и настраивать виньетирование с помощью самых современных инструментов для обработки фотографий.

По данной программе создано немало книг, статей, тематических форумов и прочей информации в Интернете. Достаточно в строке поиска ввести «Photoshop» и на нас обрушится масса информации об этом замечательном графическом редакторе.

Отфильтровав большой объем описательной информации можно акцентировать внимание на следующих особенностях программы (предполагается, что вы уже знакомы с такими элементарными понятиями Photoshop как «слой», «холст», «палитра», «режим (цветовая модель)», «цветопроба», «стили» и пр.):

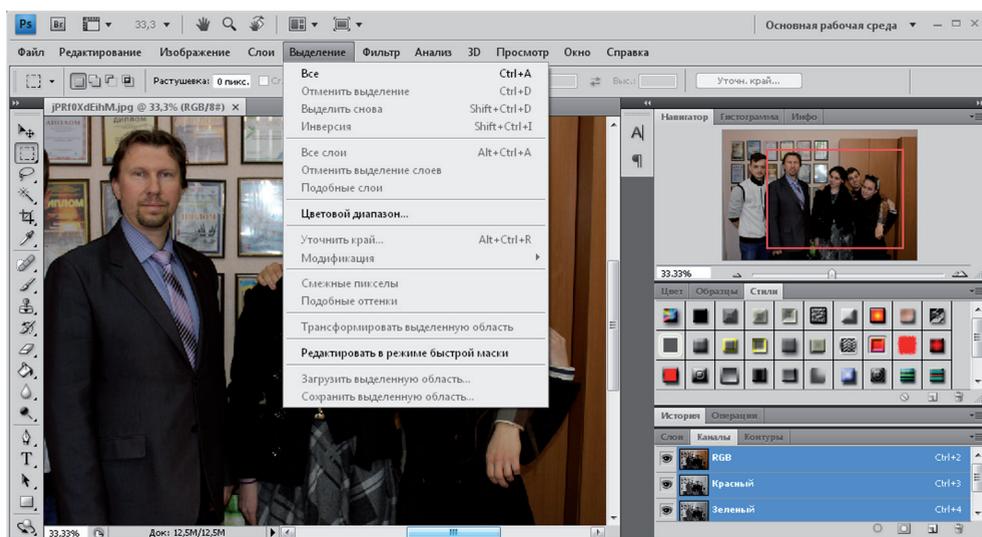


Рис. 2.21. Adobe Photoshop CS4 с загруженным изображением и активной вкладкой «Выделение»

1. Удобный доступ к мощным инструментам редактирования. Выбор необходимого инструмента для операции обработки изображений осуществляется одним щелчком мыши.
2. Интуитивно понятные и «выверенные» инструменты для смешивания цветов на холсте, выполнения мазков с натуральной текстурой, объединения различных экспозиций для создания изображений расширенного динамического диапазона (HDR).
3. Возможность перемещать, удалять, искажать и вытягивать любые элементы изображения.
4. Самые современные инструменты для обработки фотографий: расширенный динамический диапазон изображений; широкие возможности преобразования цветных изображений в монохромные; сохранение деталей цвета и тона при помощи интеллектуальных инструментов.
5. Высокое качество обработки изображений в формате RAW, используя внешний модуль Camera Raw, обеспечивающий поддержку более 275 моделей цифровых камер и функцию обратимого редактирования, которая позволяет экспериментировать, не изменяя при этом исходный снимок.
6. Автоматизированные инструменты композитинга позволяют с легкостью создавать единое изображение из целой серии раз-

- личных снимков, аккуратно смешивая цвета и тени и увеличивая глубину резкости, а также с помощью механизма точного автоматического выравнивания слоев.
7. Эффективное управление медиаданными с помощью программы Adobe Bridge для получения непосредственного доступа к материалам, связанным с обрабатываемым контентом.
 8. Поддержка различных форматов файлов — сотни различных форматов, включая форматы изображений PSD, AI, PDF, NEF, CRW, TIFF, BMP, Cineon, JPEG, JPEG2000, FXG, OpenEXR, PNG, Targa и видеоформаты 3G, FLC, MOV, AVI, DV Stream, Image Sequence, MPEG-4, FLV.
 9. Расширяемость возможностей работы программы путем специально разработанных внешних модулей и ресурсов от сторонних разработчиков, выложенных на сайте Adobe Add-ons [50], а также подробных советов и рекомендаций по работе на Adobe Community Help [51].
 10. Возможность создания 3D-изображений с помощью технологии Adobe Repoussé [52] путем применения к объектам эффектов закручивания, вращения, выдавливания, скоса и искажения, а также имеющимся широким выбором материалов, таких как стекло, хром и пробка, для выбора текстуры трехмерных объектов.
 11. Создание и редактирование анимированного контента путем применения эффектов рисования, добавления текста и клонирования множества кадров видеоряда с помощью специальных инструментов программы Photoshop.

Единственным и можно сказать «условным» недостатком для Adobe Photoshop является отсутствие свободной (бесплатной) версии (программа платная и стоит около 700 американских долларов). В настоящий момент доступна для приобретения версия **Adobe Photoshop CC (2015.0.0)** от 15 июня 2015 года с кодовым именем «Surfing Cow».



В противовес Adobe Photoshop рассмотрим один из популярных бесплатных графических редакторов — **Paint.NET** [53]. Программа Paint.NET является отличной заменой редактору графических изображений Paint, входящему в состав стандартных программ операционных систем Windows.

Среди многих полезных характеристик Paint.NET (рис. 2.22) следует отметить возможность работы со слоями (в том числе и с прозрачными), работу со сканером и цифровой камерой, масштабирование без искажений, удаление эффекта красных глаз, ведение полной истории изменений и т. п. Поддерживаемые форматы: BMP, JPEG, PNG, TIF, GIF и собственный формат PDN.

Особенности (они же достоинства) растрового графического редактора Paint.NET:

- программа **бесплатна для распространения** и использования и оптимизирована для работы с двудерными и четырехдерными процессорами;
- **простой интуитивно-понятный интерфейс**, похожий на Photoshop — все особенности и элементы интерфейса проектировались с расчетом на то, чтобы пользователь быстро разобрался с функциями, элементами управления и сразу приступил работе;

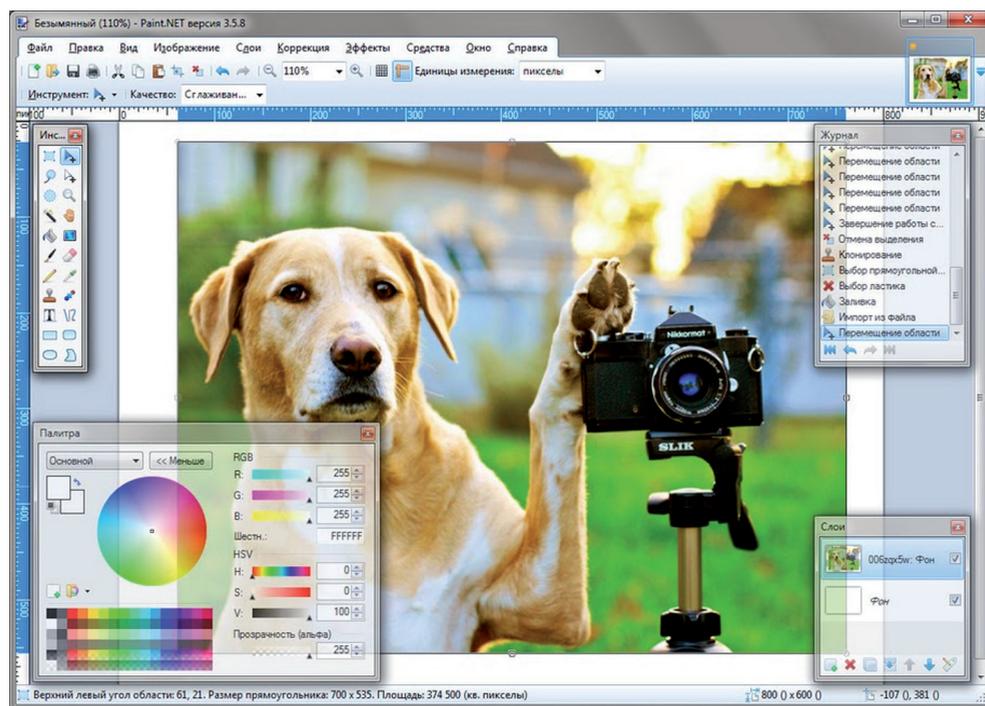


Рис. 2.22. Интерфейс программы Paint.NET (версия 3.5.8)

- **возможность работы со слоями** на уровне дорогих коммерческих программ для работы с изображениями (Paint.NET предоставляет эту возможность бесплатно);
- наличие мощных, но простых в использовании **инструментов для работы с графикой**: «умное» выделение («волшебная палочка»), клонирование изображений, текстовый редактор для надписей, инструменты для масштабирования (от 1% до 3200%), замена цвета и др.;
- **неограниченная история** для исправления возможных ошибок допущенных в процессе редактирования фотографий или рисунков (длина истории ограничивается только доступным дисковым пространством);
- Paint.NET содержит множество **базовых эффектов** для улучшения изображений, управляющих яркостью, контрастом, оттенком, насыщенностью, и т. п., а также в наличии **уникальный эффект** — 3D вращение;
- использование постоянно растущего числа **плагинов**, которые как обогащают возможности программы, так и добавляют новые спецэффекты.

Для корректной работы программы Paint.NET необходимо, чтобы в системе был установлен Microsoft .NET Framework. Также в качестве недостатка можно отметить, что многие пользователи Paint.NET, недовольны изменениями произошедшими в последних версиях этой программы (начиная с версии Paint.NET 4.0). Поэтому пользователю на официальном сайте [53] предоставляется для скачивания последняя версия (на момент написания книги это версия 4.0.5), а также версия 3.5.11, которая обладает большей стабильностью и меньшим потреблением системных ресурсов (ее также можно рекомендовать для пользователей ОС Windows XP).

* * * * *

Рассмотренных в данном разделе платного Photoshop и свободного Paint.NET более чем достаточно даже для самого требовательно пользователя, однако из общей массы можно выделить тройку, опережающих остальные по популярности во всевозможных рейтингах, **бесплатных графических редакторов**:

☑ **Gimp (GNU Image Manipulation Program)** [54] — кроссплатформенный свободно распространяемый растровый графический редактор

(ОС: Linux, Mac OS X, Windows, FreeBSD, Solaris и AmigaOS) для создания, сборки и редактирования изображений. Поддержка слоев (в том числе редактируемых текстовых слоев). Возможность пакетной обработки изображений, работа с экспозицией, полная история работы с изображением, возможность работы с анимированными изображениями.

☑ **Artweaver** [55] — программа (ОС Windows), предназначенная для обработки фотографий и изображений с широким набором художественных эффектов. В наличии широкий выбор настроек для стандартных кистей (мел, древесный уголь, карандаш, акриловые волокна, губка, масло) и фильтров (эффект ветра, размытость, волны, гранулирование, эффект масла и т. д.). Работа со слоями доступна только в «родном» формате программы (AWD). Программа также имеет стандартные инструменты редактирования: градиент, заполнение и выделение (включая «лассо» и «волшебную палочку» как в Photoshop). Бесплатная версия по условиям лицензии не может быть использована для коммерческих целей.

☑ **PhotoFiltre** [56] — компактный универсальный графический редактор для операционной системы Windows. Существуют коммерческая и бесплатная версии программы. В PhotoFiltre можно использовать как стандартные, наиболее часто используемые при редактировании функции (настройка яркости, контраста, насыщенности, цветности, освещенности), так и множество художественных фильтров. Отсутствует поддержка слоев.

Часто при создании сайтов возникает необходимость отредактировать то или иное изображение. При этом если на компьютере не установлен графический редактор, можно несложные операции над изображением выполнить онлайн, воспользовавшись соответственно «онлайновыми» растровыми графическими редакторами. Вот пятерка лучших из них:

☑ **Pixlr** [57] — один из лучших вариантов, который отличается стабильностью, скоростью работы и продуманным интерфейсом. По функциональности очень напоминает Photoshop и содержит почти все базовые инструменты. Предоставляет широкие возможности по работе со слоями и фильтрами. Есть русская локализация. Также Pixlr предоставляет собственный интерфейс программирования приложений (API),

что дает возможность встраивать графический редактор в свой сайт или веб-приложение.

☑ **Aviary** [58] — онлайн редактор от компании Adobe, которые представляет собой набор из 8 онлайн инструментов для работы с растровой и векторной графикой, создания скриншотов, 3D-изображений, графических эффектов, иконок и даже звуковых файлов. Кроме того, Aviary предоставляет онлайн-магазин где можно купить или найти бесплатные шаблоны и графические объекты для своей работы.

☑ **Sumo Paint** [59] — функциональный графический редактор, напоминающий ранние версии Photoshop. Предоставляет полноценную поддержку слоев, разнообразные эффекты и фильтры.

Хотя за последнее время онлайн-графические редакторы сделали огромный шаг вперед, они пока «не дотягивают» до профессиональных десктопных приложений (а до Photoshop им еще очень далеко...).

2.3.2. Векторные графические редакторы

В настоящее время дизайну сайтов уделяется пристальное внимание и веб-дизайнер ценен так же (если не больше) как и верстальщик HTML и CSS. Только с удачным и правильным дизайном сайт будет замечен среди миллионов подобных и найдет своего пользователя. О тонкостях веб-дизайна мы поговорим в следующей главе. Здесь же поговорим об основном инструменте дизайнера (и не только веб-) — векторном графическом редакторе.

Лидерами на рынке векторных редакторов являются две коммерческие программы. Это Adobe Illustrator и CorelDraw.



Adobe Illustrator [60] — это популярнейший графический редактор от компании Adobe Systems, в котором можно создавать векторные иллюстрации для презентаций, печати, веб-страниц. Форматы изображений программы совместимы с Photoshop и другими приложениями Adobe. Последняя версия на данный момент — Adobe Illustrator CC (19.0.1) от 16 января 2014 года поставляется для операционных систем Windows и Mac OS.

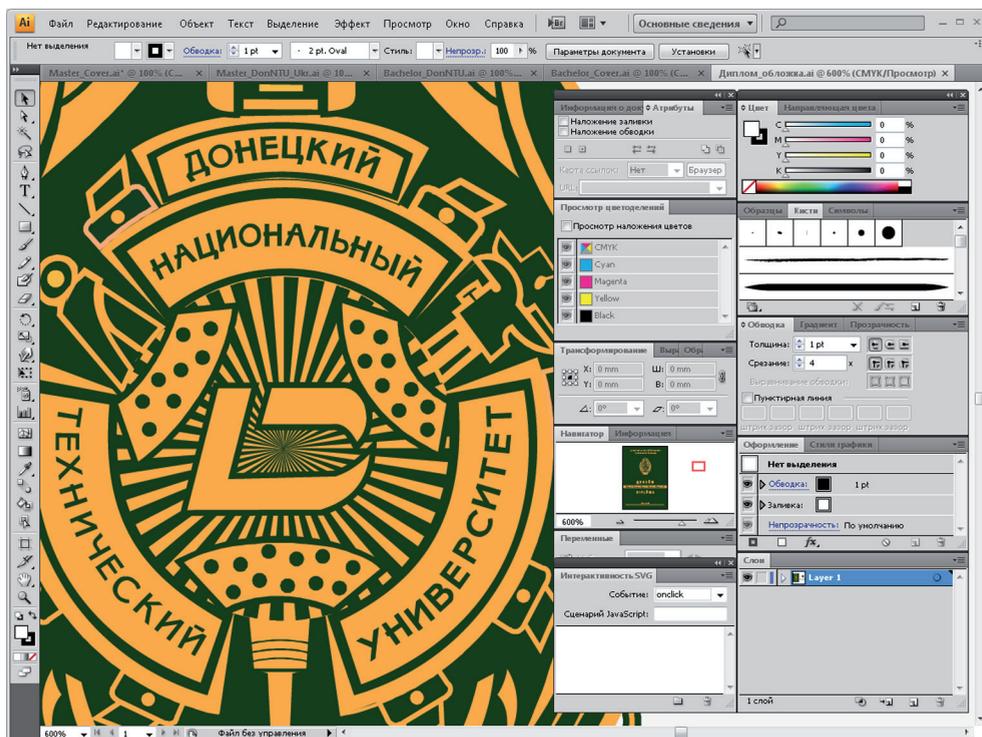


Рис. 2.23. Программа Adobe Illustrator CS4 с загруженным векторным изображением

Как и следует ожидать, Illustrator (рис. 2.23) наследует Adobe-подобный интерфейс, что позволяет легко адаптироваться пользователям, которые продолжительное время работали с продуктами компании Adobe, например, с редактором Photoshop (все программные продукты Adobe имеют некую «общую логику»). Программа обладает широким набором инструментов для рисования и продвинутыми возможностями управления цветом, текстом, что позволяет создавать векторные изображения любого уровня сложности.

Достоинства программы перечислять нет смысла — в ней заложено все, что необходимо для создания и редактирования векторной графики на профессиональном уровне. В то же время активно поддерживается интеграция с различными сервисами в лучшей традиции компании Adobe (в частности Adobe Creative Cloud). Благодаря такой интеграции с Behance можно обмениваться своими проектами с другими пользова-

телями и моментально получать отзывы о своей работе от дизайнеров со всего мира.

Основное достижение Adobe Illustrator в том, что он с недавнего времени взобрался на вершину Олимпа векторных редакторов, потеснив CorelDraw, который был до этого бесспорным лидером. Теперь же разработчики справедливо отмечают, что выбор Adobe Illustrator справедлив, если:

- вы часто работаете с Photoshop и готовы потратить время на освоение Illustrator;
- вам хотелось бы иметь возможность рисовать только с помощью клавиатуры;
- вы планируете сотрудничать с различными общедоступными торговыми площадками (стоковые сайты) или западными клиентами»;
- для вас важно сохранить адекватную цветность и избежать артефактов на рисунке.



CorelDraw [61] — широко известный графический продукт для работы с векторной графикой, разработанный канадской корпорацией Corel. В состав пакета CorelDraw входят 3 редактора: CorelDraw (рис. 2.24) — для векторной графики, Photo Paint — для растровой графики, Corel R.A.V.E. — для анимации.

В разнообразии эффектов и возможностей создания макета CorelDraw не уступает Adobe Illustrator. У него больше инструментов для рисования, за счет которых изображение можно сделать «живее» и интереснее. По скорости работы CorelDraw определенно лидирует Corel и поэтому он более предпочтителен для оперативной печати. Все дело в специфике интерфейса и отсутствии «плавающих палитр», которые у Adobe Illustrator периодически приходится перемещать при работе (см. рис. 2.23 и 2.24).

Если приходится выбирать между Adobe Illustrator и CorelDraw (к слову, на данный момент первая стоит порядка 10 долларов в месяц, вторая — в среднем 350 долларов за программу [62]), то выбор в сторону CorelDraw справедлив, если:

- вы не работаете (или очень редко работаете) с Photoshop и вам важно как можно быстрее освоить редактор векторной графики;

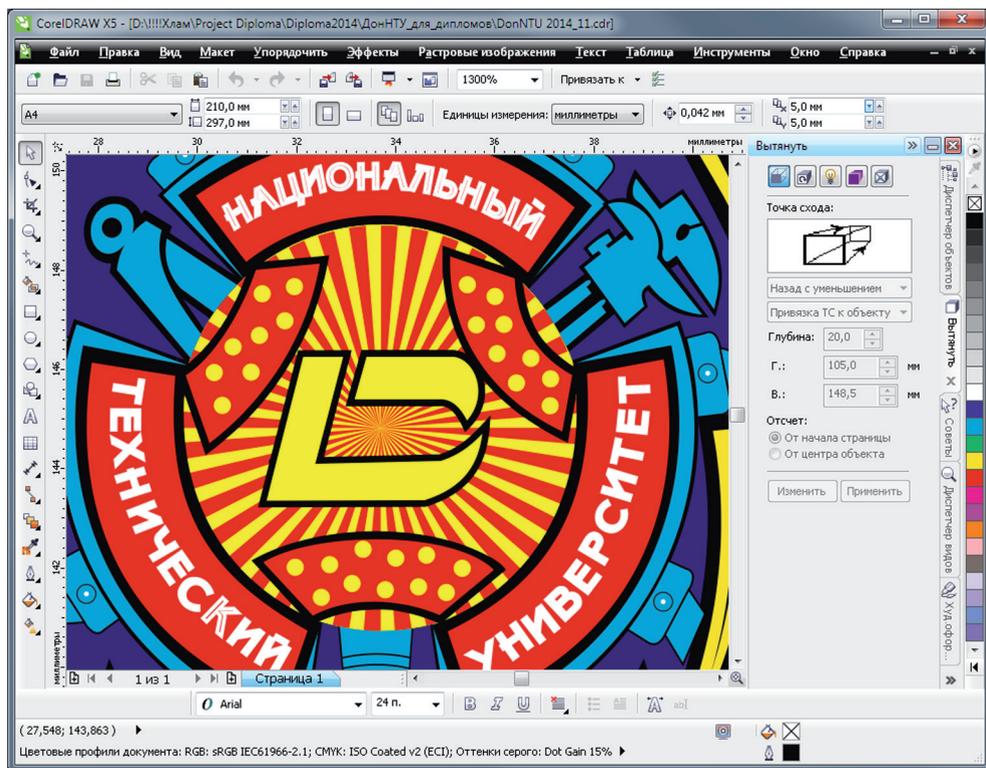


Рис. 2.24. Программа CorelDraw X4
с загруженным векторным изображением

- вы спокойно обходитесь мышью и для вас не принципиальна возможность работы с клавиатуры;
- вы занимаетесь архитектурным проектированием;
- вы готовы потратить время на возможную вычистку артефактов изображения.

Последняя версия программы — CorelDraw X7.5 от 8 июня 2015 года и доступна только для Microsoft Windows.

* * * * *

Рассмотрим теперь некоторые бесплатные векторные редакторы, которые могут стать достойной заменой коммерческих продуктов Adobe Illustrator и CorelDraw:

☑ **Inkscape** [63] — редактор векторной графики, функционально схожий с Illustrator и CorelDraw. Inkscape использует стандарт W3C под названием SVG (Scalable Vector Graphics). Кроме стандартных фи-

гур, в графическом редакторе Inkscape присутствует возможность наносить контуры, текст, маркеры. В наличии также следующие функции: клонирование, управление альфа-каналом, выполнение трансформации фигур, наложение градиентов и текстур, группирование и разгруппирование. Inkscape поддерживает правку узлов, слои, сложные операции с контурами, векторизацию растровой графики, текст по контуру, заверстаный в фигуру текст, редактирование XML-данных напрямую и многое другое. Программа импортирует файлы из форматов JPEG, PNG, TIFF и экспортирует файлы в формате PNG.

☑ **OpenOffice Draw** [64] — векторный графический редактор, по функциональности сравнимый с CorelDRAW, входит в состав свободного пакета офисных приложений OpenOffice.org. Редактор обеспечивает гибкую связность между векторными фигурами, которые могут использовать разнообразные стили линий, что позволяют быстро рисовать чертежи и блок-схемы.

☑ **Xara Xtreme** [65] — векторный редактор, широко используемый для создания разнообразных векторных иллюстраций и веб-графики профессионального качества. Он близок к Adobe Illustrator по функциональным возможностям, но, в отличие от него, отличается большей скоростью обработки данных. Пакет имеет интуитивно понятный, не загроможденный и хорошо продуманный интерфейс, благодаря чему освоить Xara Xtreme на порядок проще, чем Adobe Illustrator. Решение предлагается в двух версиях: базовой Xara Xtreme и профессиональной Xara Xtreme Pro, дополнительно позволяющей работать с многостраничными документами, создавать PDF-документы с поддержкой векторной прозрачности, экспортировать анимации в формат Flash и импортировать/экспортировать многослойные растровые изображения в формате PSD (собственный формат программы Photoshop).

2.3.3. Программы для создания анимации

При подготовке динамических иллюстраций рекомендуется разработка сценария, по которому последовательность кадров, заранее подготовленных в стандартном графическом редакторе, будет чередоваться в строгой последовательности и с определенными временными параметрами так, что, в конечном счете, приведет к задуманному анимационному эффекту.

На этапе сборки последовательности кадров в единое динамическое изображение и настройки их параметров потребуется специализированное программное обеспечение для создания анимации.

Ниже рассмотрим программы для создания простейшей анимации в формате GIF. Для анимации средней и высокой сложности (например, анимации с использованием технологии Flash) допустимо использование специализированных программ, которые достаточно сложны для неподготовленного пользователя и требуют несколько большего срока и усилий.

Создать GIF-анимацию можно в рассмотренных ранее графических редакторах Adobe Photoshop и Adobe Illustrator, однако такой процесс требует качественных знаний и навыков работы с этими программами. В Сети размещено достаточное количество информации и пошаговых инструкций для создания анимации в Photoshop и Illustrator, поэтому выносим создание анимации в них на самостоятельное рассмотрение (или, предположительно, рассмотрим в практикуме по курсу «Интернет-технологии», который выйдет отдельным изданием).

В данном разделе речь пойдет о простейших программах для GIF-анимации, которые не потребуют углубленного изучения и действуют по принципу «открыть и создать».

Самая простейшая программа для данных целей — **MP GIF Animator** от компании Microsoft, которая размещена на портале магистров Донецкого национального технического университета [66]. Программа не нуждается в установке, так что нужно всего лишь загрузить и распаковать программу, чтобы начать ее использовать (рис. 2.25).

Интуитивно понятный интерфейс MP GIF Animator позволит без труда выбрать изображения-кадры для показа, указать время кадра, время повтора и сохранить файл в формате GIF. Это и весь процесс создания анимации в данной программе. Программа обрабатывает растровые форматы (JPG, GIF, BMP), формат иконок (ICO), метафайлов (EMF, WMF) и запускается только в операционной системе Windows.

* * * * *

Также, когда нужно сделать анимацию «на лету», приходят на помощь **сервисы, которые позволяют создать собственные анимированные изображения**. При этом в таких сервисах интерфейс настолько прост, что от пользователя не требуется каких-либо супернавыков

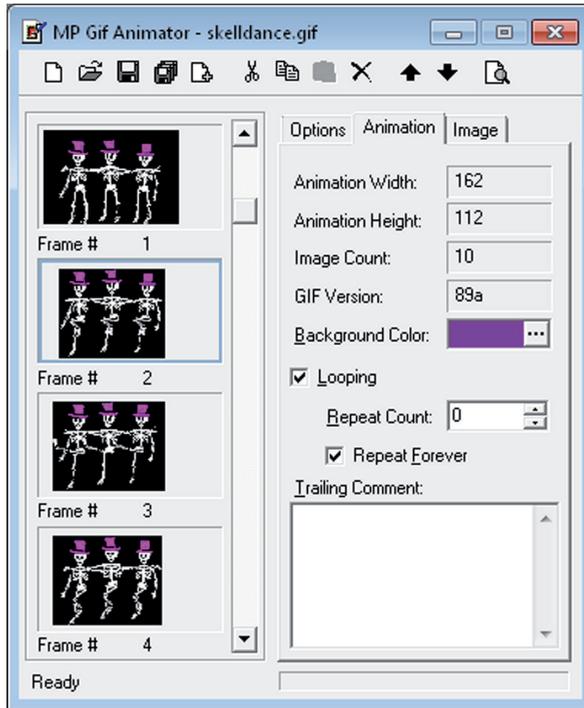


Рис. 2.25. Программа MP GIF Animator (версия 1.0)

владения компьютером. Список данных сервисов огромен и постоянно увеличивается. Чтобы выбрать подходящий, достаточно выполнить поиск в поисковике на запрос «GIF Animator online». Мы же рекомендуем следующие проверенные сервисы: **GIF Maker** [67], **Picasion** [68], **MakeaGIF** [69] и **Gickr** [70].

Контрольные вопросы

1. Что такое растровый графический редактор?
2. В чем особенности векторного графического редактора?
3. На какой графике основана компьютерная анимация?
4. В чем преимущество использования растрового графического редактора Adobe Photoshop?
5. Перечислите особенности растрового графического редактора Paint.NET.

6. Дайте краткую характеристику предложенным бесплатным растровым графическим редакторам.
7. В чем выражаются особенности и основные преимущества векторного графического редактора Adobe Illustrator?
8. Чем отличаются интерфейсы программ Adobe Illustrator и CorelDraw?
9. Какие функции выполняет программа и сервис для создания компьютерной анимации?
10. Какая программа рекомендуется для создания GIF-анимации?

Список литературы к главе 2

1. Тим Бернерс-Ли [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: **http://ru.wikipedia.org/wiki/Бернерс-Ли,_Тим**. — Загл. с экрана.
2. Стивен Пол Джобс [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: **http://ru.wikipedia.org/wiki/Стив_Джобс**. — Загл. с экрана.
3. Лавникевич Д. Первый браузер: 15 лет истории [Электронный ресурс] / Д. Лавникевич // Comprice.ru. — 2006. — Режим доступа: **<http://www.comprice.ru/articles/detail.php?ID=42470>**. — Загл. с экрана.
4. История браузеров: развитие, становление и войны [Электронный ресурс] — Режим доступа: **<http://topbrowser.ru/read/useful/istoriya-brauzerov-razvitiie-stanovlenie-i-voiny>**. — Загл. с экрана.
5. WorldWideWeb [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: **<http://ru.wikipedia.org/wiki/WorldWideWeb>**. — Загл. с экрана.
6. About NCSA Mosaic [Электронный ресурс]. — Режим доступа: **<http://www.ncsa.illinois.edu/Projects/mosaic.html>**. — Загл. с экрана.

7. Сравнение браузерных движков [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: **http://ru.wikipedia.org/wiki/Сравнение_браузерных_движков**.
8. Проект WebKit с открытым исходным кодом [Электронный ресурс]. — Режим доступа: **<http://www.webkit.org>**. — Яз. англ.
9. Internet Explorer 6 умер. Это признала даже Microsoft [Электронный ресурс] — Режим доступа: **http://topbrowser.ru/news/browser_news/internet-explorer-6-umer-eto-priznala-dazhe-microsoft**. — Загл. с экрана.
10. StatCounter — Free Invisible Web Tracker, Hit Counter and Web Stats [Электронный ресурс] — Режим доступа: **<http://statcounter.com>**. — Загл. с экрана. — Яз. англ.
11. The Chromium Projects. [Электронный ресурс] — Режим доступа: **<http://www.chromium.org>**. — Загл. с экрана. — Яз. англ.
12. Google Chrome [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: **https://ru.wikipedia.org/wiki/Google_Chrome**. — Загл. с экрана.
13. Google Code [Электронный ресурс]. — Режим доступа: **<https://code.google.com>**. — Загл. с экрана. — Яз. англ.
14. Браузер Chrome [Электронный ресурс]: официальный сайт. — Режим доступа: **<https://www.google.com/chrome/browser>**.
15. Microsoft Edge [Электронный ресурс]: официальный сайт. — Режим доступа: **<http://www.browserfordoing.com/ru-ru/>**.
16. The Best Internet Browser Software of 2015. Top Ten Reviews [Электронный ресурс]. — Режим доступа: **<http://internet-browser-review.toptenreviews.com>**. — Загл. с экрана. — Яз. англ.
17. Дополнения Firefox [Электронный ресурс]. — Режим доступа: **<http://addons.mozilla.org/ru/firefox>**. — Загл. с экрана.
18. Возьмите контроль в свои руки. Выберите Firefox [Электронный ресурс]: бесплатный веб-браузер — Mozilla. — Режим доступа: **<http://www.mozilla.org/ru/firefox/new/>**. — Загл. с экрана.
19. Браузер Opera — больше возможностей в Интернете [Электронный ресурс]. — Режим доступа: **<http://www.opera.com/ru>**. — Загл. с экрана.

20. Opera/Жесты мышью [Электронный ресурс] // Викиучебник. — Режим доступа: http://ru.wikibooks.org/wiki/Opera/Жесты_мышью. — Загл. с экрана.
21. Блог Алекса Экслера. Яндекс. Браузер [Электронный ресурс] // Экслер.Ру. — Режим доступа: <http://www.exler.ru/blog/item/12550/>. — Загл. с экрана.
22. Лаборатория Касперского. Антивирусы для дома и бизнеса [Электронный ресурс]. — Режим доступа: <http://www.kaspersky.ru>. — Загл. с экрана.
23. Скачайте удобный и быстрый браузер [Электронный ресурс]: официальный сайт Яндекс. — Режим доступа: <http://browser.yandex.ua/ru>.
24. Website builder, html editor. Download free Adobe Dreamweaver CC trial [Электронный ресурс]. — Режим доступа: <http://www.adobe.com/products/dreamweaver.html>. — Загл. с экрана. — Яз. англ.
25. Adobe Business Catalyst [Электронный ресурс]. — Режим доступа: <http://www.businesscatalyst.com>. — Загл. с экрана. — Яз. англ.
26. KompoZer — Easy web authoring [Электронный ресурс]. — Режим доступа: <http://www.kompozer.net>. — Загл. с экрана. — Яз. англ.
27. NVU [Электронный ресурс]. — Режим доступа: <http://www.nvu.com>. — Загл. с экрана.
28. Artisteer — web design software and joomla template maker [Электронный ресурс]. — Режим доступа: <http://www.artisteer.com>. — Загл. с экрана. — Яз. англ.
29. WYSIWYG Web Builder. Конструктор сайтов. [Электронный ресурс]. — Режим доступа: <http://wysiwygwebbuilder.ru>. — Загл. с экрана.
30. SharePoint Designer 2013 — средство для быстрой разработки приложений SharePoint [Электронный ресурс] // Official Microsoft Download Center. — Режим доступа: <https://www.microsoft.com/ru-ru/download/details.aspx?id=35491>. — Загл. с экрана.

31. Aptana Studio 3 [Электронный ресурс]. — Режим доступа: <http://www.aptana.com/index.html>. — Загл. с экрана. — Яз. англ.
32. Ruby on Rails [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: http://ru.wikipedia.org/wiki/Ruby_on_Rails. — Загл. с экрана.
33. Zen Coding — a new way of writing HTML and CSS code [Электронный ресурс]. — Режим доступа: <http://code.google.com/p/zen-coding>. — Загл. с экрана. — Яз. англ.
34. MARS Eclipse [Электронный ресурс]. — Режим доступа: <http://www.eclipse.org>. — Загл. с экрана. — Яз. англ.
35. Notepad++ Home [Электронный ресурс]. — Режим доступа: <http://notepad-plus-plus.org>. — Загл. с экрана. — Яз. англ.
36. OneDrive — cloud storage from Microsoft [Электронный ресурс]. — Режим доступа: <https://onedrive.live.com/about/en-us/>. — Загл. с экрана. — Яз. англ.
37. Drobbox. [Электронный ресурс]: официальный сайт. — Режим доступа: <https://www.dropbox.com/ru/>. — Загл. с экрана.
38. Notepadqq: Notepad++ Clone For Linux [Электронный ресурс]. — Режим доступа: <http://itsfoss.com/notepadqq-notepad-for-linux>. — Загл. с экрана. — Яз. англ.
39. CSSDesk | HTML5, CSS3, JS Demos, Creations and Experiments [Электронный ресурс]. — Режим доступа: <http://cssdeck.com>.
40. Welcome to NetBeans [Электронный ресурс]. — Режим доступа: <https://netbeans.org>. — Загл. с экрана. — Яз. англ.
41. Komodo IDE [Электронный ресурс]. — Режим доступа: <http://komodoide.com>. — Загл. с экрана. — Яз. англ.
42. Bluefish Editor: Home [Электронный ресурс]. — Режим доступа: <http://bluefish.openoffice.nl/index.html>.
43. PSPad — a freeware text editor [Электронный ресурс]. — Режим доступа: <http://www.pspad.com>. — Загл. с экрана. — Яз. англ.
44. jEdit — Programmer's Text Editor — overview [Электронный ресурс]. — Режим доступа: <http://www.jedit.org/index.php>.
45. Create a new fiddle — JSFiddle. [Электронный ресурс]. — Режим доступа: <http://jsfiddle.net>. — Загл. с экрана. — Яз. англ.

46. CodePen — Front End Developer Playground & Code Editor in the Browser [Электронный ресурс]. — Режим доступа: **<http://codepen.io>**. — Загл. с экрана. — Яз. англ.
47. Thimble. Наперсток Мозиллы [Электронный ресурс]. — Режим доступа: **<https://thimble.webmaker.org>**. — Загл. с экрана.
48. Liveweave — HTML5, CSS3 & JavaScript playground for web designers & developers [Электронный ресурс]. — Режим доступа: **<http://liveweave.com>**. — Загл. с экрана. — Яз. англ.
49. Adobe Photoshop CC. Совершенно новая версия 2015 года [Электронный ресурс]. — Режим доступа: **<http://www.adobe.com/ru/products/photoshop.html>**.
50. Adobe Add-ons. Популярныe надстройки Add-on для Photoshop. [Электронный ресурс]. — Режим доступа: **<https://creative.adobe.com/addons?pp=PHSP>**. — Загл. с экрана.
51. Adobe Community Help. [Электронный ресурс]. — Режим доступа: **http://help.adobe.com/en_US/CommunityHelp/**.
52. Photoshop Help / Creating 3D objects and animations [Электронный ресурс]. — Режим доступа: **<https://helpx.adobe.com/photoshop/using/creating-3d-objects-animations-photoshop.html>**. — Загл. с экрана. — Яз. англ.
53. Paint.Net. Официальный сайт [Электронный ресурс]. — Режим доступа: **<http://paintnet.ru>**. — Загл. с экрана. — Яз. англ.
54. GIMP — The GNU Image Manipulation Program [Электронный ресурс]. — Режим доступа: **<http://www.gimp.org>**. — Яз. англ.
55. Artweaver — Home. [Электронный ресурс]. — Режим доступа: **<http://www.artweaver.de/en/>**. — Загл. с экрана. — Яз. англ.
56. PhotoFiltre Studio. [Электронный ресурс]. — Режим доступа: **<http://www.photofiltre-studio.com>**. — Загл. с экрана. — Яз. англ.
57. Photo editor online — Pixlr.com. [Электронный ресурс]. — Режим доступа: **<https://pixlr.com>**. — Загл. с экрана.
58. Aviary [Электронный ресурс]. — Режим доступа: **<https://aviary.com>**. — Загл. с экрана. — Яз. англ.
59. Sumo Paint. Photoshopping in your browser [Электронный ресурс]. — Режим доступа: **<http://www.sumopaint.com/home/>**.

60. Компьютерная графика. Скачать Adobe Illustrator CC [Электронный ресурс]. — Режим доступа: <https://www.adobe.com/ru/products/illustrator.html>. — Загл. с экрана.
61. Программное обеспечение для графического дизайна — CorelDRAW Graphics Suite X7 [Электронный ресурс]. — Режим доступа: <http://www.coreldraw.com/ru/product/graphic-design-software/>. — Загл. с экрана.
62. Сравнение векторных графических редакторов [Электронный ресурс] // Википедия. Свободная энциклопедия. — Режим доступа: https://ru.wikipedia.org/wiki/Сравнение_векторных_графических_редакторов. — Загл. с экрана.
63. Inkscape. Домашняя страница. [Электронный ресурс]. — Режим доступа: <https://inkscape.org/ru/>. — Загл. с экрана. — Яз. англ.
64. Apache OpenOffice Draw. [Электронный ресурс]. — Режим доступа: <http://www.openoffice.org/product/draw.html>.
65. Graphics Software by Xara. [Электронный ресурс]. — Режим доступа: <http://www.xara.com/us/>. — Загл. с экрана.
66. MP GIF Animator. Портал магистров ДонНТУ. Рекомендуемое программное обеспечение [Электронный ресурс]. — Режим доступа: http://masters.dgtu.donetsk.ua/soft/mp_gif_animator/mp_gif_animator.rar. — Загл. с экрана.
67. Gif Maker. Create Animated Gifs Online [Электронный ресурс]. — Режим доступа: <http://freegifmaker.me>. — Загл. с экрана. — Яз. англ.
68. Picasion — Create GIF animations online — Make an Animated GIF [Электронный ресурс]. — Режим доступа: <http://picasion.com>. — Загл. с экрана. — Яз. англ.
69. Make A Gif — Animated Gifs Maker, Free Gif Creator Online [Электронный ресурс]. — Режим доступа: <http://makeagif.com>. — Загл. с экрана. — Яз. англ.
70. Gickr.com — Best gif maker, make a gif in 2 seconds — upload pictures or get images from Flickr [Электронный ресурс]. — Режим доступа: <http://gickr.com>. — Загл. с экрана. — Яз. англ.



В ПОМОЩЬ ВЕБ-РАЗРАБОТЧИКУ

3.1. Основы веб-дизайна

Прежде всего рассмотрим такие базовые понятия как «дизайн» и «веб-дизайн».

Дизайн (англ. *design* — инженер-конструктор) — это:

- 1) творческая деятельность, целью которой является определение формальных качеств промышленных изделий;
- 2) искусство компоновки, стилистики и украшения;
- 3) неразрывная взаимообусловленная связь эстетики и технологий, в которой эстетика задает содержание (суть) вещи или процесса, а технологии — их форму.

Веб-дизайн — отрасль веб-разработки и разновидность дизайна, направленная на разработку и оформление объектов информационной среды Интернета, призванная обеспечить им высокие потребительские свойства и эстетические качества. Веб-дизайнеры проектируют логическую структуру веб-страниц, продумывают наиболее удобные решения подачи информации, а также занимаются художественным оформлением веб-проекта.

Приведенная довольно таки вольная трактовка веб-дизайна отделяет его от веб-программирования, подчеркивает специфику предметной деятельности веб-дизайнера, позиционирует веб-дизайн как вид графического дизайна [1].

Поэтому и **веб-дизайнеры** — это не только те люди, которые имеют профессиональное образование в области веб-дизайна, а и те, которые в процессе создания сайтов подбирают шрифты для текстов и уделяют внимание цветовой гамме веб-страниц. Профессиональный веб-дизайнер полностью «рисует» сайт и на его плечах лежит привлекательность сайта в целом. Такой дизайнер должен уметь рисовать что называется «в цифре» — уверенно работать с растровой и векторной графикой, досконально владеть профессиональными графическими редакторами (например, рассмотренные ранее Photoshop и Illustrator), уметь оценивать и подбирать всевозможные цветовые и геометрические сочетания, а также быть «в теме» новейших тенденций дизайна.

В рамках курса «Интернет-технологии» к веб-дизайну сводится процесс выбора шрифтового начертания для текстов веб-страниц сайта и подбор гармонирующих цветовых сочетаний текста и фона.

3.1.1. Общие характеристики шрифтов

Несомненно, шрифт занимает ключевое место в веб-дизайне, поскольку выступает в качестве основного показателя привлекательности и удобочитаемости текста. В русском языке слово «шрифт» (нем. *schrift* от слова *schreiben* — писать) как название основного носителя информации можно определить следующим образом:

Шрифт — комплект символов определенного размера и рисунка (типографских литер), предназначенных для набора текста.

В настоящее время в распоряжении веб-дизайнера огромное многообразие шрифтов, которое с каждым днем пополняется новинками. Тем не менее, большинство этих шрифтов можно разделить на шесть стилей/категорий (рис. 3.1) [2].

Формальные характеристики каждой из представленных категорий — степень контрастности, относительная высота букв, форма кон-

цевых элементов и засечек, общая ширина букв и буквенного просвета и т. п. — значительно отличаются по сравнению друг с другом. Иногда эти различия кажутся незначительными, но эффект в целом влияет на общее впечатление, создаваемое шрифтом и, соответственно, на ассоциации, которые он может вызвать у аудитории.

Старый стиль (Oldstyle или Serif)	Рубленый стиль (Sans serif)
Современный стиль (Modern)	<i>Рукописный стиль (Script)</i>
Брусковый стиль (Slab serif)	Декоративный стиль (Decorative)

Рис. 3.1 Основные шрифтовые стили

Классификация шрифтов по категориям позволяет провести стилевые различия, создает общую организацию и в дальнейшем поможет выбрать подходящий шрифт для проекта.



Старый стиль (Oldstyle/Serif) основан на примерах рукописных работ, начиная с XV века. Все шрифты старого стиля имеют засечки, причем засечки строчных букв всегда располагаются под углом, что имитирует наклон пера в рукописных текстах. Благодаря такому подражанию рукописи все изогнутые линии буквенных форм имеют переход от толстого штриха к тонкому.

Если провести линию через самые тонкие части закругленного штриха, то эта линия будет диагональной. Эта линия называется нажимом — шрифт старого стиля имеет диагональный нажим. Благодаря своей «неброскости» шрифты старого стиля считаются лучшими для набора текстов больших объемов. У таких шрифтов крайне мало отличительных особенностей, мешающих процессу чтения. Таким образом, если необходим шрифт для набора «длинного» текста (с высокой долей вероятности, что его будут читать полностью), то лучше предпочесть шрифт старого стиля.

В шрифтах **современного стиля (Modern)** присутствуют засечки, но они выполнены горизонтально, без какого-либо наклона и очень тонко. Имея вертикальный нажим, внешний вид шрифта современного стиля

вертикальный нажим
засечки горизонтальные тонкие
Modern
резкий переход от толстого штриха к тонкому

отличается элегантностью и некоторой холодностью. Современные шрифты выглядят очень эффектно, если они выполнены в крупном размере. Однако из-за резкого контрастного перехода от толстой линии к тонкой шрифты современного стиля — не

самый лучший вариант для набора объемных текстов. Из-за резкого перехода от толстой линии к тонкой возникает так называемый «слепающий» эффект.

засечки горизонтальные толстые вертикальный нажим
Slab serif
отсутствие перехода от толстого штриха к тонкому

Шрифты **брускового стиля (Slab Serif)** отличаются наличием толстых засечек и слабым переходом (или полным его отсутствием) от толстой линии к тонкой. Эту группу шрифтов иногда называют

Claderton, поскольку шрифты были составлены на основе одноименного шрифта. Кроме того, подобные шрифты называют и египетскими, так как они стали популярными в период «египтомании» в Европе (конец XVII — начало XVIII вв.), а многим шрифтам этого стиля были даны египетские названия, чтобы они лучше продавались (например, Memphis, Cairo, Scarab). Многие брусковые шрифты, которые имеют очень слабый контрастный переход от толстой линии к тонкой, занимают ведущее место по шкале читабельности, то есть идеально подходят для набора объемных текстов. Вместе с тем страница, набранная таким шрифтом, выглядит значительно темнее страницы, набранной шрифтом старого стиля, поскольку штрихи брускового шрифта плотнее и более динамичны по толщине. Благодаря понятному и незатейливому внешнему виду брусковый стиль шрифта часто используется при наборе текстов, предназначенных для детской аудитории.

засечки полностью отсутствуют
Sans serif
отсутствие заметного перехода от толстого штриха к тонкому отсутствие нажима

Шрифты **рубленого стиля (Sans Serif)** — это шрифты, у которых на концах штрихов букв нет засечек (слово «sans» в переводе с французского означает «без», поэтому «*sans*

serif» буквально — «без засечек»). Удаление засечек — один из самых поздних шагов в эволюции шрифтов — не пользовалось популярностью

вплоть до начала XX века. Рубленые шрифты почти всегда единообразны по толщине (буквы имеют одинаковую толщину по всему объему) из-за отсутствия перехода от толстой линии к тонкой.

Группа шрифтов **рукописного стиля (Script)** состоит из шрифтов, которые были выполнены вручную специальной ручкой или кисточкой для каллиграфии, карандашом или технической ручкой. Эта группа подразделяется на следующие подгруппы: *рукописные шрифты с соединительными штрихами*;

рукописные шрифты без соединительных штрихов; *рукописные шрифты, имитирующие ручной почерк*; *рукописные шрифты, сильно напоминающие традиционные каллиграфические стили*, и т. д. Конечно, объемные тексты не следует набирать рукописными шрифтами, и, разумеется, их нельзя использовать в тексте, состоящем из одних только прописных букв. Но вместе с тем, подобные шрифты, набранные в крупном размере, могут быть особенно эффективными.

Шрифты **декоративного стиля (Decorative)** легко узнаваемы: типографские литеры украшены декоративными элементами. Да и сама идея прочесть целую книгу,

набранную декоративным шрифтом может вызвать головокружение. Именно с помощью таких шрифтов можно придать тексту своеобразное настроение. Декоративный стиль создан специально для усиления «декоративности» в тексте, поэтому декоративные шрифты уместно применять как выделительные: для заголовков (там, где уместны декоративные заголовки), небольших по объему текстов (например, в поздравительных открытках) и т. п. Прежде чем использовать такой шрифт, следует подумать, подходит ли он для передачи того смысла, который вкладывается в проект.

На сегодняшний день сложилась терминология, позволяющая охарактеризовать каждый конкретный шрифт по всем его параметрам.



В совокупности эти параметры составляют **общую характеристику шрифта** [3]:

☑ **Гарнитура (Type family)** — совокупность шрифтов, объединенных общими, отличными от других шрифтов, стилевыми признаками. Некоторые гарнитурные семейства располагают большим количеством начертаний, чем другие. Благодаря этому можно построить весь документ на одной гарнитуре, используя, где это необходимо, различные варианты начертаний. Примером может служить шрифт Helios, который насчитывает около 33 начертаний.

☑ **Начертание (Type face)** — комплект строчных и прописных знаков, цифр, знаков препинания, специальных знаков и символов. Начертания шрифтов любой гарнитуры отличаются цветовой насыщенностью, пропорциями, контрастностью и наклоном знаков. Наиболее распространенные начертания: *Normal* (обычный), *Bold* (полужирный), *Italic* (курсив, или наклонный), *Bold Italic* (полужирный курсив, или наклонный), *Condensed* (узкий) и *Extended* (широкий).

☑ **Насыщенность** шрифта определяется изменением толщины основных и соединительных штрихов одноименных знаков в различных начертаниях. Непрерывный ряд начертаний составляют: светлое (*light*), нормальное (*regular, book*), полужирное (*demi*), жирное (*bold*), темное (*heavy*), черное (*black*) и сверхжирное (*extra bold*).

☑ **Пропорции** — показатель изменения ширины одноименных знаков в начертаниях одной гарнитуры от сверхузких до сверхшироких. Шрифт может быть узким (*condensed*), нормальным (*normal*) и широким (*extended*).

☑ **Кегль** — это величина площадки, на которой размещается знак (буква). Такую площадку называют кегельной площадкой знаков. Кегль, или размер (*size*) шрифта определяется его высотой, измеренной в типографских пунктах (*point* или *pt*): 12 pt = 1 п. (пик), 6 п. = 72 pt = 1 дюйм (рис. 3.2). Здесь пик (от англ. *pica*) — основная единица в англо-американской системе измерения шрифтов. Шрифты, все кегельные площадки знаков (литер) которых имеют одинаковую ширину, называют моноширинными (*monospaced fonts, monowide fonts*), в отличие от пропорциональных (*proportionally spaced type*) шрифтов, в которых литеры отличаются по ширине друг от друга.

☑ **Удобочитаемость** определяется скоростью (быстротой восприятия) и удобством чтения как отдельных знаков, так и текста в целом,

а также правильностью понимания прочитанного без лишнего напряжения и повышенной утомляемости. Удобочитаемость отдельных гарнитур шрифтов в значительной степени определяется квалификацией и навыками читателей.

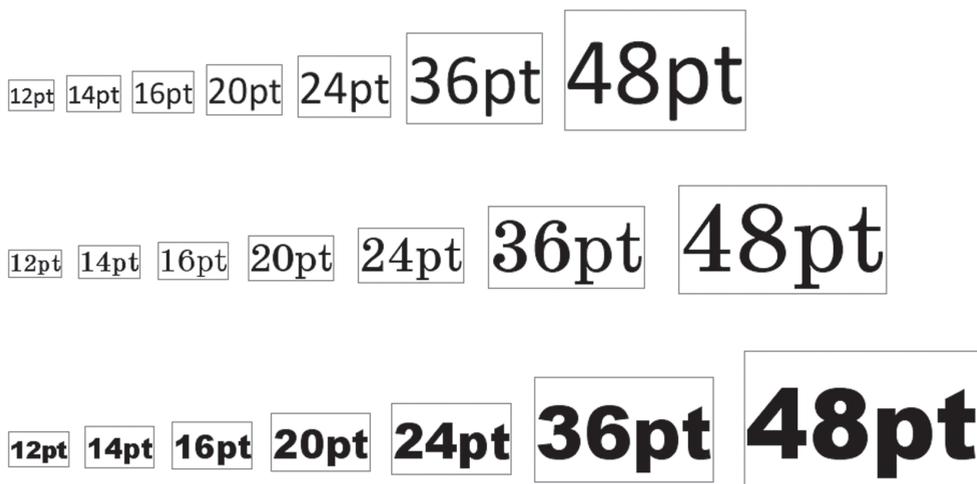


Рис. 3.2. Изменение размера шрифтов в типографских пунктах

3.1.2. Правильный подбор шрифта

Выбор идеально подходящего шрифта является непростой задачей, ведь часто нужно перебирать неисчерпаемый запас вариантов в сочетании с красотой и сложностью шрифта.

Хотя не существует строгих правил по выбору лучшего шрифта, есть много проверенных на практике принципов, которые можно (и нужно) применять при выборе шрифтов. Если постоянно использовать нижеприведенные рекомендации, можно выбрать подходящий шрифт в кратчайшие сроки.

Первое, что нужно сделать для того чтобы выбрать шрифт, — сформировать четкое представление о том, какую реакцию на читателей должен оказать искомый шрифт. Это и есть та отправная точка выбора шрифта, которая будет определять весь процесс. В любом случае, выбор шрифта должен представлять собой идеальное (или близкое к нему) сочетание **четкости** и **читабельности**, оставаясь при этом приемлемым для читательской аудитории.

Четкость относится к дизайну шрифта, например, к ширине штрихов, наличию засечек, наличию нового типа элементов дизайна и т. д. При разборчивом шрифте легко отличить одну букву от другой. Например, декоративные шрифты имеют низкую четкость потому, что они в первую очередь должны бросаться в глаза, а не читаться «в длину». И наоборот, шрифты, предназначенные для романов или газет (т. е. читающиеся «в длину»), имеют очень высокую четкость.

Поэтому, по отношению к четкости, при выборе шрифта нужно обратить внимание на:

- *стандартность знаков*: символы с уникальными формами, художественными деформациями, чрезмерными украшениями или другими элементами дизайна заставляют читателей сначала обрабатывать увиденное и мешают сразу воспринимать информацию;
- *межбуквенный интервал*: близкое расположение букв замедляет процесс восприятия информации;
- *высоту строчных знаков (x-height)*: небольшая высота строчных знаков уменьшает четкость некоторых букв.

Однако высота строчных букв не должна быть большой. Размер шрифта и ширина должны выбираться в зависимости от высоты строчных букв.

Расположение букв шрифта с сочетанием его четкости определяет уровень читабельности, т. е. уровень динамического взаимодействия стиля, размера, межбуквенного и межстрочного расстояний, цвета и других характеристик, объединенных в одно общее впечатление. Эти характеристики складываются в определенный стиль подачи текста, который обладает количественной степенью читабельности. Так как в большинстве случаев **восприятие текста важнее, чем его стиль**, то **читабельность имеет более высокий приоритет**, чем дизайн шрифта.

Для повышения читабельности, для определенных целей нужно выбирать те шрифты, которые и были разработаны для них (например, шрифты для заголовков, шрифты для основного текста и т. д.). С одной стороны, шрифт, предназначенный для привлечения внимания, например, в заголовках, означает, что его нельзя так же эффективно применить в качестве шрифта для основного текста (т. е. текста для «обильного» чтения). С другой стороны, шрифт, предназначенный для длительного чтения, теряет свое воздействие на читателя, представляя, например,

заголовки (т. е. не смотрится в больших размерах). Также необходимо убедиться, что расстояние между строками больше, чем размер выбранного шрифта, особенно для многострочного текста.

Некоторые шрифты подходят только к определенной задаче дизайна. Его уместность можно оценить, изучив применение конкретного шрифта на практике, а также информацию о шрифте, например, его историю и изначальную функцию. Оценка уместности шрифта формируется четырьмя параметрами:

☑ **Назначение дизайна.** Большинство популярных шрифтов имеет детальные обзоры, так что стоит узнать хоть что-то о выбранном шрифте. Так сформируется мнение о его принадлежности либо к шрифту для надписей или заголовков, либо для основного текста. Поиск и просмотр в Интернете информации о шрифте займет всего несколько минут, зато эта информация поспособствует грамотному подбору шрифта или, в крайнем случае, просто снабдит дополнительной информацией.

☑ **Эстетика.** Искомый шрифт должен соответствовать эстетике, ожидаемой аудиторией, для которой предназначен дизайн.

☑ **Настроение.** Представляет собой динамичный синтез эстетики шрифта с читабельностью, а также, конечно, со смыслом самого текста. Например, и одним шрифтом и одним текстом можно вызвать определенное настроение: волнение или панику. Прежде всего, благодаря читабельности и дизайну шрифт вызывает определенную реакцию, но и текст сам по себе может поднять сообщение на новый уровень.

☑ **Технические характеристики.** Данный параметр отображает возможности шрифта в зависимости от наполнения текста различными (часто небуквенными) символами. Например, если в тексте много цифр, то нужно обязательно удостовериться в том, что эти цифры будут отображаться должным образом: некоторые шрифты размещают цифры в нижнем регистре (высота цифры равна высоте строчного знака), а некоторые шрифты используют для этого верхний регистр цифр (высота цифры равна высоте прописного знака).

Таким образом, применяя эти советы на практике, можно добиться хороших результатов, которые в свою очередь, поднимут качество будущих решений по выбору шрифта на новый профессиональный уровень.

3.1.3. Компьютерные шрифты

С учетом различных интерпретаций шрифта, следует различать:

- шрифт как набор символов, созданный художником;
- шрифт как компьютерный файл;
- шрифт как набор металлических литер, используемых для типографской печати наборным способом.

Дадим определение понятию «компьютерный шрифт»:

Компьютерный шрифт — это файл, содержащий в себе описание набора буквенных, цифровых, служебных и псевдографических символов, используемый для их отображения (в частности, в виде текста — полной и связной последовательности символов) определенной программой или операционной системой.

По способу отображения все компьютерные шрифты делятся на два типа: **растровые** и **векторные**.

В растровых шрифтах каждый символ является, по сути, обычным точечным рисунком, который описан в виде набора точек (пикселей), расположенных в узлах сетки растра. Эти шрифты непригодны для высококачественной печати и используются в основном в программах с текстовым интерфейсом и в консоли. Широкое использование растровые шрифты получили в эпоху матричных принтеров и мониторов низкого разрешения.

Символы в векторных (или контурных) шрифтах представляют собой криволинейные контуры, описываемые математическими формулами, — каждый знак описан с помощью векторов, образующих контур знака без привязки к абсолютному размеру или разрешению. Такое описание позволяет увеличить масштаб изображения без потери качества, что невозможно в случае с растровыми шрифтами. Однако для вывода векторного шрифта на растровые устройства его необходимо **растеризовать** — преобразовать в набор точек. После этого векторные шрифты одинаково выглядят как на экране, так и на бумаге. При этом растровые шрифты используются для отображения шрифтов на экране, а векторные — для печати.

Существует несколько различных форматов векторных шрифтов, различающихся способом хранения и представления информации о шрифте: **PostScript Type 1**, **TrueType**, **OpenType**.

☑ **PostScript Type 1** — это общепринятый стандарт для цифровых шрифтов (ISO 9541). Шрифт формата Type 1 — специальная форма программы PostScript и особый формат файла, который ориентирован на описание шрифта. Шрифты PostScript могут содержать до 220 печатаемых символов, т. е. не поддерживают стандарт Unicode. В последние годы язык PostScript был расширен, чтобы обеспечить поддержку шрифтовых возможностей стандартов TrueType и OpenType. Поэтому шрифты типа PostScript Type 1 состоят из двух файлов: растровый шрифт и векторный файл шрифта. Растровые шрифты имеют расширение .fon. Шрифты PostScript Type 1 традиционно используются профессионалами графического дизайна. Для вывода на устройствах с низкой разрешающей способностью в ОС Windows встроен растеризатор Type1-шрифтов.

☑ **Шрифты TrueType** могут содержать до 65 000 символов, расположенных в порядке, определенном стандартом Unicode. Формат TrueType использует специальную технику обеспечения качества представления шрифта при выводе на устройствах с низкой разрешающей способностью, таких как экран компьютерного монитора. При выводе на растровые устройства шрифт TrueType использует развитый набор инструкций, позволяющий добиться того же качества, что и у растровых шрифтов. Шрифт в формате TrueType — это один файл, который в системе Windows имеет расширение .ttf.

☑ **Шрифты OpenType**, по существу, являются контейнерами, построенными по правилам формата TrueType, который может содержать как PostScript-, так и TrueType-данные. Кроме того, шрифт может содержать и новые типы данных, которых не было в формате TrueType. Шрифты OpenType, содержащие данные PostScript, имеют расширение .otf (аббревиатура шрифта: OT/PS), а шрифты на основе TrueType — уже упомянутое расширение .ttf (аббревиатура шрифта: OT/TT). формат OpenType был совместно разработан компаниями Adobe Systems и Microsoft Corporation — глобальными лидерами в сфере разработки программного обеспечения.

Вопрос об **эффективном управлении шрифтами** (быстрая установка и удаление, поиск «потерявшихся» гарнитур, решение проблемы конфликта шрифтов и др.) рано или поздно встает перед начинающими веб-дизайнерами. Вопрос этот крайне важен, потому что от умения управлять файлами шрифтов во многом зависят как продуктивность работы, так и ее конечный результат.

Программы для управления компьютерными шрифтами (их еще называют **менеджерами шрифтов**, или **фонт-менеджерами** (от англ. *Font manager*) — обязательный инструмент для опытного дизайнера. Профессиональные менеджеры шрифтов могут многое, удобны в работе, но являются платными (например, Fontonizer и FontExpert). Однако обычным пользователям могут не понадобиться все возможности профессиональных менеджеров шрифтов. Да и для многих дизайнеров работа с фонт-менеджерами часто сводится к простейшим операциям: установка, удаление и удобный просмотр шрифта. Для выполнения таких несложных операций можно использовать **бесплатные программы**, которые обладают практически полным функционалом платных аналогов.

Если на компьютере одновременно установлено слишком много шрифтов, появляются две серьезные проблемы. Во-первых, замедляется работа операционной системы, которой приходится постоянно отслеживать все шрифты (избыточное количество установленных шрифтов может вызвать «зависание» компьютера). Во-вторых, список в меню Fonts (Шрифты) для приложений становится настолько длинным, что затрудняет выбор необходимого шрифта.

Основными отличиями сторонних менеджеров шрифтов от встроенных средств операционной системы являются следующие возможности:

1. **Активация и деактивация шрифтов.** При этом появляется возможность работы со шрифтами, не устанавливая их на компьютер и, следовательно, не перегружая системные папки шрифтов.
2. **Временная установка шрифтов**, при которой шрифты остаются в памяти компьютера до первой его перезагрузки. При этом никаких записей в реестр не производится.
3. **Упорядочивание шрифтов и объединение их в наборы** для конкретной задачи. При этом удобно использовать определенный набор гарнитур или специальный комплект шрифтов, которые привязаны к какой-нибудь отдельной программе.
4. **Автоматическая установка необходимых гарнитур** в соответствии с типом открываемого документа.
5. **Группировки шрифтов согласно внутренней логике.** Поскольку операционная система хранит все шрифты в одной папке (или в нескольких труднодоступных папках), менеджер шрифтов помогает группировать их по различным критериям: по формату, по историческому стилю, по тексту, по типу отображения и т. п.

Рассмотрим несколько бесплатных программ — менеджеров шрифтов для различных операционных систем:

☑ **Linotype FontExplorer X** (рис. 3.3) [4] объединяет в себе функции управления установленными шрифтами и обзора новых шрифтов с возможностью их покупки через интегрированный в программу онлайн-магазин. Можно сгруппировать установленные шрифты с помощью библиотеки по папкам, тэгам и «умным наборам» (smart sets). Упрощается просмотр всех гарнитур от одного определенного производителя или гарнитур, помеченных определенным тегом. Интеграция программы в системе позволяет включать/отключать доступ шрифтов для конкретных приложений. В программе FontExplorer можно посмотреть полный набор символов, узнать о легальности использования определенной гарнитуры или просто увидеть на примере нескольких параграфов как будет выглядеть тот или иной шрифт в текстовом блоке.

Программа работает на платформах Windows и Mac OS. На данный момент программой поддерживаются английский и немецкий языки.

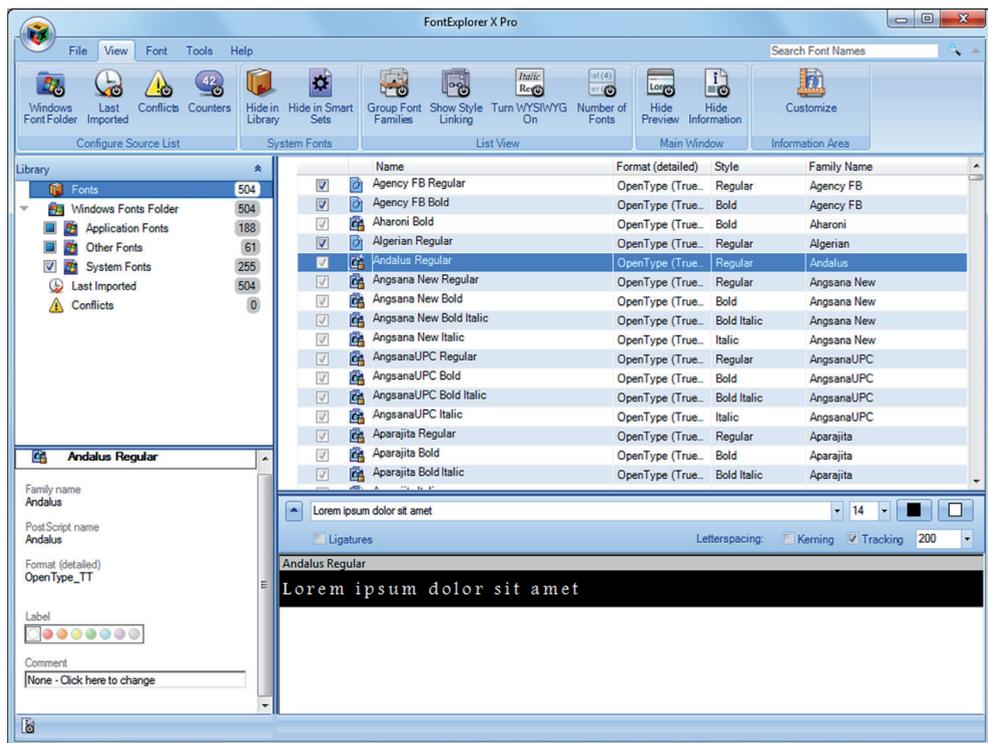


Рис. 3.3. Программа Linotype FontExplorer X

☑ **Opcion Font Viewer** (рис. 3.4) [5] — бесплатная утилита с открытым исходным кодом, написанная на языке программирования Java. Позволяет просматривать установленные и деинсталлированные шрифты, отмечать часто используемые шрифты и предварительно просматривать их в виде настраиваемого образца текста.

Главное преимущество программы Opcion заключается в том, что она работает на всех платформах. Данный инструмент предлагает просмотр по умолчанию, позволяя просматривать шрифты по одному, а также просмотр в виде списка.

Программа позволяет сохранить список избранных шрифтов, который можно использовать для будущих проектов или для поиска файлов шрифтов при выборочной установке. Для работы с Opcion Font Viewer необходимо установить программный пакет Java [6] версии 1.4.0. и выше.

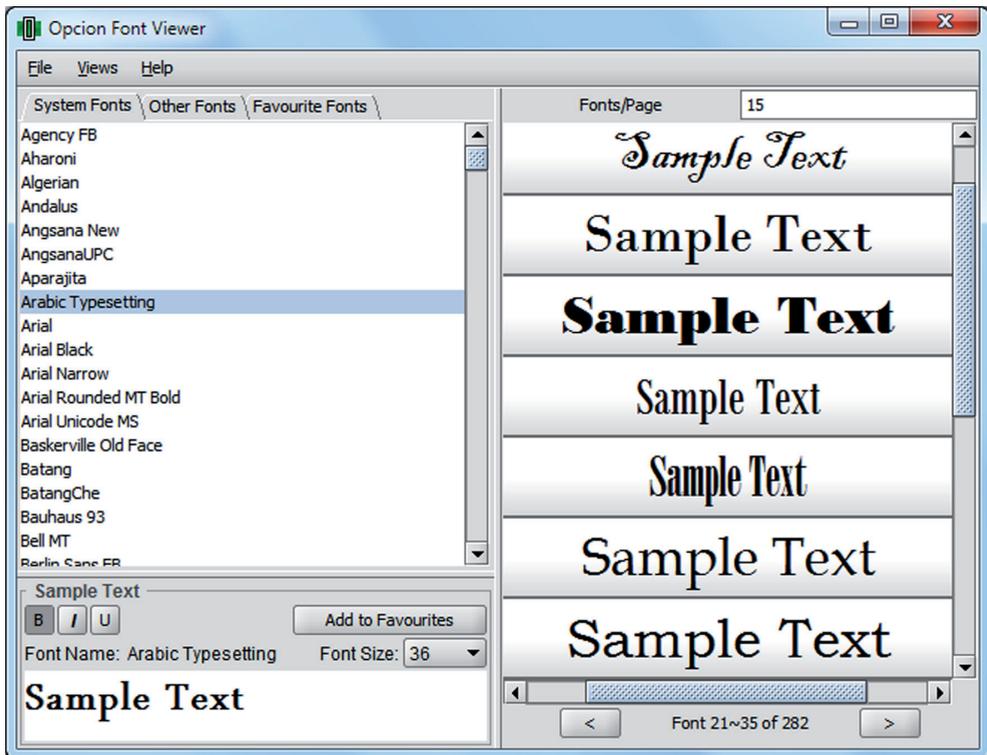


Рис. 3.4. Программа Opcion Font Viewer

☑ **FontFrenzy** (рис. 3.5) [7] имеет ряд специфичных функций. Функция FrenzySnap позволяет устанавливать «точки восстановления» и переустанавливать те шрифты, которые находились в системе в определенный момент. Функция DeFrenzy удаляет все шрифты, кроме оригинальных шрифтов из поставки ОС Windows или до предыдущего состояния системы. Можно создать временную папку куда будут сохраняться все удаленные шрифты.

Основные возможности программы:

- сохранение снимка состояния папки со шрифтами, которое используется как точка восстановления;
- простая переустановка шрифтов по создаваемой точке восстановления;
- просмотр списка шрифтов с помощью настраиваемой текстовой фразы;



Рис. 3.5. Программа FontFrenzy

- просмотр примера каждого шрифта во всплывающем окне, включая все символы и числа верхнего и нижнего регистра;
- добавление новых шрифтов, удаление и сохранение в отдельной папке выбранных шрифтов.

☑ **Adobe Type Manager (ATM) Light** (рис. 3.6) [8] — «облегченная» бесплатная версия популярной программы ATM Deluxe, предназначенная для редактирования шрифтов.

ATM Light имеет множество настроек форматирования, поддерживает все типы шрифтов, позволяет добавлять шрифты в операционную систему, а также отключать или окончательно удалять из системы. Все установленные при помощи ATM шрифты размещаются в отдельной папке, которую можно указать в настройках программы. Программа разработана в двух «релизах» — для работы на платформах Windows и Mac OS.

☑ **Font Xplorer** (рис. 3.7) [9] работает как с установленными, так и с неустановленными (полученными из различных источников шрифтовыми файлами) шрифтами TrueType.

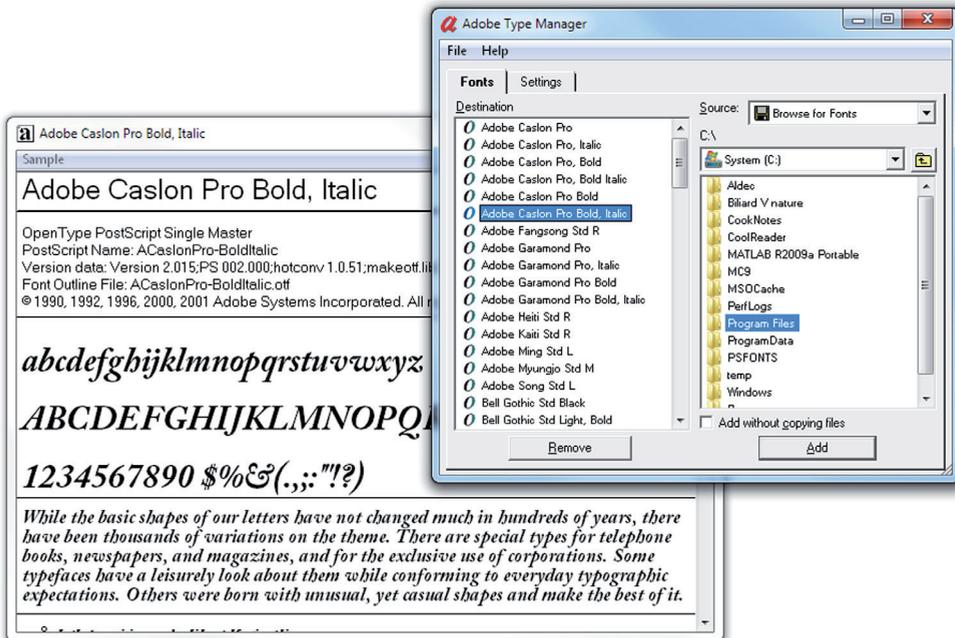


Рис. 3.6. Программа ATM Light с запущенным собственным просмотрщиком шрифтов

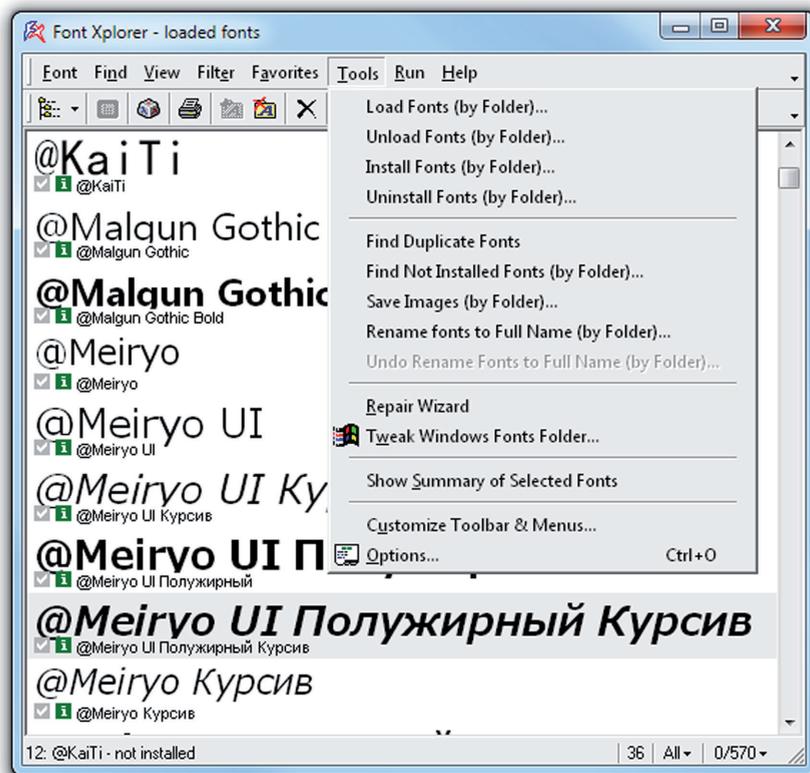


Рис. 3.7. Программа Font Xplorer с активным выпадающим списком меню «Инструменты» (Tools)

Одной из отличительных особенностей программы является переименование файла шрифта с возможностью возврата к исходному имени файла (например, для использования полного названия шрифта можно переименовать файл `aparajbi.ttf` в `Aparajita Bold Italic.ttf`).

Еще одна особенность Font Xplorer — возможность копирования символа шрифта в буфер обмена в виде векторного изображения или в формате RTF. Также возможна фильтрация шрифтов по кодировке, чтобы видеть только шрифты символов одной кодовой таблицы.

Интегрированный Мастер Ремонта (Integrated Repair Wizard) позволяет решить большинство проблем со шрифтами.

☑ **AMP Font Viewer** (рис. 3.8) [10] — простой менеджер файлов для работы с установленными и с неустановленными шрифтами. Поддерживает возможность системной и временной (до закрытия программы)

установки шрифтов. Также присутствуют стандартные опции для организации шрифтов по категориям и управления ими.

С помощью AMP Font Viewer можно распечатать список всех или части установленных шрифтов с примерами для каждого из них. В приложении есть так называемый Черновик (Scratchpad) для проверки того, как выглядят символы шрифта, а также опции для группирования шрифтов и управления этими группами.

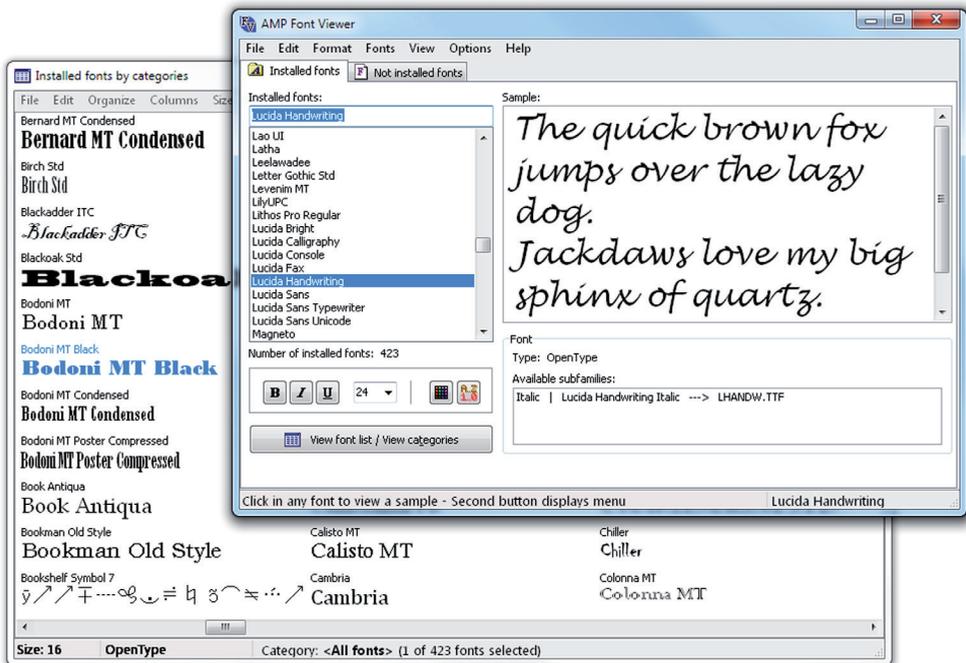


Рис. 3.8. Программа AMP Font Viewer с запущенным окном списка установленных шрифтов (Installed fonts by categories)

3.1.4. Шрифтовые онлайн-сервисы

В начале 90-х годов в дизайне ощущалась острая нехватка компьютерных шрифтов как по количеству, так и по качеству, однако последнее время эта ситуация резко изменилась. Появилась огромная масса разнообразнейшего шрифтового материала любых стилей и категорий. Появились интернет-ресурсы (веб-сервисы, сайты, программы), предоставляющие услуги по хранению, поиску и предоставлению разнообраз-

ных шрифтов, в том числе и кириллических. Эти «шрифтовые электронные библиотеки» можно условно поделить на две категории (не считая сервисы, занимающиеся «пиратским» распространением коммерческих шрифтов): сервисы первой категории безвозмездно и безвременно предоставляют шрифтовые файлы (изначально свободные и нелицензионные шрифты) для использования их на персональных компьютерах; сервисы второй категории «торгуют» лицензионными шрифтами — предоставляют право на их использование по соответствующему лицензионному соглашению.

Приведем несколько интернет-ресурсов, представляющих собой сайты для загрузки бесплатных шрифтов на персональный компьютер.

☑ **Сайт XFont.RU** [11] предоставляет для скачивания как бесплатные, так и лицензионные шрифты (рис. 3.9). На сайте организованы расширенный поиск шрифтов и просмотр раскладки выбранного шрифта. Сформирован список самых просматриваемых шрифтов «Топ 100». Русские шрифты выделены в отдельный раздел, который обновляется и пополняется свежими шрифтами. На сайте минимум рекламы, что выгодно отличает его от конкурентов. В разделе «Статьи» можно узнать об истории шрифтов, познакомиться с их авторами, а также скачать шрифты, используемые в статье. Раздел «Полезные программы» включает в себя программы для просмотра и редактирования шрифтов.

База шрифтов на Xfont.RU пополняется ежедневно и на данный момент составляет более 40 000 шрифтов.

☑ **Портал шрифтов «RuFont»** [12]. На этом веб-ресурсе (рис. 3.10) предлагается огромный и постоянно пополняющийся выбор бесплатных шрифтов для скачивания — как отдельных шрифтовых файлов, так и коллекций (сборников) шрифтов. На портале открыт форум, организована почтовая рассылка новостей. Также портал предоставляет возможность заказа диска со шрифтами по почте.

☑ **Ресурс FontRiver.com** [13] представляет собой огромную коллекцию шрифтов (рис. 3.11), которые можно бесплатно скачать и использовать (если лицензия позволяет это сделать) на персональном компьютере. Для удобства поиска все шрифты разбиты на множество категорий: основные, декоративные, необычные, праздничные, и т. п. Также возможно сформировать список шрифтов по названию (точнее, по первой букве в их названии). В базе FontRiver.com более 11 000 шрифтов.

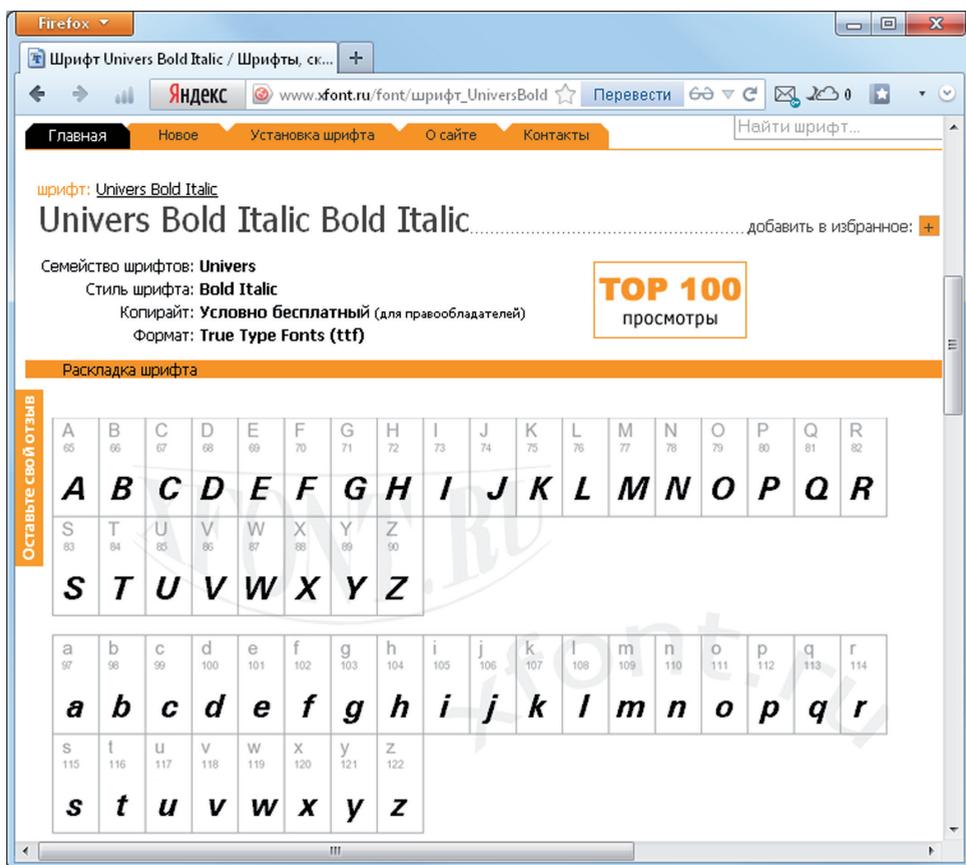


Рис. 3.9. Открытая «личная карточка» шрифта Univers на сайте XFont.RU

Отдельного внимания заслуживают три онлайн-сервиса. Это **WhatTheFont** [14] — веб-сервис, позволяющий определять шрифт по образцу (рис. 3.12), и **Identifont** [15] — веб-сервис, осуществляющий поиск начертания шрифта путем перебора характерных признаков (рис. 3.13) и **Google Web Fonts** [16] — онлайн-сервис от компании Google, предоставляющий услуги по поддержке разных форматов шрифтов на сайте пользователя всеми браузерами:

☑ **Сервис «WhatTheFont»** предоставляет пользователю уникальную услугу определения имени шрифта по его изображению (полученного, например, при сканировании текста). Загрузка пользователем образца шрифта в виде графического файла (размер которого не должен превышать 2 МБ) производится через веб-интерфейс по адресу или URL-ссылке (рис. 3.12).

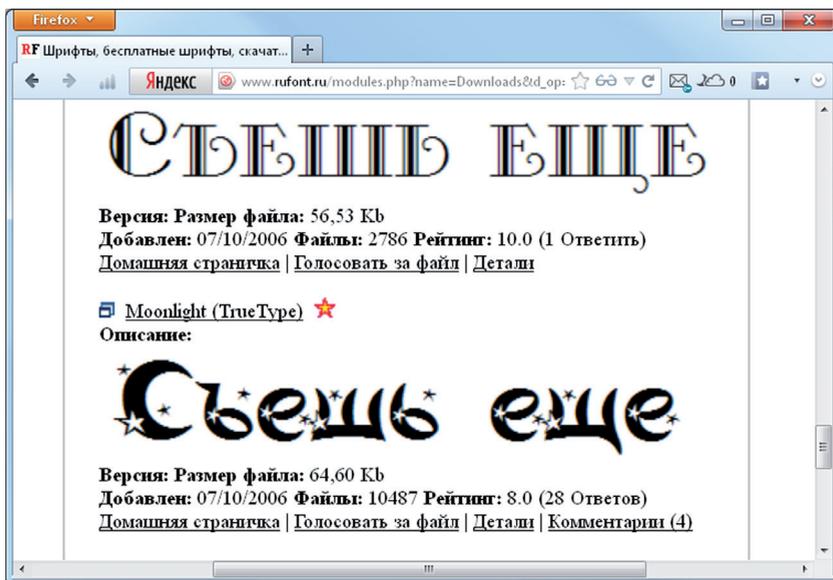


Рис. 3.10. Каталог шрифтов на портале «RuFont»

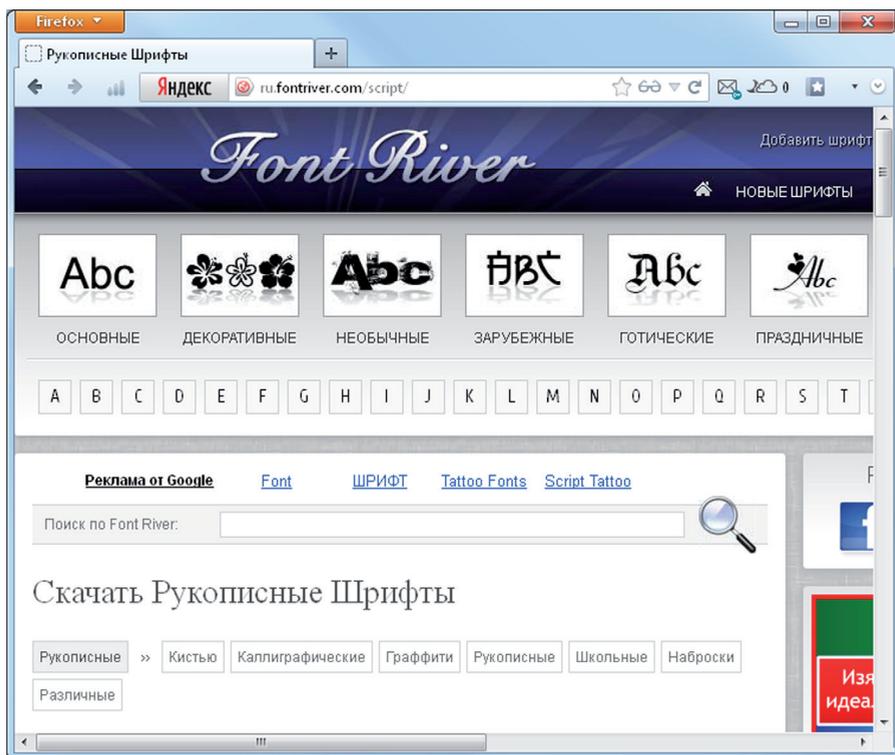


Рис. 3.11. Выбор каталога рукописных шрифтов на сайте FontRiver.com

Полученное изображение сравнивается с имеющимися в базе данных образцами. В случае успешного определения пользователю предоставляются данные об искомом шрифте с указанием имени, начертания и ссылки на файл распознанного шрифта для загрузки и установки на персональном компьютере.

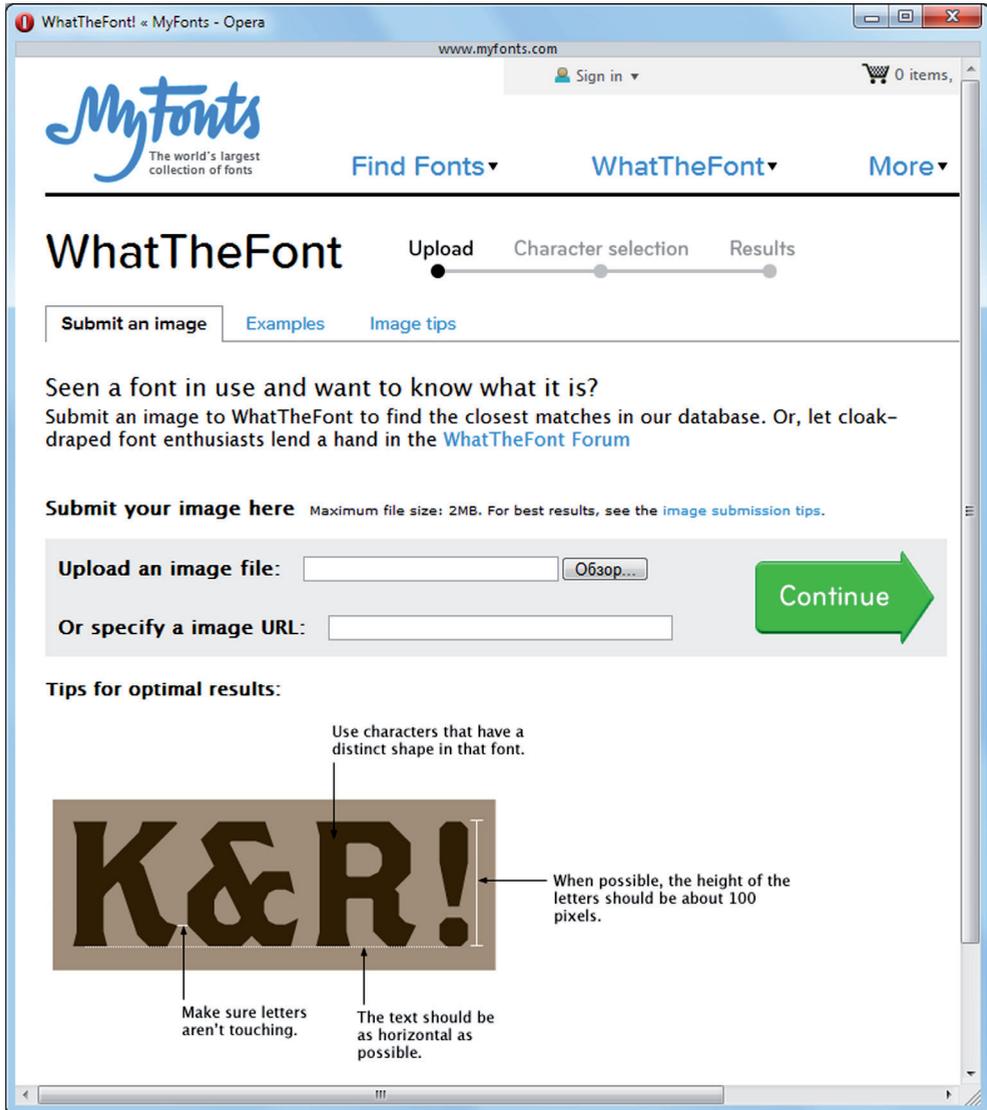


Рис. 3.12. Веб-сервис WhatTheFont

☑ **Сервис «Identifont»** (запущен в ноябре 2000 года) является крупнейшим независимым каталогом цифровых шрифтов в Интернете. Он предоставляет набор услуг для облегчения поиска шрифтов или информации о них. В этот набор входят следующие функции:

- поиск шрифта по внешнему виду;
- поиск шрифта по названию и сходству;
- поиск издателей и дизайнеров шрифтов.

Каждая функция снабжена подробным описанием и пошаговыми инструкциями (рис. 3.13).

☑ **Сервис Google Web Fonts** прост в использовании и не только предоставляет коллекцию разных шрифтов, но и генерирует код подключения выбранного шрифта для встраивания на сайт пользователя (при этом сам шрифт физически находится на сервере Google Web Fonts). Предоставляемые сервисом шрифты подобраны очень качественно и свободны для использования. При встраивании шрифта на сайт пользователя, сервис определяет версию браузера и обеспечивает автоматическое встраивание нужного формата шрифта.

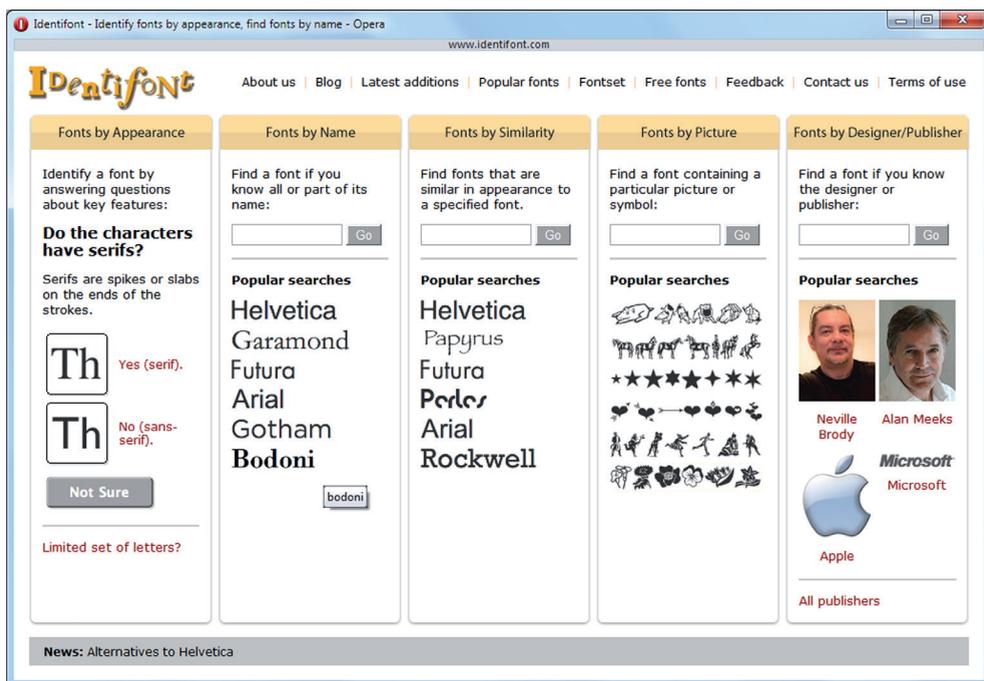


Рис. 3.13. Веб-сервис Identifont

Подключение шрифта сервиса Google Web Fonts происходит в несколько этапов. Порядок подключения и использования шрифта для веб-страницы производится в следующей последовательности:

1. На сайте сервиса (рис. 3.14) выбирается необходимый шрифт и добавляется в коллекцию (кнопка «Add to Collection»).

2. По нажатию кнопки «Use» в информационной строке коллекции (в нижней части окна) открывается новая страница, в которой пользователю предлагается выставить ряд настроек.

3. Использование предоставленного кода (рис. 3.15) у себя на сайте.

На рис. 3.16 показан результат подключения шрифта Bad Script, который предоставлен сервисом Google Web Fonts и подключен с помощью правила CSS @import (на клиентском компьютере этот шрифт не установлен).

К недостаткам сервиса Google Web Fonts можно отнести следующее:

- при первой загрузке страницы текст может исчезнуть, а появиться только тогда, когда загрузка шрифта из сети доставки

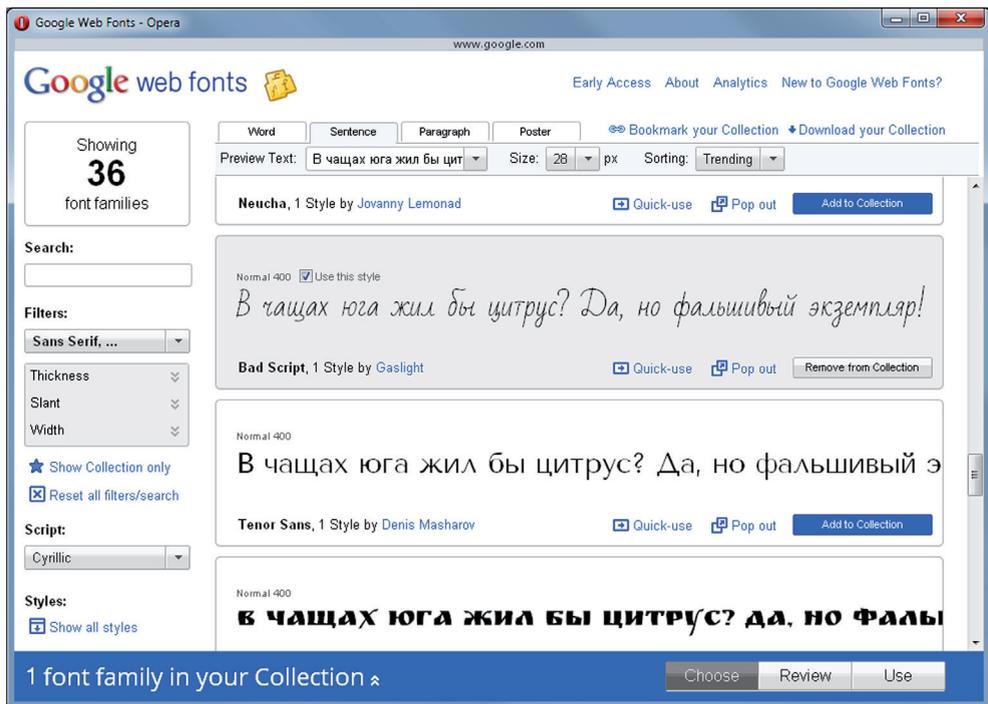


Рис. 3.14. Сервис Google Web Fonts с выбранным и занесенным в коллекцию кириллическим шрифтом Bad Script

и дистрибуции контента (англ. Content Delivery Network, CDN) от Google будет завершена;

- малый объем коллекции кириллических веб-шрифтов;
- зависимость сайта от Google: если вдруг шрифт удалится из коллекции или сервис по каким-либо причинам станет временно недоступным, то на сайте также исчезнет задуманное отображение текста.

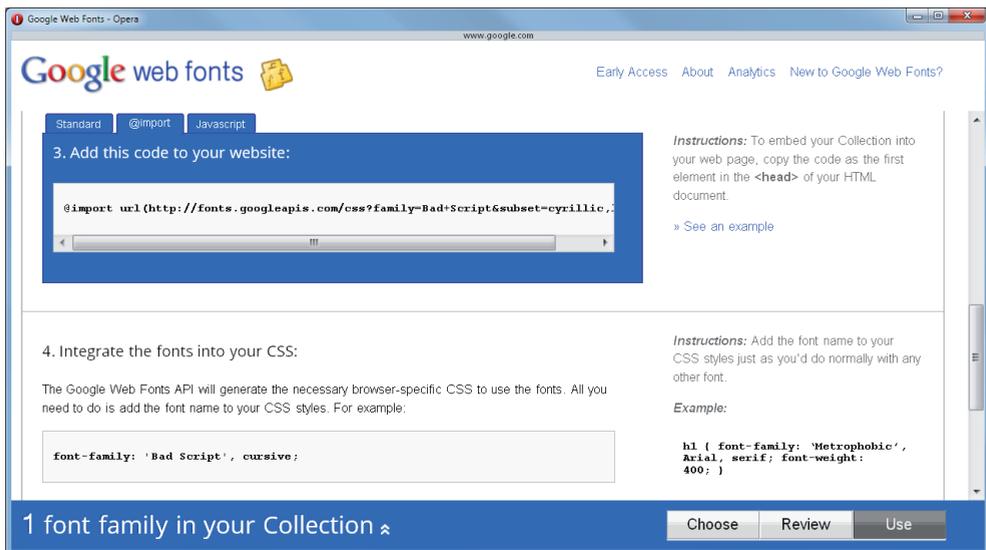


Рис. 3.15. Сервис Google Web Fonts со сгенерированными строками кода для подключения выбранного шрифта

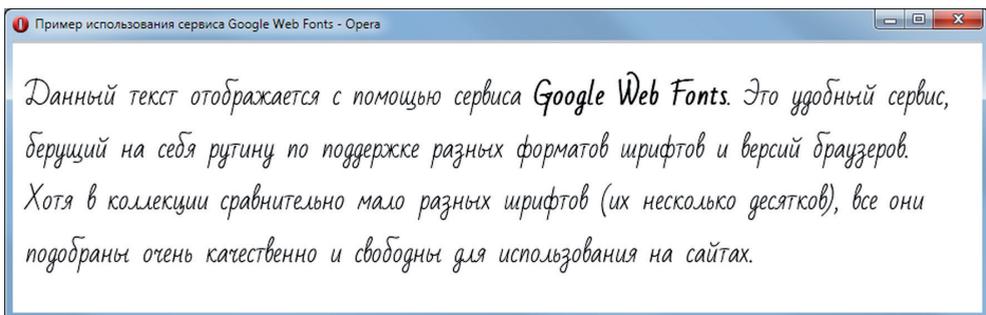


Рис. 3.16. Пример использования сервиса Google Web Fonts (шрифт Bad Script)

3.1.5. Что нужно знать о цвете

Восприятие цветов, которые человек видит вокруг себя, вызывается действием на глаза сложного цветового потока, состоящего из световых волн различной длины. Глаз человека — это также мощный инструмент для «улавливания» оттенков одного цвета. На основании возникающего физиологического зрительного ощущения и ряда физических и психологических факторов цвет оказывает влияние (как подсознательное, так и сознательное) на эмоциональный и психофизический фон человека. Поэтому цвет является одним из самых значимых элементов любого дизайна.

Каждый цвет характеризуется четырьмя основными качествами: тоном, насыщенностью, оптической плотностью и температурой:

- **Тон** — идентификация цвета, определяемая характером распределения излучения в спектре видимого света. Именно тон определяет название цвета: например, «красный», «синий», «зеленый» и др. Термин «тон» может использоваться и для обозначения других цветовых характеристик: например, фразы «светлый тон» или «темный тон» выражают степень насыщенности цвета.
- **Насыщенность** — интенсивность определенного цветового оттенка. Насыщенностью определяется яркость или тусклость цвета. В полиграфии насыщенность оценивается по равноконтрастным цветовым таблицам, например, атласу цветов.
- **Оптическая плотность** — аспект, который является мерой непрозрачности слоя вещества для световых лучей. Фактически оптическая плотность выражает процент содержания черного или белого в цвете.
- **Температура цвета** является субъективным, эмпирическим качеством и является основой объективности впечатления от цвета. Единицами измерений в цветовой температуре выступают градусы Кельвина (К). Шкалу температуры цвета (рис. 3.17) можно условно разделить на категории «теплых», «нейтральных» и «холодных» цветовых тонов.

Восприятие тона является фактически абсолютным. Синий цвет видится именно синим, красный — красным. Но если разместить рядом

два похожих оттенка одного цвета, то они будут восприниматься по-разному. Например, из двух оттенков синего цвета один будет восприниматься «более красным», а другой — «более зеленым» по отношению к другому.

Ощущение оптической плотности, насыщенности и температуры цвета также может меняться в контексте сочетания с другими цветами. Например, фиолетовый цвет может казаться ярким на фоне цвета светлого тона (например, желтого), но тусклым на фоне цвета темного тона (например, коричневого).

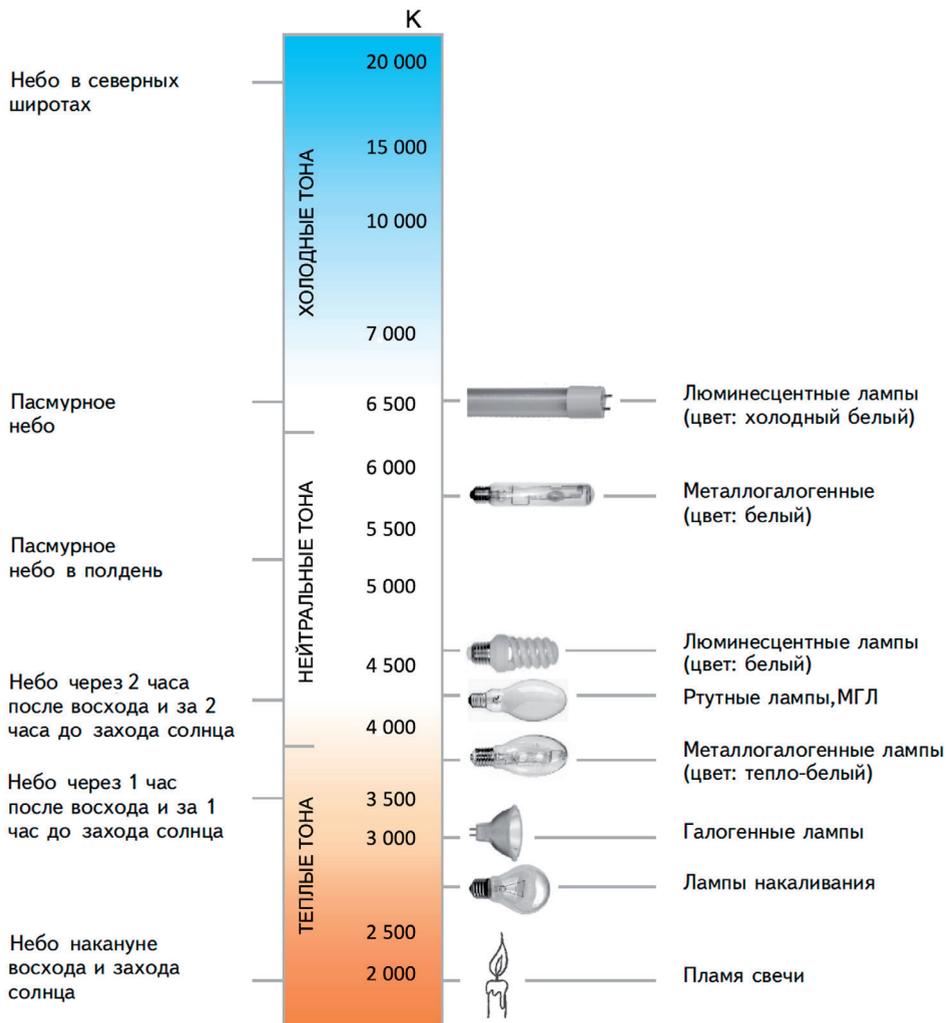


Рис. 3.17. Шкала цветовой температуры в Кельвинах

По сути цвет является событием, происходящим при участии трех составляющих: **источника света, объекта и наблюдателя**. Цветовое событие порождается способностью наблюдателя воспринимать световые волны, излучаемые источником света и видоизменяемые отражающим свет объектом. Если любая из трех составляющих цветового события изменяется, это событие становится другим — иными словами, наблюдатель увидит другой цвет.

Цвета, которые могут быть получены световыми волнами одной длины (или очень узким диапазоном), называются **спектральными цветами**. Ниже перечислены основные спектральные цвета (в скобках указаны диапазоны длин волн в нанометрах) [17]:

- фиолетовый (380–440 нм);
- синий (440–485 нм);
- голубой (485–500 нм);
- зеленый (500–565 нм);
- желтый (565–590 нм);
- оранжевый (590–625 нм);
- красный (625–740 нм).

Наблюдаемые в природе цвета разделяются на две группы:

1. **Хроматические цвета** — это те цвета и их оттенки, которые человек различает в спектре белого света (от фиолетового цвета до красного).
2. **Ахроматические цвета** представляют группу, состоящую из белого, различных оттенков серого и черного цветов. Эти цвета характеризуются лишь количеством отраженного света, или, иначе говоря, отличаются по коэффициенту отражения. Человеческий глаз различает в гамме ахроматических цветов около 300 оттенков.

Черный цвет представляет либо наличие световых волн всех длин в субтрактивной цветовой модели, либо полное отсутствие цвета в аддитивной цветовой модели. Это самый «сильный» цвет видимого спектра, плотность и насыщенность которого доминируют над другими цветами. Белый цвет представляет либо наличие световых волн всех длин (аддитивная цветочная модель), либо полное отсутствие цвета (субтрактивная цветочная модель).

Цвета разной длины волны производят различный эффект на нервную систему. У более теплых цветов (желтый, оранжевый, красный)

длина волны больше, поэтому требуется больше усилий от человеческого глаза и мозга для ее обработки. Соответственно, повышение энергетического уровня и скорости процесса метаболизма приводит к общему ощущению подъема в организме. И наоборот, цвета с короткой длиной волны — зеленый, голубой, синий и фиолетовый — требуют гораздо меньше энергии для обработки, что приводит к замедлению процесса метаболизма и оказывает успокаивающее действие [18].

Ощущение цвета также зависит от комплекса физиологических, психологических и культурно-социальных факторов. Первоначально исследования восприятия цвета проводились в рамках цветоведения¹. Позже вопросами восприятия света человеком стали заниматься этнографы, социологи и психологи.

Одни и те же цвета могут иметь разные психологические характеристики, которые зависят от индивидуального и культурного опыта человека (наблюдателя). В то же время одни и те же эмоциональные ощущения могут ассоциироваться с различными цветами, что подчеркивает индивидуальность цветовых ассоциаций. Так, во многих культурах красный цвет приравнивается к ощущениям голода, гнева или энергии, а, например, у вегетарианцев с голодом ассоциируется зеленый цвет. В западной культуре, которая в основном христианская, черный цвет ассоциируется со смертью и трауром, а у индусов со смертью ассоциируется белый цвет, который христиане считают цветом чистоты и невинности. В большинстве культур синий цвет ассоциируется с водой или жизнью, а фиолетовый — с властью и роскошью [18, с. 13].

Ниже приведены **психологические характеристики часто встречающихся цветов** и их символические значения, которые нашли отражения в культуре и традициях разных народов [19–22]:

☑ **Серый цвет** — нейтральный, который может восприниматься как уклончивый, неопределенный, так и формальный, благородный. Символизирует некую отстраненность, поскольку не обладает эмоциональностью цветности. Серый цвет, особенно в серебристом варианте, ассоциируется с высокими технологиями. Но всегда выглядит раздвоенным, поскольку соединяет в себе противоположные качества черного и белого цветов.

¹ **Цветоведение** — анализ процесса восприятия и различения цвета на основе систематизированных сведений из физики, физиологии и психологии.

Люди, предпочитающие серый цвет, не верят, что эмоции могут что-то решить, не верят в искренность эмоциональных переживаний; считают, что эмоции можно проявлять только в определенных обстоятельствах (но только не сейчас). Отсюда их эмоциональная сдержанность и, следовательно, нервное истощение.

☑ **Синий цвет** успокаивает и создает ощущение защищенности и безопасности. У многих народов символизирует, прежде всего, небо и воду, реке — доброту, вечность, постоянство и расположение. В геральдике обозначает целомудрие, честность, добрую славу и верность. Вызывает не чувственные, а духовные впечатления. Синий цвет — это постоянство, упорство, настойчивость, преданность, самоотверженность, серьезность, строгость.

Кроме того, синий цвет близок к черному и получает сходные с ним символические значения. Он считался траурным в Древнем Египте и у некоторых народов Южной Африки. Французы называют ужас «синим страхом». У славянских народов синий цвет служил цветом печали, горя, ассоциировался с «бесовским» миром. Старинные предания описывают черных и синих бесов. Однако в мифах синий цвет — это божественное проявление, цвет загадочности и ценности.

Люди, предпочитающие этот цвет, стараются все привести в порядок, систематизировать. Они всегда имеют собственную точку зрения, преданы тому, что делают и их преданность людям может доходить до рабства. Согласно статистике, синий — наиболее любимый цвет у людей.

☑ **Зеленый цвет** — происходит от слияния синего и желтого, при этом взаимодополняются качества того и другого цвета. Является цветом природы (трава и листья), отсюда появляется покой и неподвижность. В зеленом цвете всегда заложена жизненная возможность, он содержит в себе потенциальную энергию — не покоится, а отражает внутреннее напряженное состояние. Выражает отношение человека к самому себе.

У многих народов зеленый цвет символизирует юность, надежду, веселье, хотя порой — и незрелость, недостаточное совершенство. Зеленый цвет предельно материален и действует успокаивающе, но может производить и угнетающее впечатление (не случайно тоску называют «зеленой», а сам человек «зеленеет» от злости). У иранцев зеленый цвет

ассоциируется как с бурным ростом и свежестью, так и с несчастьем, печалью, скорбью, поэтому о злополучном человеке говорят «зеленая нога», а о кладбище — «зеленый дом». В средневековой Европе шуты носили зеленую с желтым одежду, а банкроты в Германии должны были надевать зеленые шапки.

Люди, предпочитающие этот цвет, как правило, видят обе стороны ситуации, умеют взвешивать и оценивать шансы на благоприятный исход, подавлять своим авторитетом, отличаются высокой работоспособностью. Они помнят только то, что нужно, при этом они склонны помогать другим людям, даже в ущерб себе. Они приветливы, но скрытны, — у них есть свой собственный мир, который они никому не раскрывают.

Зеленый цвет нейтрализует действие остальных цветов, помогает рассеивать негативные эмоции. Приносит спокойствие и умиротворенность, помогает сконцентрироваться при принятии решения и обладает снотворным действием.

☑ **Красный цвет** является «живым» ярким цветом и считается наиболее заметным. Символизирует физиологическое состояние, связанное с расходом энергии. Вызывает чувство страсти или эмоционального подъема и прежде всего, ассоциируется с кровью и огнем. Его символические значения очень многообразны и, порой, противоречивы: с одной стороны, красное символизирует радость, красоту, любовь и полноту жизни, а с другой — вражду, месть, войну. Красный цвет выражает жизненную силу, нервную активность, стремление к успеху, желания всех жизненных благ. У многих народов красный цвет символизирует страсть, воинственность, потерю самообладания. В Древней Африке в красное одевали царей и вождей, красный цвет связывали с жарким временем года. У крестоносцев красный цвет в геральдике выражал ярость и жестокость. В христианстве красный цвет — символ великолепия, пролитой крови. В древних племенах тело умершего человека окрашивали в красный цвет, указывая на активность и после смерти.

Красный цвет способствует работе желез внутренней секреции, выработке адреналина, связан с органами размножения, влияет на повышение температуры тела.

☑ **Желтый цвет** вызывает ощущение счастья и радости, поскольку ассоциируется с солнцем и теплом. Это также цвет осени, цвет зрелых

колосьев и увядающих листьев, но также и цвет болезни, смерти, потустороннего мира.

В мифологии желтый цвет олицетворяет собой Солнце, тепло, весну и цветы. У многих народов женщины отдавали предпочтение желтой одежде. Нередко желтый цвет служил отличительным признаком знатных особ и высших сословий. Например, монгольские ламы носят желтую одежду с красным поясом. С другой стороны, у некоторых народов Азии желтый цвет является цветом траура, скорби, печали. В Европе желтый или желто-черный флаг обозначал карантин, а желтый крест — чуму. У славянских народов желтый цвет считается цветом ревности, измены, а на Тибете ревность называют буквально «желтый глаз».

☑ **Оранжевый цвет** — цвет теплоты, блаженства, накала, но в то же время — мягкого блеска заходящего солнца. Он всегда радует глаз и способствует хорошему настроению. Практически всегда имеет благотворное влияние, т. к. показывает радостные стороны жизни (в отличие от синего).

Этот цвет отвечает за удовлетворение различных способностей, постоянно держит в тонусе и обладает всеми возможностями красного, но без агрессии. Воздействие, оказываемое им, является теплым, радостным и возбуждающим. Этот цвет связан со стремлением к достижению самоутверждения.

Оранжевый цвет дает способность что-то делать и считается одним из лучших цветов в психотерапии.

☑ **Коричневый цвет** соединяет в себе желтый и красный цвета. В нем остается жизненность, которая потеряла свою активность. Выражает жизненные ощущения тела, ассоциируется с землей и деревом и приносит ощущение комфорта и безопасности. Коричневый цвет олицетворяет стабильность, преданность (например, при приеме на работу оказывает положительное воздействие). Успокаивает, поддерживает во время тревоги, волнений. Выбирается при нервном истощении, когда человеку кажется, что ситуация конфликта неразрешима.

Люди, предпочитающие этот цвет, желают физического отдыха, покоя. Игнорируется состояние здоровья при отвержении коричневого цвета.

☑ **Черный цвет** у многих народов, как правило, символизирует несчастье, горе, траур, гибель. Так, в Древней Мексике при ритуальном

жертвоприношении человека лицо и руки у жрецов были окрашены в черный цвет. Черные глаза и поныне считаются опасными, завистливыми. В черное одеты зловещие персонажи, появление которых предвещает смерть.

Однако черный цвет имеет и благоприятные значения. Например, в засушливых районах Африки черные тучи сулят плодородие и изобилие. Духам-хранителям, посылающим дождь, приносят в жертву черных быков, коз или птиц, а жрецы при этом тоже облачаются в черные одеяния. Считается также, что существует связь между черным цветом и сексуальной привлекательностью. У некоторых африканских племен женщины с очень черной кожей высоко ценятся как любовницы (но не как жены). Черное может символизировать нечто сокровенное и желанное (например, «покрытую темнотой и тайной» любовную страсть). У арабов выражение «чернота глаз» означает возлюбленную, «чернота сердца» — любовь.

Черный цвет дает шанс отдохнуть нервной системе, содержит в себе обещание, надежду, он притягивает, поглощает в себя, но не призывает к действию.

☑ **Белый цвет** символизирует чистоту, незапятнанность, невинность, добродетель, радость. Он ассоциируется с дневным светом, а также с производящей силой, которая воплощена в молоке и яйце. Белый цвет характеризуется совершенством и завершенностью, демонстрирует абсолютное и окончательное решение, полную свободу для возможностей и преодоления препятствий.

Еще с античности белый цвет имел значение отрешенности от мирского, устремления к духовной простоте. Он очень широко используется в христианстве, как символ веры, чистоты, истинности и светлости. В белом одеянии изображаются ангелы, святые и праведники. У некоторых народов белую одежду носили цари и жрецы, что символизировало торжественность и величие.

Однако белый цвет может получать и противоположное значение. По своей природе он как бы поглощает, нейтрализует все остальные цвета и соотносится с пустотой, бестелесностью, ледяным молчанием и в конечном итоге — со смертью.

Славяне одевали умерших в белую одежду и покрывали белым саваном. У некоторых племен Африки и Австралии принято раскраши-

вать тело белой краской после кончины кого-нибудь из близких. В Китае и в некоторых других странах Азии и Африки белый цвет является цветом траура. В старину белый траур использовался и у славян.

В мифологии белый цвет часто используется в Африке — его носили жрецы, т. к. белый цвет оберегал и давал возможность зачаровывать. Белый цвет отражал социальное согласие и мир. Если принести в жертву белое животное — это пакт перемирия с богами. Если выкрасить дом внутри белым — в нем будет мир, если выкрасить только косяки дверей — человек, входя, оставит свое зло снаружи.

* * * * *

При подборе (сочетании) определенных цветов проявляются новые «психологические способности» цвета, задающие также положительный эмоциональный фон в случае гармонирующих цветов, или наоборот, оказывающие негативное влияние в случае отсутствия гармонии (дисгармонирующие цвета).

Дисгармонирующие цвета вызывают негативные эмоции у потребителя. Например, соединение фиолетового с оранжевым говорит о безысходности, суициде. Тот же эффект вызывает надпись, написанная, например, белыми буквами на сплошном черном фоне, хотя черный и белый цвета вместе гасят друг друга (т. е., по сути являются сочетаемыми), но уже не несут своей первоначальной информации, а оказывают давление на психику.

3.1.6. Цветовые сочетания

Влияние цвета на человека не ограничивается психолого-социальными и этническими аспектами. Цвет имеет еще и пространственные качества, которые влияют на эмоциональный фон человека, на его восприятие. Например, синий цвет кажется удаленным, желтый — приближенным, а красный не изменяет своего среднего положения в пространстве. Также, чем меньше цвета, тем более темным он кажется. Например, большая и малая фигуры одного цвета, расположенные на белом фоне будут казаться разными по оптической плотности: цвет большой фигуры будет казаться светлее, потому что малая фигура окружена большим пространством яркого белого цвета, и на его фоне выглядит контрастнее.

Используя в веб-дизайне подобные качества цвета, можно добиться определенных эффектов, которые выведут веб-страницу на новый композиционный уровень, а также сделают ее более привлекательной и запоминающейся.

Чтобы правильно подбирать гармоничные цвета для сайта и понимать принципы работы с цветом, необходимо иметь представление о **цветовом круге**, который широко используется в дизайне и программах для работы с графикой. Существует масса вариаций цветового круга, среди которых наиболее удачным и логически понятным (по мнению авторов) видится **12-секторный цветовой круг Иоханнеса Иттена** (рис. 3.18), созданный всемирно известным швейцарским художником в 1961 году [23].

На рис. 3.19 показана взаимосвязь цветовых секторов круга с цветовым шестиугольником в центре. В центре круга — вписанный в треугольник **три основных цвета**: желтый, красный и синий. Их еще называют первичными или чистыми, потому что их нельзя получить при смешивании других цветов. Эти цвета занимают 1-й, 4-й и 8-й сектора соответственно.



Рис. 3.18. Цветовой круг Иоханнеса Иттена

К сторонам треугольника основных цветов примыкают треугольники с **переходными вторичными** (составными, второго порядка) цветами (оранжевый, фиолетовый, зеленый), образуя правильный шестиугольник. Цвета этих треугольников получаются смешением (наложением) пары основных цветов — тех, которые примыкают к основанию:

- желтый + красный = оранжевый (2-й сектор);
- красный + синий = фиолетовый (6-й сектор);
- синий + желтый = зеленый (10-й сектор).

Третья группа цветов — **промежуточные третичные** (третьего порядка) цвета, которые получаются смешением близких по расположению в шестиграннике пар цветов первого и второго порядка:

- желтый + оранжевый = желто-оранжевый (1-й сектор);
- желтый + зеленый = желто-зеленый (11-й сектор);
- красный + фиолетовый = красно-фиолетовый (5-й сектор);
- красный + оранжевый = красно-оранжевый (3-й сектор);
- синий + зеленый = сине-зеленый (9-й сектор);
- синий + фиолетовый = сине-фиолетовый (7-й сектор).

Незамысловатые названия третичных цветов получены как составные от образующих их пар цветов, причем сначала ставится название основного цвета, затем — вторичного.

Итак, 12-секторный цветовой круг содержит 3 цвета первого порядка (первичных), 3 цвета второго порядка (вторичных) и 6 цветов третьего порядка (третичных) (рис. 3.20).

Кроме этого, цветовой круг делится на две половины, одна из которых — теплые цвета (11, 12, 1–4 сектора), другая — холодные цвета (5–10 сектора).

Впечатление, созданное взаимодействием двух и более цветов, называется **цветовой гармонией**, если цветовые сочетания лишены сильного контраста и построены на сочетании различных цветов, близких по светлоте, либо на сочетании цветов, близких по своему характеру. Цветовая гармония является весьма субъективным понятием, а оценка гармоничного сочетания цветов может быть построена по принципу «нравится – не нравится» для каждого отдельно взятого наблюдателя.

Однако существует естественная потребность человеческого глаза достичь цветового равновесия. Если некоторое время смотреть на жел-

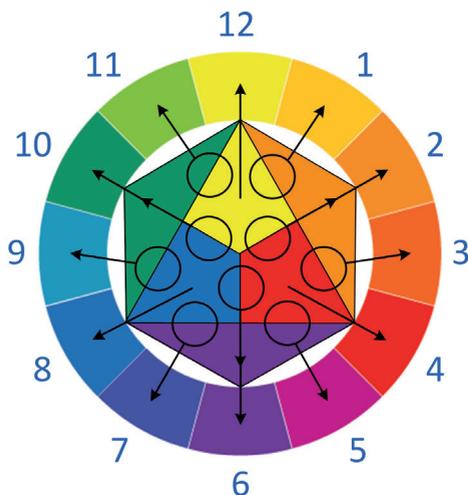


Рис. 3.19. Формирование цветов в секторах цветового круга



Рис. 3.20. Первичные, вторичные и третичные цвета цветового круга

тый квадрат, а потом закрыть глаза, то возникнет изображение фиолетового квадрата. И наоборот, закрыв глаза после наблюдения фиолетового квадрата, будет видаться желтый. Подобный опыт можно производить со всеми цветами, и всякий раз, закрывая глаза, можно увидеть цвет, дополнительный к тому, который только что рассматривался.

☑ **Дополнительные (комплементарные¹) цвета** в цветовом круге расположены строго в противоположных секторах. Например, цвета фиолетовый и желтый, синий и оранжевый, красный и зеленый являются комплементарными, т. е. дополняющими друг друга.

На рис. 3.21 показано формирование шести пар комплементарных цветов для 12-секторного цветового круга (каждая такая пара состоит из теплого и холодного цвета):

- 1) желтый || фиолетовый;
- 2) желто-оранжевый || сине-фиолетовый;
- 3) оранжевый || синий;
- 4) красно-оранжевый || сине-зеленый;
- 5) красный || зеленый;
- 6) красно-фиолетовый || желто-зеленый.

¹ От англ. *complementary* — дополнительный, дополняющий.

При смешении комплементарные цвета поглощают (убирают) друг друга и их смесь представляет собой нейтральный серый цвет. Такие цвета обладают **полярной контрастностью**¹ и гармонируют² друг с другом так, что, помещенные рядом, они усиливают друг друга, делают друг друга ярче и «живее». Поэтому комплементарные цвета — идеальное контрастное сочетание цветов заголовков и фона. Для выделения основных текстовых блоков такой прием не целесообразен, поскольку текст будет «пестрить» и «выпирать» по отношению к фону, что быстро утомит читателя.

Комплементарные цвета являются одной из классических комбинаций (схем) для подбора контрастных гармоничных сочетаний двух цветов. Существует несколько базовых сочетаний (цветовых схем), использующих три и более гармонирующих цвета:

☑ **Аналоговая триада** образуется тремя соседними цветами в цветовом круге (рис. 3.22). Обратные комплементарным, **соседствующие цвета**, располагающиеся на цветовом круге в соседних секторах, очень хорошо смотрятся вместе, но не образуют никакого контраста. Используются для создания ощущения умиротворения и комфорта. Позволяют создавать плавные «перетекания» и градиентные переходы между любыми, сколь угодно удаленными друг от друга цветами, используя цепочку аналоговых триад. На рис. 3.22 приведены примеры четырех (из 12 возможных) аналоговых триад:

- 1) желтый || желто-зеленый || зеленый;
- 2) сине-фиолетовый || фиолетовый || красно-фиолетовый;
- 3) красно-оранжевый || красный || красно-фиолетовый;
- 4) сине-зеленый || зеленый || желто-зеленый.

☑ **Классическая триада** образуется тремя равноудаленными по цветовому кругу цветами. На цветовом круге цвета классической триады являются вершинами равностороннего треугольника (рис. 3.23). Композиция цветов классической триады выглядит достаточно «жи-

¹ Сравнимые между собой цвета, которые обладают четко выраженными различиями (эти различия формируют оптическая плотность, насыщенность и температура сравниваемых цветов), называют контрастными. При достижении предела в этих различиях, цвета становятся диаметрально или полярно контрастными.

² Два или более цвета являются гармоничными, если их смесь представляет собой нейтральный серый цвет.

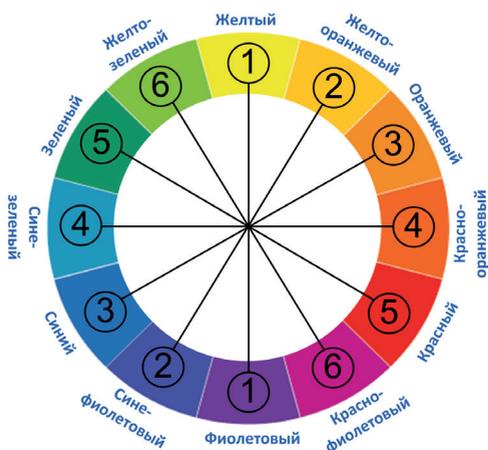


Рис. 3.21. Формирование шести пар комплементарных цветов

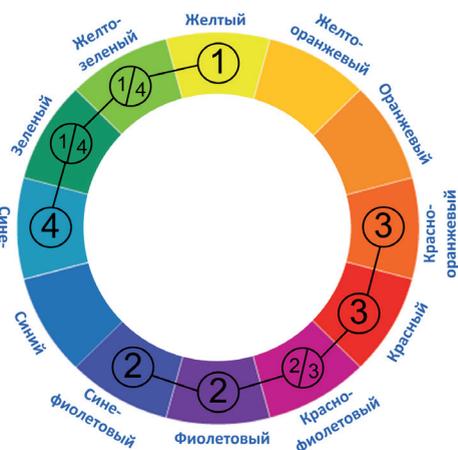


Рис. 3.22. Примеры четырех аналоговых триад на цветовом круге

вой» даже при использовании бледных и ненасыщенных цветов. Для обеспечения гармоничности при использовании классической триады, один цвет принимается как доминирующий (главный), а два других — как акцентные (вспомогательные).

На цветовом круге Иоханнеса Иттена возможно получить четыре варианта цветов классической триады:

- 1) желтый || красный || синий;
- 2) желто-зеленый || красно-оранжевый || сине-фиолетовый;
- 3) зеленый || оранжевый || фиолетовый;
- 4) сине-зеленый || желто-оранжевый || красно-фиолетовый.

☑ **Контрастная триада** — вариант комплементарного сочетания цветов, только вместо противоположного цвета используются соседние для него цвета (рис. 3.24). Расположение цветов контрастной триады на цветовом круге можно представить вершинами равностороннего треугольника. При этом цвет вершины равностороннего треугольника принимается как основной, а цвета основания — как вспомогательные.

На рис. 3.24 приведены примеры цветов двух (из 12 возможных) контрастных триад:

- 1) желтый || красно-фиолетовый || сине-фиолетовый;
- 2) желто-оранжевый || синий || фиолетовый.

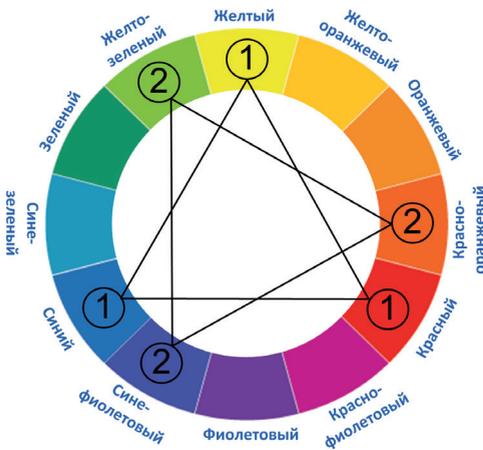


Рис. 3.23. Примеры двух классических триад на цветовом круге

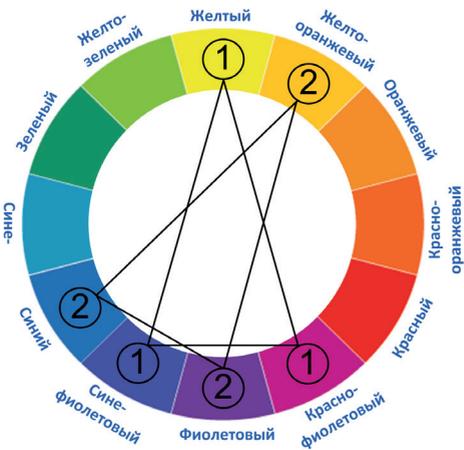


Рис. 3.24. Примеры двух контрастных триад на цветовом круге

☑ **Прямоугольная схема** состоит из четырех цветов, которые на цветовом круге являются вершинами прямоугольника. При этом цвета противоположных вершин прямоугольника являются комплементарными. Чтобы проще было сбалансировать прямоугольную схему, один цвет надо выбрать доминирующим, остальные — вспомогательными. На рис. 3.25 приведены примеры цветов двух (из 6 возможных) прямоугольных схем 12-секторного цветового круга:

- 1) желтый || красный || фиолетовый || зеленый;
- 2) желто-зеленый || сине-зеленый || красно-фиолетовый || красно-оранжевый.

☑ **Квадратная схема** практически повторяет прямоугольную схему, но цвета в ней равноудалены по кругу (рис. 3.26). В квадратной схеме также целесообразно выбрать один доминирующий цвет.

На рис. 3.26 приведены примеры трех квадратных схем сочетания цветов для цветового круга Иоханнеса Иттена:

- 1) желтый || красно-оранжевый || фиолетовый || сине-зеленый;
- 2) зеленый || желто-оранжевый || красный || сине-фиолетовый;
- 3) оранжевый || красно-фиолетовый || синий || желто-зеленый.

Использование вышеприведенных схем сочетания позволяет с легкостью подобрать гармоничные цвета. Следует заметить, что классические цветовые схемы всегда можно подкорректировать под конкретную

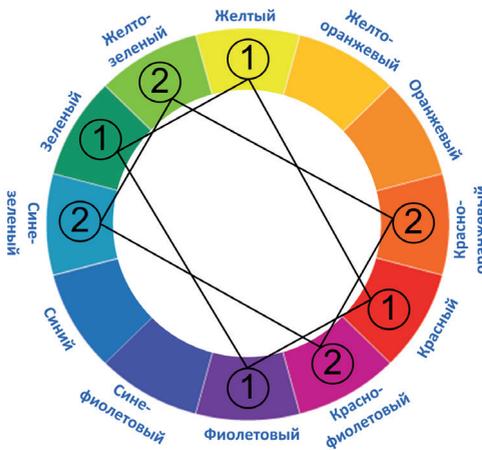


Рис. 3.25. Примеры двух прямоугольных схем на цветовом круге

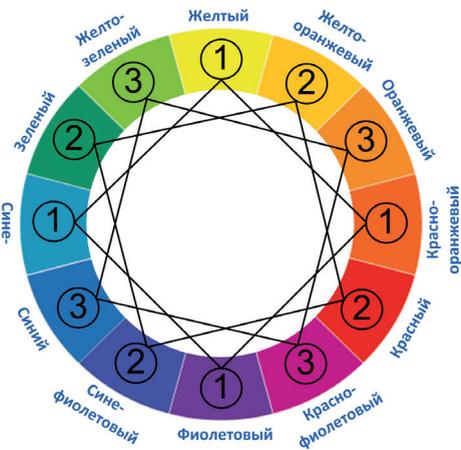


Рис. 3.26. Формирование трех квадратных схем на цветовом круге

область применения. Для достижения оптимального результата возможна «подводка» насыщенности или яркости цветов так, чтобы отыскать «золотую середину» в их сочетании.

Выбор цветовой схемы зависит от того, сколько цветов необходимо для работы, например, для создания сайта. На этот вопрос нет однозначного ответа, но в целом можно предостеречь веб-дизайнеров от чрезмерного злоупотребления цветами: **риск использования большого количества цветов больше, чем риск использования их в слишком малом количестве.** Использование большого количества цветов делает веб-страницу более загруженной и, как правило, ухудшает восприятие размещенной на ней информации. К тому же такая веб-страница в большей степени утомляет глаза посетителей. Однако использование слишком малого количества цветов может привести к тому, что сайт в целом будет выглядеть скучно, и посетитель может перейти на сайты более привлекательного и «веселого» дизайна!

Таким образом, для сайта оптимальным является сочетание **трех-четырёх цветов.** На этот принцип и сориентированы схемы сочетания цветов по цветовому кругу.

Для подбора сочетаемых цветов также можно воспользоваться готовыми таблицами, которые бывают двух типов: описательные и графические. Описательные таблицы оперируют названиями цветов. В них

собраны основные цвета и списки сочетающихся (гармонирующих) и несочетающихся с ними цветов (например, таблица 3.1). Графические таблицы, как правило, составляются аналогично описательным, но с демонстрацией самих цветов (могут также указываться цветовые коды HTML).

Таблица 3.1. Сочетания цветов

Основной цвет	Цвета, сочетающиеся с основным цветом	Цвета, несочетающиеся с основным цветом
красный	зеленый, синий, ультрамарин, золотисто-желтый, серый, цвет резеды	фиолетовый, кирпичный, оранжевый, оливковый, розовый, коричневый, каштановый
розовый	бордо, коричневый, серый	синий, оливковый, красный, каштановый, ультрамарин, сиреневый
оранжевый	ультрамарин, зеленый, фиолетовый, лиловый, белый, коричневый	пурпурный, красный
коричневый	оливковый, миртовый, золотистый, беж, серый	бордо, сиреневый, розовый
золотистый	ультрамарин, зеленый, фиолетовый, коричневый, голубой	розовый, сиреневый
желтый	зеленый, коричневый	бордо, розовый
зеленый	фиолетовый, пурпурный	синий, оранжевый
голубой	красный, кирпичный, киноварный, ультрамарин, оранжевый, светло-фиолетовый	бордо, сиреневый
синий	оливковый, красный, золотистый, бордо	зеленый, розовый, коричневый
фиолетовый	золотистый, желтый, оранжевый	красный, кирпичный
серый	черный, ультрамарин, розовый, желтый, голубой, сиреневый	коричневый, бежевый

При выборе цветовой гаммы для сайта, важно сделать это правильно, руководствуясь основными принципами теории цвета. Рассмотренные наиболее значимые аспекты при выборе цветов, основные принципы их сочетания, роль в веб-дизайне и их психологическое значение позволяют эффективно использовать цвета при создании сайта и на порядок повысить привлекательность его содержимого.

Существует также большое количество онлайн-сервисов и приложений для подбора цветов и оптимальных цветовых схем. Некоторые из них подробно рассмотрены в следующем разделе.

3.1.7. Инструментарий веб-дизайнера для работы с цветом

Понимание теории цвета и грамотное использование ее на практике — это один из важнейших навыков хорошего веб-дизайнера. Однако, владея только теоретической базой, профессиональная работа с цветом практически невозможна (или достаточно трудоемка) без специализированных инструментов — **программ и сервисов для подбора цвета**.

Представленные далее инструменты, несомненно, укажут верное направление к выбору оптимальной цветовой палитры сайта. В Интернете можно найти и другие полезные программы для выбора цвета, но данные программы, во-первых, бесплатны, во-вторых, просты в использовании и, в третьих, в совокупности обладают исчерпывающим функциональным набором.

К основным функциям программ для подбора цвета можно отнести:

1. Получение образца именованного цвета и его кода.
2. Получение кода цвета по установленным координатам на цветовых моделях (или цветовых палитрах).
3. «Захват экрана» — получение образца, координат цветовой модели и кода цвета экрана «под курсором мыши».
4. Просмотр образцов, подбор и получение кодов гармоничных цветовых сочетаний в результате работы с различного рода цветовыми схемами (например, с цветовым кругом).

Отдельного внимания заслуживают онлайн-сервисы для работы с цветом, удобство общения с которыми очевидно: они не требуют установки (инсталляции) в операционную систему, поэтому они доступны с любого, подключенного к сети Интернет, компьютера и запускаются в браузере как обычные сайты. Многие онлайн-сервисы позволяют сохранять выбранные цветовые сочетания, а также получать доступ к цветовым сочетаниям, созданными другими пользователями.

Ниже представлены несколько инструментов веб-разработчика для работы с цветом: онлайн-сервисы и бесплатная программа для операционной системы Windows.

☑ **Яндекс. Цвета** — полезный инструмент для дизайнеров и верстальщиков, предоставленный поисковой системой Яндекс [24]. Он показывает, что из себя представляет цвет, указанный в запросе (на рис. 3.27 сервис вызван по ключевому слову «цвет» в строке поиска). При этом не важно, это название цвета (например, «грушевый цвет», «yellow») или его HEX-код (например, «#94FCA8»), сервис отобразит образец цвета, его название и значения в координатах цветовых моделей RGB и HSV. «Яндекс. Цвета» отображает цвета даже с такими необычными названиями, как, например, «фуксия», «влюбленная жаба» или «бедрa испуганной нимфы».

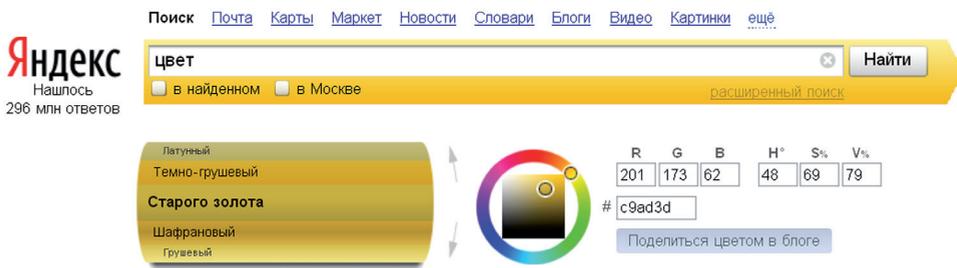


Рис. 3.27. Онлайн-сервис «Яндекс. Цвета»

В центре динамического списка отображается выбранный (или искомый) цвет и соседствующие с ним цвета. В цветовой палитре можно, изменяя цветовые каналы (в цветовом кольце) и насыщенность (в градиентном квадрате), получать коды и названия выбранных цветов.

Конечно же, трактовка некоторых цветов является очень и очень спорной, но, скорее всего, разработчики Яндекса подошли к данному проекту с определенной долей юмора и порадовали отличным сервисом как дизайнеров и веб-разработчиков, так и обычных людей.

☑ **Adobe Kuler** [25] — онлайн-сервис, цветовой микшер (генератор цветовых схем) известной компании Adobe, который на сегодняшний день является лучшим помощником профессионального веб-дизайнера (рис. 3.28). Adobe Kuler направлен на создание, редактирование и хранение цветовых схем и наборов цветов в соответствии с их гармоничным сочетанием.

Цвета в Kuler представляются в различных форматах, включая RGB, CMYK, LAB и HSV. В сервисе развернуто обширное комьюнити,

поэтому, пройдя регистрацию на сайте, можно делиться созданными палитрами или использовать и модифицировать под свои нужды чужие цветовые схемы (на сайте ведется рейтинг цветовых решений). Kuler совместим со многими приложениями из Adobe Creative Suite. Например, поддерживается экспорт палитры в формат .ase, что дает возможность загрузить созданный (или выбранный) набор цветов в небезызвестный графический редактор Photoshop.

☑ **ColorScheme.Ru** [26] — веб-сервис, представляющий собой удобный инструмент для подбора цветов и генерации цветовых схем (рис. 3.29). Является русской версией англоязычного сервиса Color Scheme Designer [27], который в качестве идеального инструмента по работе с цветом можно порекомендовать начинающим веб-дизайнерам. Конструктор цветовых схем ColorScheme.Ru использует подбор цветов по различным вариантам их сочетания в рассмотренном ранее цветовом круге.

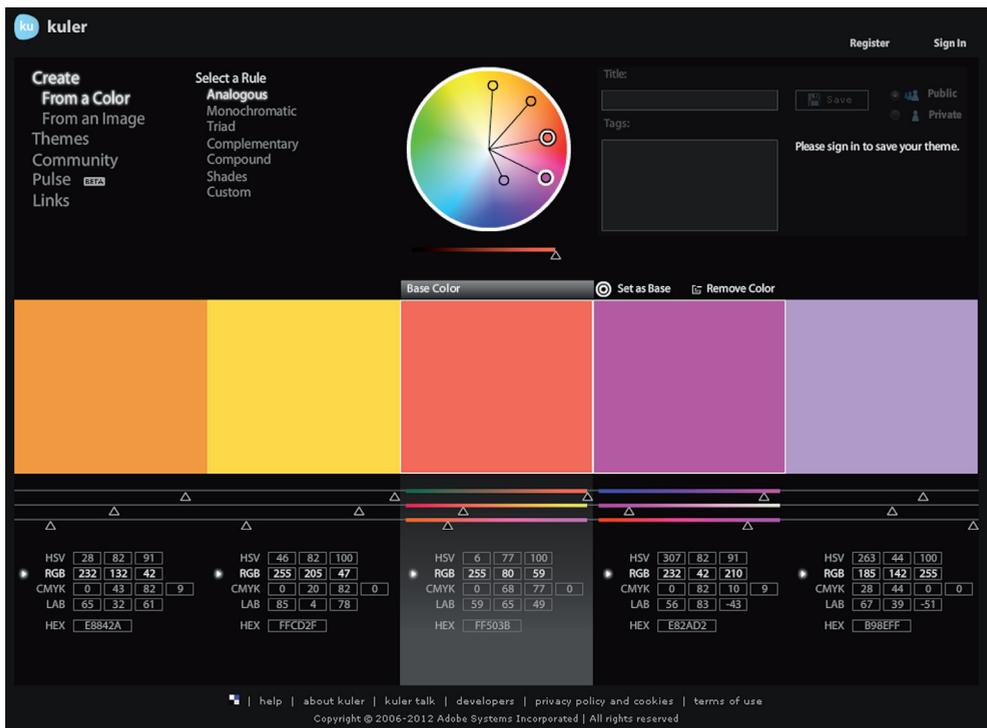


Рис. 3.28. Онлайн-сервис Adobe Kuler. Создание новой палитры

В виду популярности данного сервиса, стоит подробно рассмотреть некоторые аспекты создания собственной палитры цветов.

На первом этапе выбирается цветовая схема для будущей палитры (в левой верхней части имеются пиктограммы шести цветовых схем). На рис. 3.29 выбрана цветовая схема «акцент аналогия», задающая четыре цвета. В левой части размещена палитра «Цветовой круг». В зависимости от выбранной цветовой схемы, на палитре размещены от одного до четырех маркеров (темных для основных цветов, светлых — для дополнительных), определяющих координаты цветов. Перемещая маркеры с помощью мыши, можно подкорректировать цветовую схему или выбрать другие цветовые сочетания. Переместив один из маркеров, например, в область синего цвета, видим, что остальные маркеры «подтягиваются» к перемещенному не нарушая геометрии выбранной цветовой схемы.

Следующий шаг — изменение яркости и контраста (вкладка «Регулировка схемы»). Кроме ручной настройки есть предварительно подготовленные стили по типу «средний контраст», «темный сайт», «пастельные тона» и т. п.

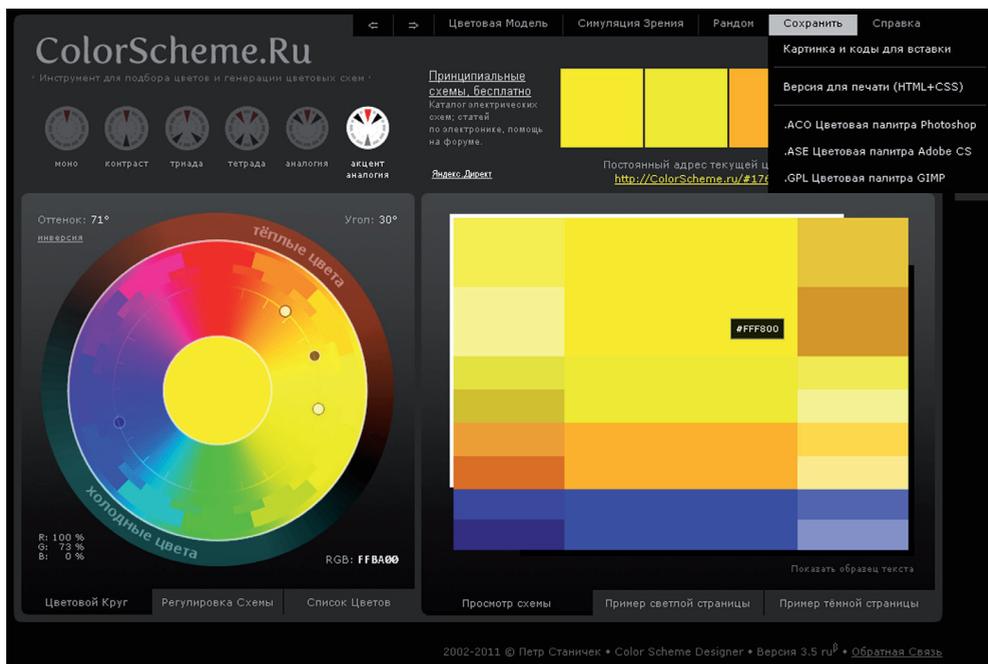


Рис. 3.29. Онлайн-сервис ColorScheme

Справа отображается результат работы с сервисом — пользовательская цветовая палитра (при наведении указателя мыши на цвет всплывает его код в HEX-формате). На цвета палитры можно добавить текст («Показать образцы текста»), а также просмотреть пример готового сайта (в темных и светлых тонах) на основании созданной палитры.

Во вкладке «Список цветов» отображаются все цвета созданной палитры и их html-коды. В меню «Сохранить» (см. рис. 3.29) предоставляется возможность сохранить созданную палитру в виде файлов HTML и CSS, а также экспортировать ее в файлы цветовых палитр различных графических редакторов.

К преимуществам данного веб-сервиса (по отношению, например, к тому же Adobe Kuler) можно отнести, во-первых, наличие русскоязычного интерфейса и, во-вторых, повышенную «гибкость отладки» созданной цветовой палитры. Также отсутствие регистрации (которая всегда занимает какое-то время) вносит свою долю комфорта при работе с сервисом. Недостатком ColorScheme.Ru является отсутствие базы готовых (созданных другими пользователями) цветовых решений, что частично компенсируется наличием функции «Случайная палитра» (пункт меню «Рандом»).

☑ **ColoRotate** [28], в отличие от большинства других генераторов цветовых схем, показывает палитру в 3D-пространстве, используя трехмерный конус (рис. 3.30). Как и в Adobe Kuler, пользователи могут сохранять и редактировать свои цветовые схемы, а также просматривать палитры, сделанные другими дизайнерами.

ColoRotate имеет интуитивно понятный интерфейс, что избавляет от необходимости запоминать или записывать сочетание цветов или цифр. Этот веб-сервис позволяет редактировать цвета, смешивать их между собой, изменять значения в различных пространствах цветов или генерировать гармоничные цветовые темы автоматически. В ходе этого процесса раскрываются взаимосвязи между цветами, при учете которых получается эстетически сбалансированная палитра цветов. Действительно, каждый пользователь может в трехмерном пространстве цвета выбрать ту палитру, которая будет соответствовать его потребностям.

Этот веб-сервис можно интегрировать в популярные дизайнерские программы Adobe Fireworks и Adobe Photoshop.



Рис. 3.30. Веб-сервис ColoRotate

☑ **Color Palette Generator** [29] — автоматический генератор цветowych палитр на основе предоставленной пользователем графической информации (рис. 3.31). Непохожесть данного сервиса на вышеописанные состоит в том, что Color Palette Generator получает цвета и собирает их в палитры при обработке цифровых изображений в графических форматах JPG или PNG (соответственно, файлы с расширением .jpg и .png).

Такой подход подбора цветовой гаммы для будущего сайта очень удобен, когда, например, в процессе поиска цвета пользователь «натолкнулся» на привлекательную (с точки зрения цветовой гаммы) фотографию. Далее пользователю остается загрузить эту фотографию в Color

DeGraeve.com contact about

Color Palette Generator

dull

- #334433
- #6699aa
- #88aaaa
- #aaccce
- #447799

vibrant

- #225533
- #44bbcc
- #88dddd
- #bbeeff
- #0055bb

URL of image:

Make color schemes. Enter the URL of an image to get a color palette that matches the image. This is useful for coming up with a website color scheme that matches a stock photo a client wants to work with. If you like this color palette generator, you might like [ColorHunter.com](#)

Hey Developers: Would you be interested in an **API** to get colors within an image? What would you be willing to pay for such a service? [Drop me a line.](#)

Рис. 3.31. Сервис Color Palette Generator

Palette Generator (загрузка осуществляется только по ссылке¹) и получить 2 варианта (тусклую (*dull*) и яркую (*vibrant*) цветовой палитры.

☑ **ColorMania** [30] — бесплатная программа выбора цвета, специально созданная для веб-дизайнеров, художников и разработчиков приложений (рис. 3.32). Это идеальное решение для просмотра, определе-

¹ Если фотография находится не в Интернете, а, например, на локальном компьютере, то ссылку (URL) для этой фотографии можно получить, предварительно загрузив ее либо на специализированный файлообменный сервер, либо в фотобанк — электронную библиотеку изображений.

ния, проверки и организации значений цветов. Программа ColorMania поддерживает различные цветовые модели и имеет возможность выбора цвета в любом месте на экране (захват экрана). Точность выбора цвета при захвате экрана (вплоть до одного пикселя) регулируется степенью увеличения встроенной в программу экранной лупы (от 1- до 20-кратного увеличения). В правом нижнем углу программы отображается увеличенная часть экрана: перекрестие в центре — точка захвата цвета.

В центре окна программы размещены «ползунки» цветовых каналов моделей RGB и HSV, с помощью которых можно манипулировать параметрами текущего цвета (образец текущего цвета — в левом верхнем углу окна). В правом верхнем углу окна расположена модель цветового пространства, которая также позволяет выбрать цвет (по наведению курсора на избранный цвет и одинарному клику левой кнопкой мыши).

Программа способна работать с палитрами из 6-ти цветов, составлять, загружать или записывать их на жесткий диск в специализированном формате (разрешение файла .plt). В программу также встроена возможность автономного выбора цвета по его названию: внизу окна

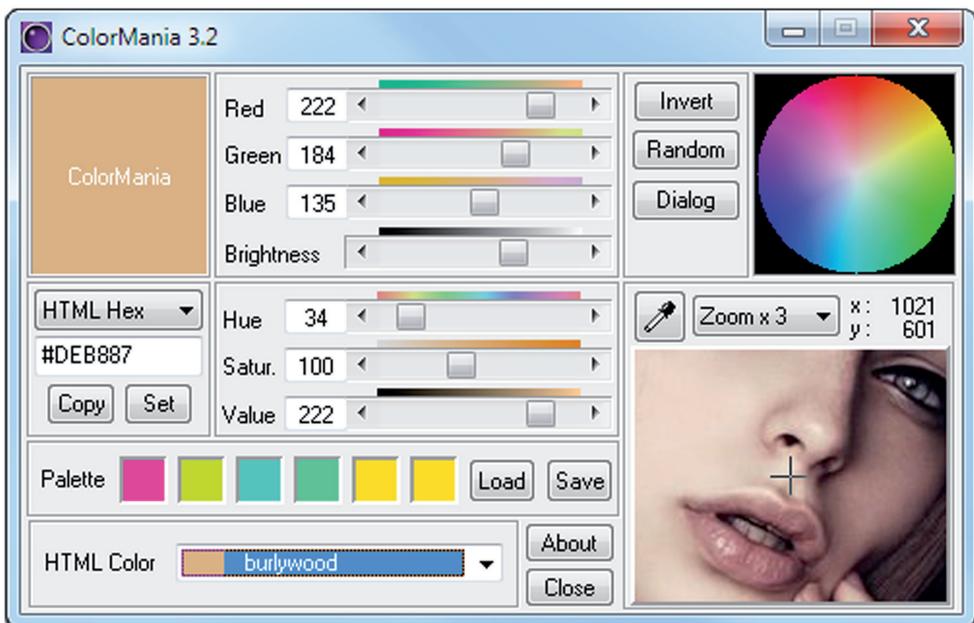


Рис. 3.32. Интерфейс программы ColorMania (версия 3.2)

программы в выпадающем списке «HTML Color» можно выбрать цвет по принятому в HTML имени (именованный цвет).

Для полноценного использования программы в сочетании с большим количеством других графических программ и в различных областях (дизайн, полиграфия, реклама, ...), ColorMania поддерживает большое количество форматов кодирования цвета, среди которых важный для нас HTML Hex — наиболее распространенный цветовой шестнадцатеричный код, который используется в HTML и CSS.

TypeTester [31] представляет собой интернет-приложение для подбора (тестирования) и сравнения шрифтов с целью поиска наиболее подходящего для веб-проекта. Отличительная особенность данного сервиса в том, что при тестировании шрифта учитываются не только начертания и размер символов шрифта, а также возможность задания цвета для символов и фона. При этом, поскольку работа с сервисом производится через веб-браузер, тестирование и подбор шрифтов выполняется в «полевых» условиях: можно сразу оценить, как будет отображаться выбранный шрифт тем или иным браузером.

Ни для кого не секрет, что подбор нужного шрифта для сайта — процесс долгий и кропотливый. Даже самые небольшие изменения влияют на удобочитаемость информации неожиданным образом, и потому часто приходится перепробовать множество разных вариантов, перед тем как найти нужный.

На сегодняшний день онлайн-сервис TypeTester (рис. 3.33) является лучшим решением, призванным облегчить соответствующий «удел» разработчиков и дизайнеров сайтов.

Сервис предоставляет отображение исходного текста (пользователь может набрать текст на клавиатуре или вставить его из буфера обмена в поле «Simple text») в трех отдельно настраиваемых столбцах. Каждый из столбцов имеет свою панель настройки параметров шрифта (рис. 3.34), в которой установленные значения применяются к тексту в столбце в виде повторяющихся абзацев с различными вариациями шрифтовых начертаний:

- Regular — нормальное начертание шрифта.
- Bold — полужирное начертание шрифта.
- Italic — курсивное начертание шрифта.
- Bold Italic — полужирный курсивный шрифт.

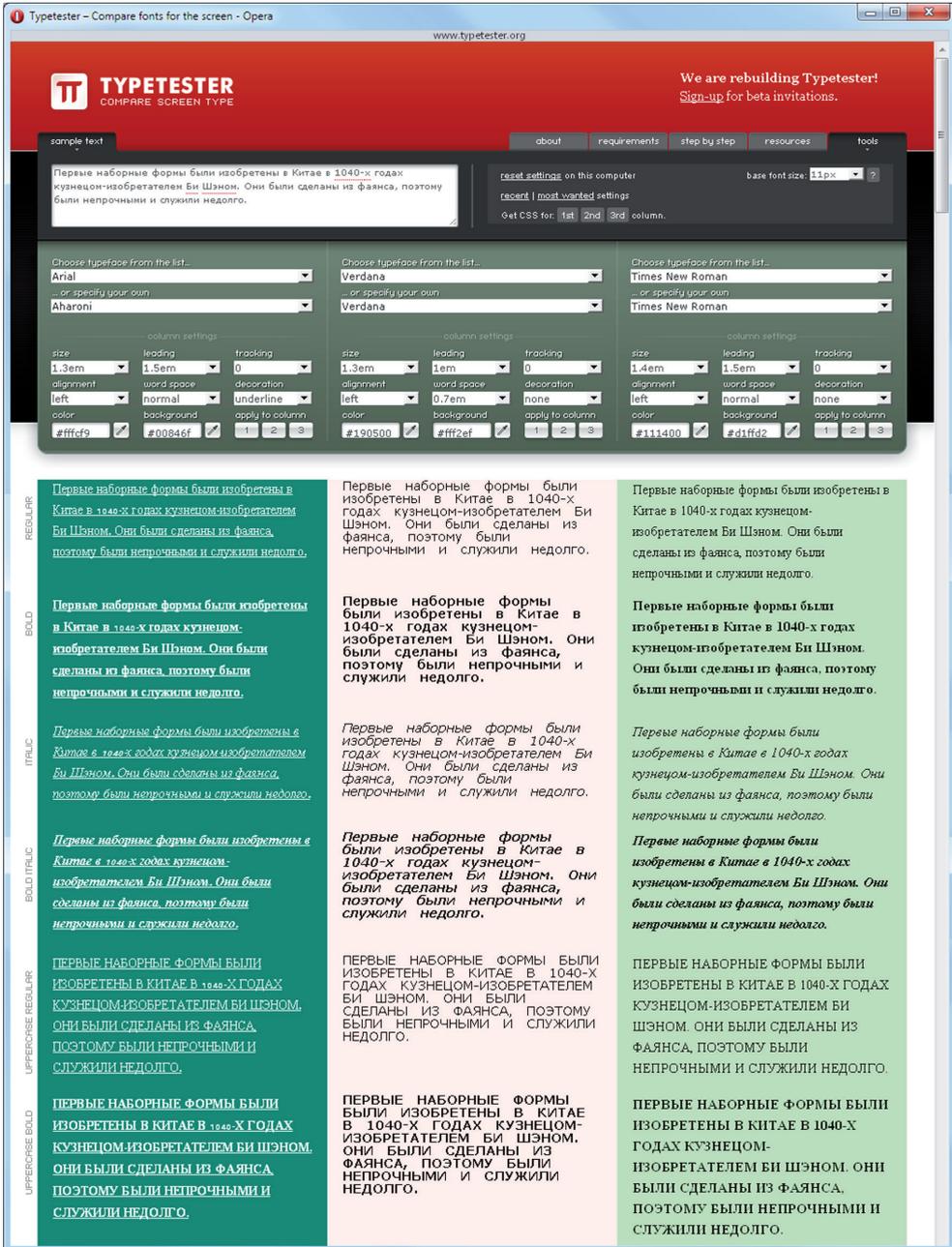


Рис. 3.33. Сервис TypeTester, показывающий образец текста при различных начертаниях и настройках параметров для выбранных гарнитур Arial, Verdana и Times New Roman

- Uppercase (Uppercase Regular, Uppercase Bold, Uppercase Italic, Uppercase Bold Italic) — перечисленные выше четыре варианта начертания для текста, набранного строчными буквами.
- Small Caps (Small Caps Regular, Small Caps Bold, Small Caps Italic, Small Caps Bold Italic) — первые четыре варианта шрифтового начертания для текста, набранного капителью.

Панель настроек (см. рис. 3.34) позволяет выбрать следующие параметры:

1. Гранитура шрифта. Шрифты, выбранные из выпадающего списка «Choose typeface from the list...» делятся на ряд групп (список составлен разработчиками сервиса).

2. Шрифты, выбранные из второго выпадающего списка «... or specify your own», установлены на компьютере пользователя, в котором запущен этот сервис.

3. Размер шрифта (выпадающий список «size») позволяет установить размер в относительных единицах измерения CSS.

4. Интерлиньяж (выпадающий список «leading») — расстояние между строками. Также задается в относительных единицах измерения по отношению к базовому размеру шрифта.

5. Трекинг (выпадающий список «tracking») — расстояние между буквами, задается положительными и отрицательными значениями в пунктах.

6. Выключка текста (выпадающий список «alignment»), при установке значений которой (justify, left, right, center) тексту задается выравнивание по краям, слева, справа, по центру.

7. Расстояние между словами (выпадающий список «word space») устанавливается в относительных единицах и может принимать отрицательные (в сторону сужения) и положительные (в сторону расширения) значения.

8. Декорирование (выпадающий список «decoration») позволяет подчеркнуть, надчеркнуть или перечеркнуть строки текста.

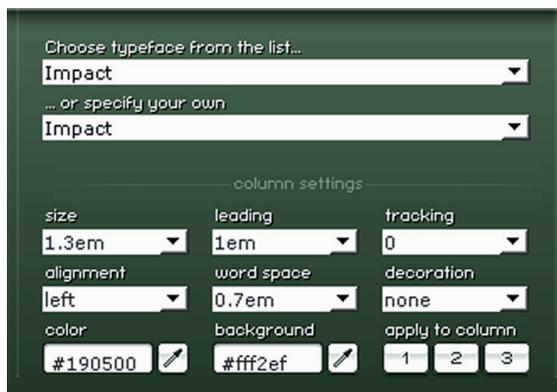


Рис. 3.34. Панель настроек параметров для одного из трех столбцов

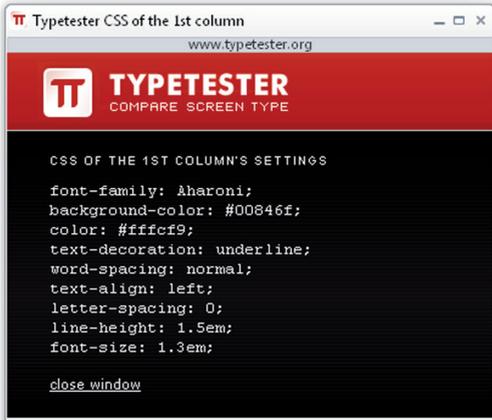


Рис. 3.35. Результат работы с сервисом TypeTester (код CSS для текста из 1-го столбца (см. рис. 3.33))

метров соседних столбцов (выбираются по номеру столбца).

После того как идеальное решение найдено (в одном из столбцов), во вкладке «Tools» в строке «Get CSS for: 1st 2nd 3rd column» выбирается номер того самого «идеального» столбца. После выбора открывается отдельное окно со стилями шрифта (рис. 3.35), которые нужно просто скопировать в файл CSS своего проекта.

Таким образом, веб-сервис TypeTester помогает подобрать идеальные стиль текста и цветовую гамму веб-страницы путем сравнения трех вариантов: «сравниваю лучшее с лучшим — и выбираю наилучшее!». Используя TypeTester можно быстро и эффективно не только подобрать нужный шрифт, но и улучшить дизайн разрабатываемого сайта.

Контрольные вопросы

1. Что такое веб-дизайн? В чем заключается деятельность веб-дизайнера?
2. Перечислите основные стили шрифтов. Кратко охарактеризуйте каждую шрифтовую категорию.
3. Какие параметры составляют для шрифта общую характеристику?
4. Какие параметры могут повлиять на четкость шрифта?

9. Цвет букв (поле «color») устанавливает цвет символов текста по введенному hex-коду. Также доступен выбор цвета из цветовой палитры по нажатию пиктограммы «Пипетка» (повторное нажатие закрывает цветовую палитру).

10. Цвет фона (поле «background») устанавливается аналогично цвету букв.

11. Кнопки в группе «apple to column» позволяют устанавливать значения параметров в текущей панели из панелей параметров соседних столбцов (выбираются по номеру столбца).

5. Как повысить читабельность текстов с помощью шрифта?
6. В чем заключается уместность шрифта с точки зрения дизайнера?
7. Что такое компьютерный шрифт? Перечислите типы компьютерных шрифтов.
8. Дайте характеристику форматам векторных шрифтов PostScript Type 1, TrueType и OpenType.
9. Какие задачи возложены на шрифтовые менеджеры? Перечислите основные возможности фонт-менеджеров.
10. Для каких целей предназначены шрифтовые онлайн-сервисы?
11. Какие функциональные возможности у сервиса Google Web Fonts?
12. В чем заключается особенность работы сервиса «WhatTheFont»?
13. Перечислите и опишите четыре основных качества цвета.
14. Что такое спектральный цвет? Перечислите основные спектральные цвета.
15. Дайте определение хроматическому и ахроматическому цвету. Сколько оттенков из гамм хроматических и ахроматических цветов различает глаз человека?
16. Из каких цветов состоят 12 секторов цветового круга Иоханнеса Иттена?
17. Перечислите цвета, входящие в группы первичных, вторичных и третичных цветов цветового круга Иоханнеса Иттена.
18. Что такое комплементарный цвет? Перечислите пары комплементарных цветов цветового круга Иоханнеса Иттена.
19. Какими функциональными особенностями должна обладать программа для подбора цвета?
20. Какое назначение и функциональные возможности имеет сервис TypeTester? Опишите порядок работы с этим интернет-приложением.

3.2. Основы веб-типографики

Наш век справедливо называют «веком информации», так как большая часть поступающей к человеку информации представлена в текстовом виде. Однако в большинстве случаев авторы не занимаются оформлением своих текстов должным образом. Поэтому огромное количество полезной информации невозможно быстро найти или просто понять из-за пренебрежения правилами оформления и представления этой информации. Если автор хочет донести какую-либо мысль до своих читателей, необходимо следовать определенным правилам типографики, являющейся мощным инструментом для передачи письменной информации. Эти правила основаны на особенностях визуального восприятия человека. В противном случае можно остаться **неуслышанным и непонятым**.

Типографика представляет собой процесс графического оформления печатного текста посредством набора и верстки с использованием норм и правил для данного языка.

С появлением новых носителей информации (в частности, электронных текстов) типографика стала адаптироваться и для них. Поскольку текст является основной формой представления информации в Интернете, то его разработка должна заслуживать самого пристального внимания.

Распространение веб-ресурсов в Интернете, в том числе и как источников любой информации, определило становление экранной типографики, для которой в качестве объекта выступает текст, отображенный на экране мониторов (очевидно, что объектом книжной типографики является печатный текст). Поскольку поставщиками подобных текстов являются веб-ресурсы, то эти тексты справедливо можно назвать веб-текстами, а используемую для них типографику — **веб-типографикой**.

Макро-типографика (определяет общую структуру текста), в отличие от **микро-типографики** (рассматривает детальные аспекты шрифта и интервалов), охватывает многие грани того, что сегодня называют «информационным дизайном». Современные информационные дизай-

неры проделывают такую же работу над своими проектами, какую до недавнего времени выполняли типографы при ручном наборе текстов для печатных изданий. Только теперь ручной набор текста вытеснен машинным и представляет собой по большей части автоматизированный процесс.

Веб-типографика представляет собой процесс оформления текста для сайтов таким образом, чтобы сделать его «заметным» для пользователя: акцентировать внимание на важных моментах, сделать его чтение приятным и комфортным.

Веб-типографика — это форма искусства, достаточно важный элемент дизайна сайта. Овладев правилами и приемами веб-типографики, можно повысить эффективность сайта и, в частности, расширить аудиторию пользователей.

Веб-типографика как составляющая «информационного дизайна» текста, полностью основана на «человеческом» восприятии информации и является искусством придания веб-текстам привлекательности и удобочитаемости.

Комфортное чтение начинается с комфортного отображения текста на сайте. Отображение текста в абзацах, списках и блоках поможет пользователям с первого взгляда выделить самое важное в тексте, визуально отделить разные части статьи от основного тела страницы на сайте или в блоге.

Оформление текста для сайта все же отличается от подготовки текста для печатной продукции и вызвано, прежде всего, несопоставимостью сайта с печатным изданием:

- **Каждый экземпляр печатного издания — материальный объект**, который можно взять в руки, перенести в другое место и, наконец, уничтожить. Сайт является информационным объектом и обычно не связан с определенным материальным носителем. В этом смысле сайт больше похож на такие способы распространения информации, как телевидение или радио.
- **Материалы печатных изданий расположены линейно, последовательно.** Читатель перемещается по изданию с учетом его линейной структуры (фактически последовательно перелистывает страницы). Для сайтов характерна иерархическая структу-

ра, при которой отдельные веб-страницы невозможно выстроить последовательно. При перемещении по сайту пользователь ориентируется не на линейную, а на «древовидную» последовательность веб-страниц.

- **Отсутствие пользовательских настроек** в печатном издании предполагает, что все читатели видят одни и те же параметры текста (размеры, начертания и пр.) при одних и тех же условиях. При просмотре содержимого сайта пользователь может настроить веб-страницу «под себя», например, увеличить размер шрифта, отключить выполнение сценариев и загрузку изображений.
- И, наконец, **история развития печатных изданий** — несколько столетий (а если учитывать письма и документы, созданные до изобретения книгопечатания, — несколько тысячелетий). История развития сайтов — чуть больше двадцати лет!

Перечисленные выше положения делают непохожими книжную типографику и набирающую обороты веб-типографику, и в то же время это не убавляет значимости последней, ведь недалек тот день когда компьютерные тексты будут полностью доминировать над печатными.

3.2.1. Типографское форматирование текста

Практически весь опыт, все знания об окружающем мире человечество накапливает в текстовой форме. Поэтому текст является основным способом представления информации как на печатных, так и на веб-страницах. В Интернете именно текстовую информацию ищут большинство людей, поэтому во многом от того, насколько грамотно и удобно структурирован текст на веб-страницах, будет зависеть — задержится ли посетитель на сайте или станет искать более удобный источник информации.

В то время как шрифт определяет наборы символов, отличающихся внешним видом, текст представляет собой определенное сочетание (чередование) этих символов:

Текст (от лат. *textus* — ткань; сплетение, сочетание) — связная и полная последовательность символов.

Так определяется текст, напечатанный на листе бумаги (типографский, печатный текст) или выведенный на экран монитора (электронный, цифровой текст), но в более широком смысле под текстом понимают словесную запись мыслей, сообщений, речи.

В компьютерной терминологии можно выделить два определения, связанные с понятием «оцифрованного» текста (фактически перевод печатного текста в цифровую форму):

Текстовые данные (текстовый формат) — представление информации в вычислительной системе в виде последовательности печатных символов.

Текстовый файл — компьютерный файл, содержащий текстовые данные, как правило, организованные в виде набора (массива) строк.

Текстовые данные также часто понимаются в более узком смысле как текст на каких-либо языках (формальных или естественных), который может быть прочитан и понят человеком. Следовательно, от того как «подан» текст зависит восприятие содержащейся в нем информации.

Процесс оформления (настройки внешнего вида) текста, связанный с подбором сочетания различных признаков текста (шрифтов, цвета и др.), изменяющих его внешний вид (но не содержание!), называют **форматированием текста**.

Форматирование текста в веб-типографике целесообразно рассматривать, определив основные понятия и назначения атрибутов стиля текста, принятых в типографике, и выполнив форматирование компьютерного текста для веб-страниц с использованием средств HTML и CSS.

По своей **структуре** текст делится на следующие категории:

- Основной текст** (англ. *main text*) несет в себе самую важную информацию и в основном предназначен для сплошного чтения.
- Дополнительный текст** (англ. *supplementary text*) поясняет основной текст или несет дополнительную информацию (примечания, сноски, комментарии, подрисуночные подписи, текстовые или цифровые таблицы и т. д.). Под категорию дополнительного текста также попадают аннотации, предисловия, послесловия и приложения.

- **Вспомогательный** (справочновспомогательный, служебный) текст (англ. *auxiliary text*) — важная часть навигационного аппарата текста, предназначенная для поиска нужной информации. Под эту категорию попадают оглавление или содержание, колонцифры¹, колонтитулы², указатели.
- **Рубрикация** (англ. *rubrication*) — часть навигационного аппарата текста, служащая для ориентирования в нем. Это система заголовков и возглавляемых ими подразделов (рубрик), выражающая логическую связь и соподчиненность обозначаемых ими частей текста (произведений, разделов, глав, параграфов).

Для удобства сплошного или выборочного чтения основной текст должен быть разделен на смысловые фрагменты. Самый простой способ такого деления — **разбивка текста на абзацы**.

Абзац (англ. *paragraph*) — фрагмент текста, состоящий из нескольких предложений и выражающий некую смысловую единицу.

В HTML такая разбивка текста на абзацы возложена на тег `<p>`, который и определяет **текстовый абзац**. Текстовый блок, заключенный в теги `<p>...</p>`, является абзацем и всегда начинается с новой строки. Абзацы текста, идущие друг за другом, разделяются между собой специальным отступом (отбивка), величиной которого можно управлять с помощью стилей CSS.

Так для чего необходимо форматирование текста? Ответ очевиден: в первую очередь для того, чтоб сам текст лучше воспринимался. Если представить, например, в газете или книге, текст без форматирования (просто сплошной однородный текстовый поток), то ни нашлось бы ни одного человека, которому было бы удобно и комфортно читать такое издание.

¹ **Колонцифра** — порядковый номер страницы издания, помещаемый вверху или внизу страницы, в ее наружных углах или посередине.

² **Колонтитул** (фр. *colonne* — столбец и лат. *titulus* — надпись, заголовок) — заголовочный текст или иные данные, помещаемые над (верхний колонтитул) или под (нижний колонтитул) основным текстом страницы (или нескольких страниц) печатно-го издания или электронного документа.

Под форматированием (разметкой) текста подразумевается его визуальное оформление. К примеру, важные части текста можно выделить полужирным шрифтом или курсивом (или каким-нибудь другим способом внутритекстового выделения), сразу привлекая к ним внимание читателей. Позиционирование текста, его метрические характеристики (выравнивание, интерлиньяж, трекинг, кернинг и пр.) и элементы художественного оформления (буквицы, линейки) — это все является элементами форматирования, призванными, в итоге, получить качественный текст с точки зрения его комфортного чтения и эффективного восприятия прочитанного.

3.2.2. Гарнитура, размер и начертание шрифта

Выбор гарнитуры шрифта для текста — это не только эстетическое решение. Конкретный вид шрифта может украсить текст, который он представляет, но помимо этого шрифт также может нести и полезную информацию, например, выделять что-либо.

Выбор гарнитуры шрифта для текста также сказывается на его **удобочитаемости** (англ. *readability*) (не путаем с читабельностью!)¹, которая основана на том, что некоторые шрифты читаются легче, чем другие. Факт удобочитаемости давно интересует типографов, однако проведенные исследования степени легкости восприятия текста в зависимости от гарнитуры шрифта не привели к однозначным результатам, поскольку выводы часто оказываются противоречивыми [32; 33]. Можно сказать, что удобочитаемыми являются те шрифты, которые привычнее. Подтверждением этого могут служить слова британского типографа Эрика Гилла: *«Практически удобочитаемо то, к чему мы привыкли. Но это не значит, что из-за привычки к чему-то намного менее удобочитаемому, чем то, что могло бы нам послужить при надлежащем навыке, следует отказываться от попыток изменить существующее положение вещей»* [34, с. 127].

Поскольку браузер может использовать для отображения веб-страницы только те шрифты, которые установлены на компьютере у пользователя (посетителя) сайта, то с точки зрения корректного отображения текста в браузере шрифты можно условно разделить на две категории:

- шрифты, которые без проблем отобразятся у подавляющего большинства пользователей;
- шрифты, которые могут отсутствовать у достаточно большой группы пользователей.

Чтобы разобраться к какой категории относится тот или иной шрифт, нужно рассмотреть ряд моментов и определиться в понятиях «**стандартный шрифт**», «**нестандартный шрифт**» и «**безопасный шрифт**», а также уметь отличать эти понятия друг от друга. Одно ясно: на набор шрифтов, установленный непосредственно на компьютере, в котором создается сайт, опираться нельзя!

Стандартные шрифты — это набор шрифтов, устанавливаемый вместе с операционной системой. Но поскольку на клиентских компьютерах установлены разные операционные системы, то и стандартные шрифты у них будут различными.

Перечень стандартных шрифтов разных версий ОС Windows можно посмотреть, например, в статье «Список стандартных шрифтов Windows» [35] или на сайте Microsoft [36], а перечень стандартных шрифтов Mac OS — на веб-странице [37]. Что касается операционных систем Unix/Linux, то единый набор шрифтов у них отсутствует. Согласно статистике [38], более 60 % пользователей Unix/Linux имеют на своем компьютере шрифты набора «Core fonts for the Web» [39].

В Интернете исторически сложилось такое понятие, как «безопасные» веб-шрифты. **Безопасные шрифты** — это такие шрифты, которые являются стандартными для всех операционных систем. Однако такая ситуация далека до идеала, поэтому на сегодняшний день **абсолютно безопасных шрифтов не существует!** Но отдельные шрифты можно называть безопасными с некоторыми оговорками.

Основой для определения «безопасных» шрифтов послужили шрифты операционной системы Windows, которые кроме того используются в других ОС. На основе шрифтов Windows, использующихся и в других операционных системах, сформировался следующий список «безопасных» кириллических веб-шрифтов, т. е. тех, которые без проблем отобразятся у подавляющего большинства пользователей:

- | | | |
|---|--|---|
| <input type="checkbox"/> Arial; | <input type="checkbox"/> Courier New; | <input type="checkbox"/> Times New Roman; |
| <input type="checkbox"/> Arial Black ; | <input type="checkbox"/> Georgia; | <input type="checkbox"/> Trebuchet MS; |
| <input type="checkbox"/> Comic Sans MS; | <input type="checkbox"/> Impact ; | <input type="checkbox"/> Verdana. |

Подключить одну или несколько гарнитур определенной категории к веб-странице можно с помощью свойства CSS `font-family`.

С учетом возможностей стилевого свойства `font-family` целесообразно собирать шрифты в списки (линейки) по названиям и типу (родовому семейству). И хотя не существует абсолютно безопасных шрифтов, существуют безопасные линейки, которые называют шрифтовыми стеками CSS.

Ниже приведены безопасные стеки CSS, учитывающие поддержку кириллицы в шрифтах:

```
1 font-family: "Arial Black", "Helvetica CY",
2 "Nimbus Sans L", sans-serif;
3 font-family: Arial, "Helvetica CY", "Nimbus Sans L",
4 sans-serif;
5 font-family: "Comic Sans MS", "Monaco CY", cursive;
6 font-family: "Courier New", "Nimbus Mono L", monospace;
7 font-family: Georgia, "Century Schoolbook L", serif;
8 font-family: Impact, "Charcoal CY", sans-serif;
9 font-family: "Lucida Console", Monaco, monospace;
10 font-family: "Lucida Sans Unicode", "Lucida Grande",
11 sans-serif;
12 font-family: "Palatino Linotype", "Book Antiqua",
13 Palatino, serif;
14 font-family: Tahoma, "Geneva CY", sans-serif;
15 font-family: "Times New Roman", "Times CY",
16 "Nimbus Roman No9 L", serif;
17 font-family: "Trebuchet MS", "Helvetica CY", sans-serif;
18 font-family: Verdana, "Geneva CY", "DejaVu Sans",
19 sans-serif;
```

Нестандартные шрифты, в отличие от стандартных, — это шрифты, которые с большой вероятностью будут отсутствовать у большинства посетителей сайта. Внедрять нестандартные шрифты можно с помощью CSS-директивы `@font-face`, которая позволяет определить настройки шрифтов, а также загрузить специфичный шрифт на компьютер пользователя.

Использование **разных размеров шрифтов** позволяет выразить иерархию содержания текста (например, титулы крупнее заголовков, которые, в свою очередь, крупнее подзаголовков, а те — крупнее основного текста), а использование **полужирных или курсивных шрифтов** — традиционный способ создания типографического выделения

фрагмента текста (как в этом абзаце). Но если в пределах основного текста полужирный шрифт выглядит слишком вызывающе, то более уместен курсивный шрифт.

Важнейшее назначение курсивного шрифта — выделение фрагмента текста и придание ему определенного отличия. Однако нужно помнить, что текст, набранный курсивным шрифтом, тяжелее воспринимается визуально, чем текст со шрифтом стандартного набора. Курсив чаще всего используется для выделения [32]: заголовков книг в тексте; названий произведений искусства и музыкальных композиций; названий фильмов, пьес и телевизионных передач; терминов при первом их упоминании в тексте; пояснений к иллюстрациям; цитат и эпиграфов.

Свойство `font-size` задает размер шрифта в абсолютных и относительных величинах — в поддерживаемых браузерами единицах измерения размера или в процентных долях (относительно размера шрифта родительского элемента¹).

Абсолютный размер шрифта элемента задает следующий набор констант:

- `xx-small` — эквивалентно заданию размера `size="1"` относительно настроек браузера в устаревшем теге ``;
- `x-small` — размер шрифта, установленный в настройках браузера по умолчанию;
- `small` — эквивалентно ``;
- `medium` — эквивалентно ``;
- `large` — эквивалентно ``;
- `x-large` — эквивалентно ``;
- `xx-large` — эквивалентно ``.

В следующем коде HTML показано использование задания абсолютного размера шрифта с помощью набора констант, а результат данного примера показан на рис. 3.36.

¹ Объекты CSS (элементы, селекторы), которые объявлены в теле другого объекта, называются дочерними, в то время как этот другой объект — родительский. Обычно дочерние объекты наследуют большинство свойств родительских объектов по умолчанию. Однако ряд свойств, наследование которых противоречит здравому смыслу (отступы, размеры, границы и пр.), не наследуется. За «принудительное» наследование значений свойств у родительского элемента в CSS определено ключевое слово (значение свойства) **inherit**.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="windows-1251">
5     <title>набор констант для задания абсолютного
6       размера шрифта</title>
7   <style>
8     p {
9       font-family: "Lucida Sans Unicode",
10        "Lucida Grande", sans-serif;
11    }
12    .sm1 { font-size: xx-small; }
13    .sm2 { font-size: x-small; }
14    .sm3 { font-size: small; }
15    .med { font-size: medium; }
16    .lr1 { font-size: xx-large; }
17    .lr2 { font-size: x-large; }
18    .lr3 { font-size: large; }
19  </style>
20 </head>
21 <body>
22   <p class="sm1">Размер xx-small</p>
23   <p class="sm2">Размер x-small</p>
24   <p class="sm3">Размер small</p>
25   <p class="med">Размер medium</p>
26   <p class="lr3">Размер large</p>
27   <p class="lr2">Размер x-large</p>
28   <p class="lr1">Размер xx-large</p>
29 </body>
30 </html>
```

Абсолютные размеры шрифта также можно задавать числовыми значениями, выраженными в следующих единицах измерения (рис. 3.37):

- pt — единица измерения высоты символа, заданная в типографских пунктах, и являющаяся самой часто используемой абсолютной единицей измерения, поскольку также используется для задания размера шрифта в текстовых редакторах;
- pc — единица измерения высоты символа, заданная в типографских пиках;
- in — единица измерения высоты символа, заданная в дюймах (1 дюйм равен 2,54 см);
- cm — единица измерения высоты символа, заданная в сантиметрах;

- mm — единица измерения высоты символа, заданная в миллиметрах.

Относительные размеры шрифта устанавливают следующие константы:

- larger — увеличивает размер шрифта на 1 относительно родительского элемента (эквивалентно ``);
- smaller — уменьшает размер шрифта на 1 относительно родительского элемента (эквивалентно ``).

Поскольку размер унаследован, то он непосредственно зависит от значения свойства `font-size` у родительского элемента (рис. 3.38).

Размер шрифта (или, точнее, высоту символов) также можно задать в виде числового значения с указанием поддерживаемых браузерами **относительных единиц измерения**:

- em — единица измерения высоты шрифта, зависящая от размера шрифта в браузере. Поэтому изначально 1em равен размеру шрифта, заданного в браузере по умолчанию. В том случае, когда em используется для определенного дочернего элемента, за 1em принимается размер шрифта его родительского элемента.

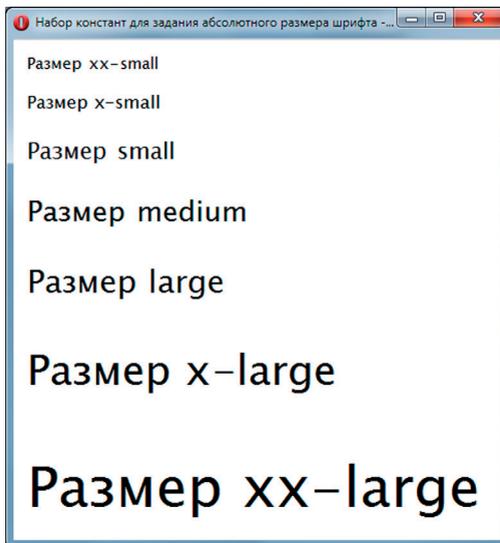


Рис. 3.36. Задание абсолютного размера шрифта с помощью набора констант

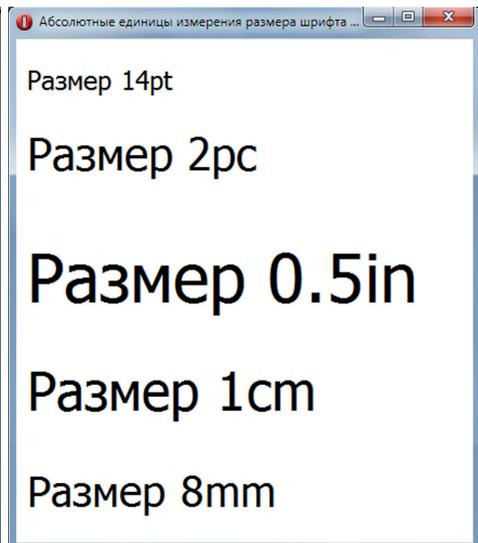


Рис. 3.37. Задание размера шрифта с помощью абсолютных единиц измерения

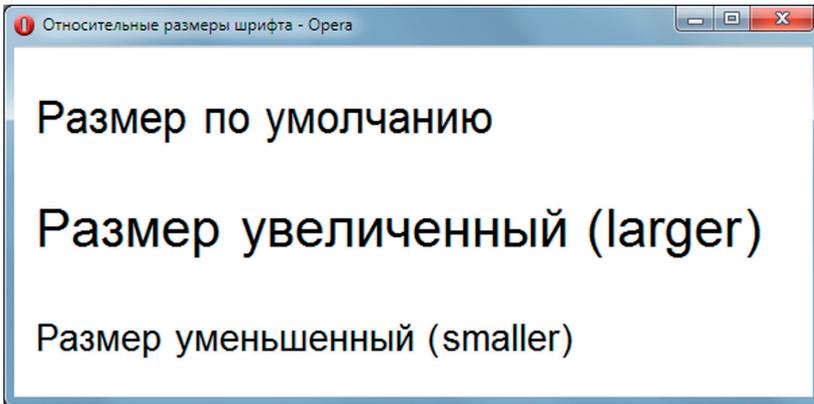


Рис. 3.38. Задание относительного размера шрифта с помощью констант *larger* и *smaller*

- *ex* — единица измерения высоты шрифта, равная высоте символа «x» в нижнем регистре. Значение параметра *ex*, также как и *em*, привязано к размеру шрифта, заданного в браузере по умолчанию, или к размеру шрифта родительского элемента.
- *px* — единица измерения высоты символа, заданная в пикселях. Размер пикселя зависит от разрешения экрана монитора, его технических характеристик, а также от используемой платформы (например, для платформы Macintosh экранное разрешение равно 72 пикселя на дюйм, тогда как для платформы Windows — 96 пикселей на дюйм).
- % — единица измерения высоты шрифта, выраженная в процентах относительно размера шрифта, заданного в браузере по умолчанию, размера шрифта родительского элемента. При этом 100 % = 1 *em*, 120 % = 1,2 *em*, 200 % = 2 *em* и т. д.

В следующем примере показано использование задания размера шрифта относительными единицами измерения, а результат данного примера показан на рис. 3.39:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="windows-1251">
5     <title>Относительные единицы измерения
6       размера шрифта</title>
7   <style>
```

```

8     p {
9       font-family: "Lucida Sans Unicode",
10        "Lucida Grande", sans-serif;
11     }
12     .em { font-size: 2.8em; }
13     .ex { font-size: 4.5ex; }
14     .px { font-size: 25px; }
15     .per { font-size: 125%; }
16   </style>
17   </head>
18   <body>
19     <p class="em">Размер 2.8em</p>
20     <p class="ex">Размер 4.5ex</p>
21     <p class="px">Размер 25px</p>
22     <p class="per">Размер 125%</p>
23   </body>
24 </html>

```

Свойство CSS `font-style` задает стиль шрифта, **под которым подразумевается его начертание**. Начертание шрифта может быть обычным, курсивным или наклонным. При этом следует различать курсивное и наклонное начертания шрифта.

Начертание шрифта устанавливается с помощью следующих значений:

- `normal` — обычное начертание;
- `italic` — курсив;
- `oblique` — наклон.

Когда для текста установлено курсивное начертание, браузер обращается к операционной системе для поиска подходящего шрифта из гарнитуры. Если заданный шрифт не найден, браузер использует специальный алгоритм для имитации нужного

вида текста, т. е. фактически выводит текст в наклонном начертании (при этом качество «отрисовки» символов текста может оказаться неудовлетворительным, особенно при печати веб-документа).

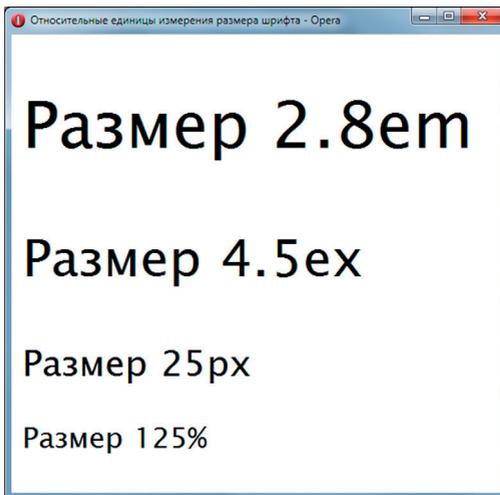


Рис. 3.39. Задание размера шрифта с помощью относительных единиц измерения

В большинстве случаев браузер не может отобразить наклонное начертание и заменяет его курсивным. Это хорошо заметно на приведенном выше рисунке.

Свойство `font-weight` задает начертание шрифта определенной насыщенности. Значение насыщенности устанавливается в цифровом эквиваленте (условные единицы): от 100 до 900 с шагом 100.

Сверхсветлое начертание, которое может отобразить браузер, имеет значение 100, а сверхжирное — 900. Нормальное (установленное по умолчанию) начертание шрифта эквивалентно насыщенности, равной 400 условных единиц, стандартный полужирный текст — значение насыщенности, равное 700 условных единиц.

Кроме этого, **насыщенность шрифта** можно задавать и с помощью ключевых слов:

- `bold` — полужирное начертание (700 условных единиц);
- `bolder` — жирное начертание;
- `lighter` — светлое начертание;
- `normal` — нормальное начертание (400 условных единиц).

Обычно браузеры не могут адекватно показать требуемую насыщенность шрифта из всего диапазона условных единиц, поэтому переключаются между значениями `bold`, `normal` и `lighter`. На практике начертание шрифта в браузерах обычно ограничено всего двумя вариантами: нормальным и полужирным начертаниями.

* * * * *

С помощью **собирательного свойства** `font` можно одновременно задать стилевые свойства шрифта в следующем порядке¹:

```
font: [font-style || font-variant || font-weight]
      font-size [/line-height]
      font-family
```

¹ Здесь и далее при описании синтаксиса HTML и CSS применяются следующие обозначения: вертикальная черта между значениями («|») указывает на то, что надо выбрать только одно значение из предложенных; двойная вертикальная черта («||») обозначает, что каждое значение может использоваться самостоятельно или совместно с другими через пробел; квадратные скобки («[», «]») помечают группу, из которой, как правило, выбирается одно значение, причем оно не является обязательным к использованию.

Синтаксис свойства `font` можно условно разделить на три группы. Первая группа [`font-style` || `font-variant` || `font-weight`] состоит из рассмотренных ранее свойств `font-style`, `font-weight` и свойства `font-variant`, служащего для отображения текста капителью. Вторая группа `font-size` [/`line-height`] задает размер шрифта (свойство `font-size`) и устанавливает интерлиньяж текста (свойство `line-height`), отделяя первое значение от второго косой чертой. Последняя группа — стилевое свойство `font-family`.

В качестве **обязательных значений** свойства `font` указываются размер шрифта (`font-size`) и его семейство (`font-family`). Остальные значения являются опциональными (в синтаксисе заключены в квадратные скобки) и задаются по усмотрению веб-дизайнера или верстальщика веб-страницы.

Рассмотрим **примеры использования** свойства `font` из приведенного ниже фрагмента файла CSS:

```
1 .c1 { font: normal 12pt "Helvetica Neue", serif; }
2 .c2 { font: oblique small-caps 10mm/10px fantasy; }
3 .c3 { font: bold italic 110% sans-serif; }
```

Для класса `c1` определены следующие параметры шрифта: нормальное начертание (значение `normal` применяется сразу к двум свойствам: `font-style` и `font-weight`.); размер шрифта 12 пунктов; гарнитура Helvetica Neue или любой шрифт с засечками (`serif`).

Текст, параметры которого определены в классе `c2`, имеет жирное начертание (`oblique`), отображен капителью (значение `small-caps` для свойства `font-variant`), имеет сантиметровую высоту (10mm) символов и межстрочное расстояние в 10 пикселей (/10px), а также представлен декоративным шрифтом (`fantasy`).

Для текста класса `c3` установлено полужирное (`bold`) и курсивное (`italic`) начертание текста (порядок значений в первой группе `font` не важен, поэтому `bold` и `italic` можно поменять местами), размер текста задан в процентах, а в качестве гарнитуры используется шрифт без засечек (`sans-serif`).

3.2.3. Капитель

Капитель (англ. *small caps*; от лат. *capitellum* — головка) — начертание в гарнитуре, в которой строчные знаки выглядят, как уменьшенные прописные. При использовании капители прописные (заглавные) и строчные буквы сохраняются, т. е. первая буква предложения больше, чем остальной текст (рис. 3.40).

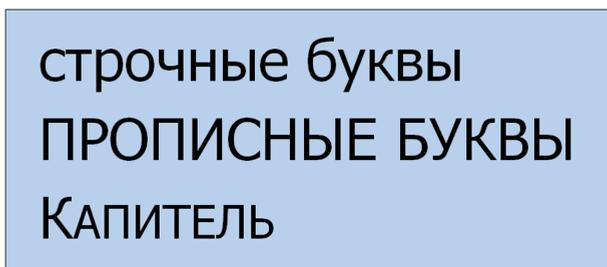


Рис. 3.40. Вид капители и ее отличие от прописных и строчных букв

Капитель применяется в латинском наборе со времен рукописных книг. Раньше капитель использовалась для набора заголовков и первых строк разделов. Сейчас она используется для тех же целей, а также:

- для набора имен;
- для набора некоторых названий и аббревиатур;
- для шрифтовых выделений в тексте.

Для установки капители в HTML применяется атрибут `font-variant` со значением `small-caps`:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>капитель</title>
6     <style>
7       p {
8         font: 24pt serif;
9       }
10      .small_capitals {
11        font-variant: small-caps;
12      }
```

```

13     .lower_case {
14         text-transform: lowercase;
15     }
16     .upper_case {
17         text-transform: uppercase;
18     }
19 </style>
20 </head>
21 <body>
22     <p class="lower_case">Строчные Буквы</p>
23     <p class="upper_case">Прописные Буквы</p>
24     <p class="small_capitals">Капиталь</p>
25 </body>
26 </html>

```

В этом примере (рис. 3.41) использовано свойство `text-transform`, которое управляет преобразованием текста элемента в строчные или прописные символы. Значения для `text-transform` могут быть следующими:

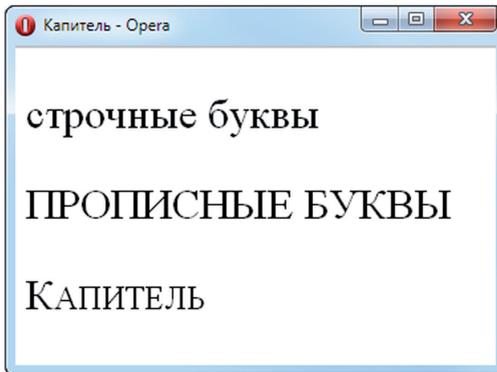


Рис. 3.41. Отображение текста капителью, а также строчными и прописными буквами

щими:

- `capitalize` — первая буква каждого слова в тексте будет заглавной (остальные буквы свой вид не меняют).
- `lowercase` — все символы текста выводятся в нижнем регистре (строчные буквы);
- `uppercase` — все символы текста выводятся в верхнем регистре (прописные буквы);
- `none` — регистр символов не меняется.

3.2.4. Втяжка и отбивка текстов

Втяжка (англ. *indent*) — сокращение строки или нескольких строк (например, абзаца) относительно формата основного текста за счет отступа слева, справа или с обеих сторон.

Отбивка (англ. *space*) задает интервал в тексте путем увеличения промежутков между отдельными его фрагментами или элементами текстового поля (заголовки, абзацы, ссылки и пр.). Отбивка улучшает

читабельность текста за счет дробления монолитного текста на несколько блоков, это, в свою очередь, позволяет легко переводить взгляд читателя от одного фрагмента текста к другому.

Чаще всего отбивка применяется к абзацу (даже в случае отбивки, например, заголовка от основного текста, применяется отбивка к абзацу заголовка). В этом случае используются параметры **отбивка до абзаца** (англ. *space before*) — увеличение межстрочного пробела перед абзацем и **отбивка после абзаца** (англ. *space after*) — увеличение межстрочного пробела после абзаца (рис. 3.42).

Так, армия любит побеждать и не любит затяжной войны. Поэтому, полководец, понимающий войну, является хозяином судеб людей, хранителем безопасности государства.

ПЛАНИРОВАНИЕ НАПАДЕНИЯ

Сунь-Цзы сказал: «Метод использования войск следующий:

Наилучшее — сохранить столицу государства врага, на втором месте — разрушить его столицу. Наилучшее — сохранить его армию, на втором месте — разбить ее. Наилучшее — сохранить соединения врага, на втором месте — уничтожить их. Наилучшее — сохранить подразделения врага, на втором месте — уничтожить их. Наилучшее — сохранить «пятерки» врага, на втором месте — уничтожить их.

Поэтому одержать сто побед в ста сражениях — это не вершина превосходства. Подчинить армию врага не сражаясь — вот подлинная вершина превосходства».

Так, армия любит побеждать и не любит затяжной войны. Поэтому, полководец, понимающий войну, является хозяином судеб людей, хранителем безопасности государства.

ПЛАНИРОВАНИЕ НАПАДЕНИЯ

Сунь-Цзы сказал: «Метод использования войск следующий:

Наилучшее — сохранить столицу государства врага, на втором месте — разрушить его столицу. Наилучшее — сохранить его армию, на втором месте — разбить ее. Наилучшее — сохранить соединения врага, на втором месте — уничтожить их. Наилучшее — сохранить подразделения врага, на втором месте — уничтожить их. Наилучшее — сохранить «пятерки» врага, на втором месте — уничтожить их.

Поэтому одержать сто побед в ста сражениях — это не вершина превосходства. Подчинить армию врага не сражаясь — вот подлинная вершина превосходства».

Рис. 3.42. Фрагмент текста до и после отбивки абзацев. Параметры отбивки: 6 пунктов перед и после абзаца текста; 12 пунктов перед абзацем заголовка

Интерлиньяж (англ. *leading*) — это расстояние в пунктах между базовыми линиями шрифта соседних строк (рис. 3.43). Интерлиньяж часто называют междустрочием или вертикальным пробелом¹, а большинство компьютерных программ использует термин «межстрочный

¹ В контексте типографских определений для текстовых блоков (а не символов), значение слова «пробел» трактуется как незаполненная часть чего-либо. Пробел, как «пустой» символ, функционально принадлежит к знакам препинания.

пробел» (англ. *line spacing*). Оптимальная величина интерлиньяжа, как правило, равна сумме размера (кегля) шрифта и половины высоты символа (для межстрочного зазора). От правильно подобранного значения интерлиньяжа зависит удобочитаемость текста (которая также зависит и от правильных горизонтальных пробелов, подробнее о которых можно узнать в [32, с. 156–163, с. 204]). Во многих текстовых редакторах и типографических программах интерлиньяж по умолчанию составляет 120 % от кегля шрифта.

Интерлиньяж 1 пункт	Интерлиньяж 1,5 пункта	Интерлиньяж 2 пункта
Краев чужих неопытный любитель И своего всегдашний обвинитель, Я говорил: в отечестве моем Где верный ум, где гений мы найдем? Где гражданин с душою благородной, Возвышенной и пламенно свободной?	Краев чужих неопытный любитель И своего всегдашний обвинитель, Я говорил: в отечестве моем Где верный ум, где гений мы найдем? Где гражданин с душою благородной, Возвышенной и пламенно свободной?	Краев чужих неопытный любитель И своего всегдашний обвинитель, Я говорил: в отечестве моем Где верный ум, где гений мы найдем? Где гражданин с душою благородной, Возвышенной и пламенно свободной?

Рис. 3.43. Фрагмент текста при различных значениях интерлиньяжа

Отбивка для абзацев на веб-странице, которые создаются с помощью тега `<p>`, задана по умолчанию. Интервал в тексте путем изменения промежутков между отдельными его фрагментами или элементами текстового поля задается стилевыми свойствами `margin-top` (отступ сверху) и `margin-bottom` (отступ снизу).

Втяжка текстового блока задается стилевыми свойствами `margin-left` (отступ слева) и `margin-right` (отступ справа).

В атрибуте `margin` разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом:

- *одно значение* задает отступы одновременно от каждого края;
- *два значения* — отступ от верхнего и нижнего края (первое значение) и от правого и левого края (второе значение);
- *три значения* — отступ от верхнего края (первое значение), от левого и правого края (второе значение) и от нижнего края (третье значение);
- *четыре значения* — поочередно устанавливается отступ от верхнего, правого, нижнего и левого краев.

Величину отступов можно указывать в допустимых для CSS единицах измерения: пикселях (px), процентах (%), сантиметрах (cm) и др.

Значение может быть как положительным, так и отрицательным числом. Если в качестве какого-либо значения указано слово `auto`, то размер отступов будет автоматически рассчитан браузером.

Для отбивки и втяжки текстового блока также может использоваться стилевое свойство `padding`, которое устанавливает значение полей вокруг содержимого элемента. Количество и порядок значений для `padding` устанавливаются аналогично свойству `margin` (однако не используются `auto` и отрицательные значения).

В следующем примере показан код HTML с применением свойств `margin` и `padding` для абзаца текста, заключенного в рамку (стилевое свойство `border`):

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Отбивка и втяжка текста</title>
6     <style>
7       body {
8         margin: 0;           /* Убираем все отступы */
9       }                     /* в браузере по умолчанию */
10      .class_parent {
11        margin: 10% 5%;     /* Отступы вокруг элемента */
12        background: #036;   /* Цвет фона */
13        /* Поля вокруг с внешней стороны от рамки */
14        padding: 2em 10px;
15      }
16      .class_child {
17        border: 3px solid lightgray ; /* Параметры рамки:
18                                       ширина 3 px;
19                                       сплошная линия (solid);
20                                       цвет светло-серый */
21        padding: 10px 2cm; /* Поля между текстом и рамкой:
22                               втяжка по 2 см с каждой стороны;
23                               отбивка сверху и снизу
24                               по 10 пикселей */
25        margin: 10px;      /* Отступы вокруг текста */
26        color: #BEFFF1;    /* Цвет текста */
27      }
28    </style>
29  </head>
30  <body>
31    <div class="class_parent">
32      <div class="class_child">
```

```
33     Роман не двигается, сказали мы, но, кроме того,  
34     еще ни один характер, ни одно почти положение в нем  
35     не развиваются вплоть до половины третьего тома.  
36     Они только меняются, показывают новые стороны  
37     с каждым поворотом картины, когда она их  
38     захватывает, но не развиваются.  
39     Иначе и быть не могло.  
40     </div>  
41 </div>  
42 </body>  
43 </html>
```

В этом примере для наглядности используются различные системы изменения полей в одном свойстве `padding`. На практике не рекомендуется использовать такой подход, а указывать все значения (хотя бы в пределах одного стилевого свойства) в выбранной единице измерения. Рамка (`border`) в текстовом блоке размещена для облегчения анализа кода и помогает разобраться во взаимодействии границ полей и отступов для вложенных элементов `div` со стилями `class_parent` и `class_child`.

В примере используется блочный элемент `div`, который предназначен для выделения фрагмента веб-документа с целью изменения вида содержимого. Теги `<div>...</div>` применяются, как правило, для управления с помощью стилей их содержимого. Содержимое в тегах `<div>...</div>` всегда начинается с новой строки. После закрывающего тега `</div>` также добавляется перенос строки. На рис. 3.44 приведено выполнение рассмотренного примера в браузере Firefox.

* * * * *

Интерлиньяж, т. е. расстояние между базовыми линиями строк текста, в веб-странице вычисляется браузером автоматически в зависимости от размера и типа шрифта. Также значение интерлиньяжа можно установить самостоятельно с помощью упомянутого стилевого атрибута `line-height` или `font`.

В качестве значения `line-height` можно использовать множитель, процентную запись или точное значение, заданное в любых единицах, принятых в CSS — пикселях (`px`), дюймах (`in`), пунктах (`pt`) и др. Любое число больше нуля у аргумента `line-height` воспринимается как множитель от размера шрифта текущего текста. Например, значение 1.5

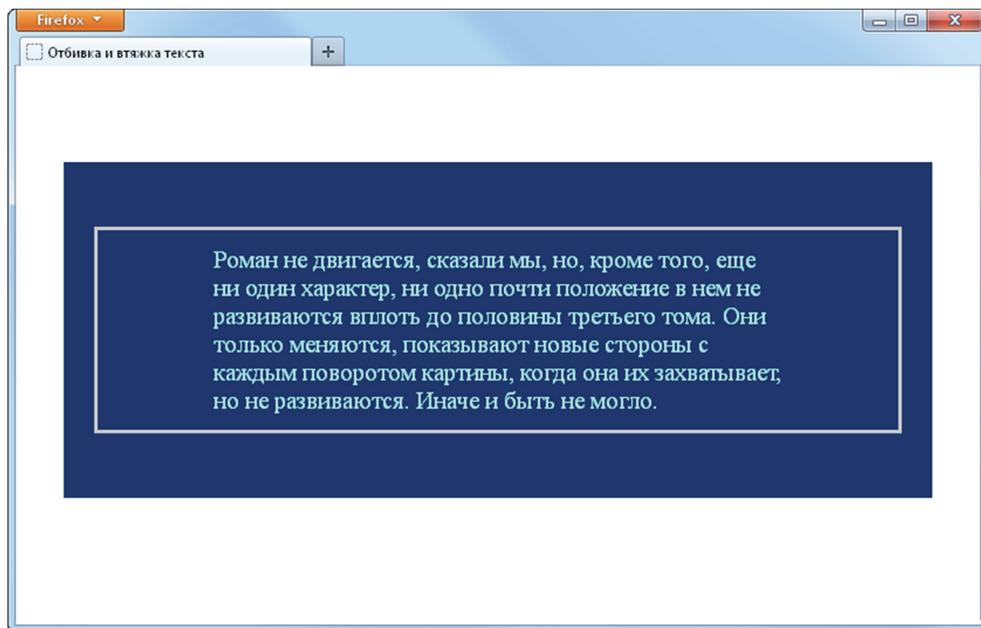


Рис. 3.44. Использование свойств CSS для отбивки и втяжки текстового блока

устанавливает полуторный межстрочный интервал. При использовании процентной записи за 100 % берется высота шрифта.

В следующем примере объявлены два идентификатора (ID селектора), которые имеют уникальные имена «withLineHeight» и «withFont». В первом идентификаторе устанавливается значение интерлиньяжа 35 пикселей с помощью свойства `line-height` (строка 7), во втором — интерлиньяж 14 пунктов с помощью свойства `font` (строка 8). Следует обратить внимание, что размер шрифта (в примере равен 12 пунктам) и значение интерлиньяжа пишутся через слеш (знак «/»). На рис. 3.45 показано выполнение данного примера в браузере.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Интерлиньяж</title>
6     <style>
7       #withLineHeight { line-height: 35px; }
8       #withFont { font: 12pt/14pt sans-serif; }
9     </style>
10  </head>
```

```
11 <body>
12 <p id="withLineHeight">
13     Известно, что человек имеет способность погрузиться
14     весь в один предмет, какой бы он ни казался
15     ничтожный.
16     И известно, что нет такого ничтожного предмета,
17     который бы при сосредоточенном внимании, обращенном
18     на него, не разросся до бесконечности.
19 </p>
20
21 <p id="withFont">
22     Предмет, в который погрузилась вполне Наташа, –
23     была семья, то есть муж, которого надо было держать
24     так, чтобы он нераздельно принадлежал ей, дому, –
25     и дети, которых надо было носить, рожать, кормить,
26     воспитывать.
27 </p>
28 </body>
29 </html>
```

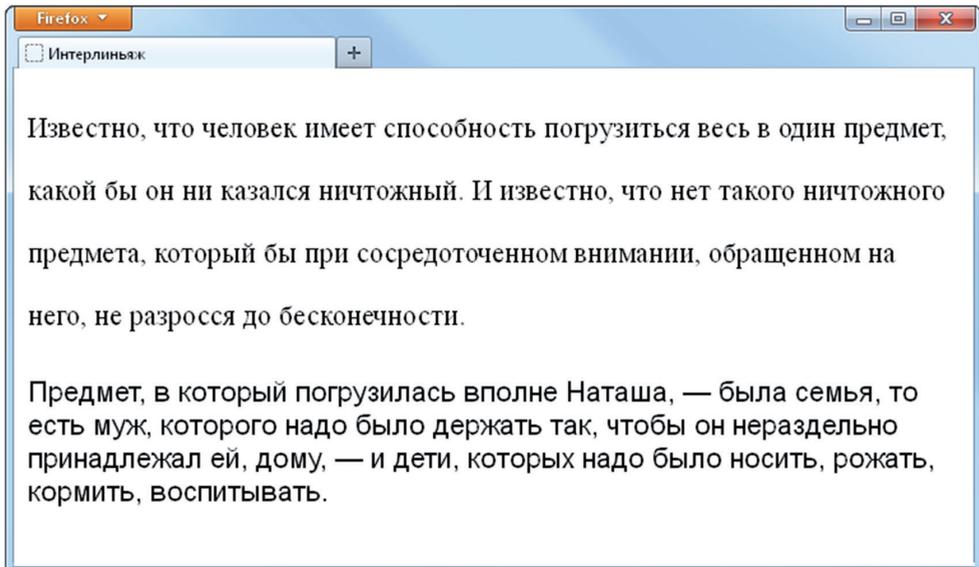


Рис. 3.45. Интерлиньяж 35 пикселей и 14 пунктов для абзацев текста веб-страницы

3.2.5. Заголовки текста

Рубриками называют **заголовки частей текста** (заглавия произведений, внутренние заголовки их подразделов и пр.), а также сами выделенные заголовками части (главы, параграфы и т. д.)

Благодаря различному расположению заголовков относительно текста, их рисунку, начертанию и размеру шрифта читатель видит, какой заголовок и его подраздел старший, а какой — младший, подчиненный и какие заголовки и подразделы равны по значимости. Это помогает читателю точно ориентироваться в структуре и композиции всего текста и воспринимать соотносительное значение каждого заголовка и его подраздела. К **основным функциям** заголовков относятся:

1. Организация, направление и облегчение чтения:
 - прерывая текст, заголовки заставляют читателя невольно остановиться и осмыслить прочитанное перед тем, как приступить к чтению нового подраздела;
 - раскрывая тему последующего текста, заголовки подготавливают читателя к его восприятию;
 - заголовки создают благоприятные условия для чтения выборочно, когда читателю нужны материалы только одной или нескольких подтем.
2. Углубленное понимание прочитанного, т. к. заголовки наглядно раскрывают строение текста и соотносят значение заголовков к их подразделам.
3. Усиление справочной функции, облегчение и упрощение поиска частей текста.

Классификация заголовков достаточно обширна и делится на различные категории. Ниже собраны основные виды заголовков по степени их содержательности, форме и составу:

- Тематические заголовки** состоят из слова или словосочетания, выражающих тему (содержание) той части текста, к которой они относятся.
- Нумерационные и литерные заголовки** состоят из цифрового номера или буквы (в порядке алфавита). Такие заголовки наименее содержательные, поскольку используются только для членения и соподчинения текстовых блоков.

- **Заголовки с подзаголовками.** Тематические заголовки и все заголовки, в которые входит тематическая часть, могут сопровождаться тематическими подзаголовками, когда необходимо конкретизировать заголовок, привести датировку и т. п. Подзаголовок подключается к заголовку либо как текст в круглых скобках, либо как самостоятельная часть, графически отличающаяся от заголовка.

В одном издании могут использоваться как заголовки только одного из этих видов или их сочетания, так и комбинированные — для каждой ступени свой вид (рис. 3.46).

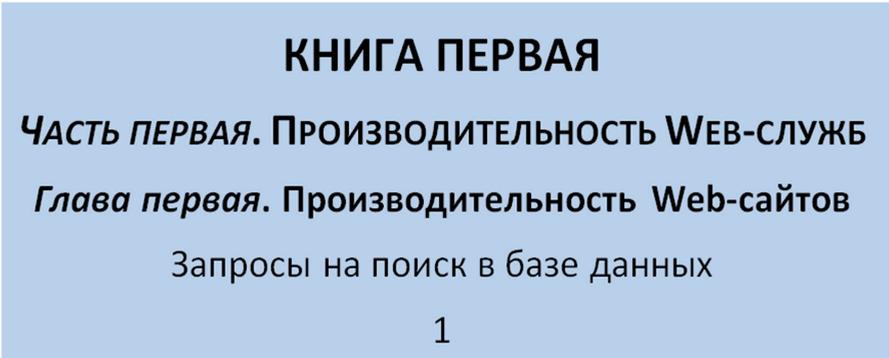
- 
- The diagram illustrates four levels of heading hierarchy for a book, labeled a) through d). The text is centered within a light blue rectangular background.
- a) **КНИГА ПЕРВАЯ**
 - б) **Часть первая. Производительность WEB-служб**
 - в) **Глава первая. Производительность Web-сайтов**
 - г) **Запросы на поиск в базе данных**
 - д) **1**

Рис. 3.46. Различные виды заголовков для одного издания:

а – родо-нумерационный заголовок; *б*, *в* – родо-нумерационно-тематический заголовок; *г* – тематический заголовок; *д* – нумерационный заголовок

HTML предлагает **шесть заголовков разного уровня**, которые показывают относительную важность текста, расположенного после заголовка. Эти заголовки — содержимое тегов `<h1>`, ..., `<h6>`. Элемент `h1` представляет собой наиболее важный заголовок первого уровня, а элемент `h6` служит для обозначения заголовка шестого уровня и является наименее значительным.

По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги от `<h1>` до `<h6>` относятся к блочным элементам: всегда начинаются с новой строки, и после закрытия тега осуществляется перенос на следующую строку. Браузеры отображают тексты заголовков в полужирном начертании. Кроме того, текст заголовков отбивается сверху и снизу, т. е. перед заголовком и после него добавляется пустое пространство.

Основной атрибут, применяемый к тегам заголовков, — `align`, который определяет выравнивание заголовка (рис. 3.47).

Заголовки в веб-странице также можно установить с помощью стилей: как минимум, задать большой размер, отбивку и выравнивание для любого текста, тем самым сделав его заголовком. Однако **для заголовков лучше всегда использовать элементы `h1`, ..., `h6`**, поскольку поисковые системы лучше индексируют текст в заголовках (фактически находят текст по тегам для заголовков) и тем самым повышают рейтинг сайта. При этом ничто не мешает использовать стили CSS для определения правил форматирования (цвет, шрифт, размер, отступы и т. п.) у селекторов тегов `h1`, ..., `h6`.

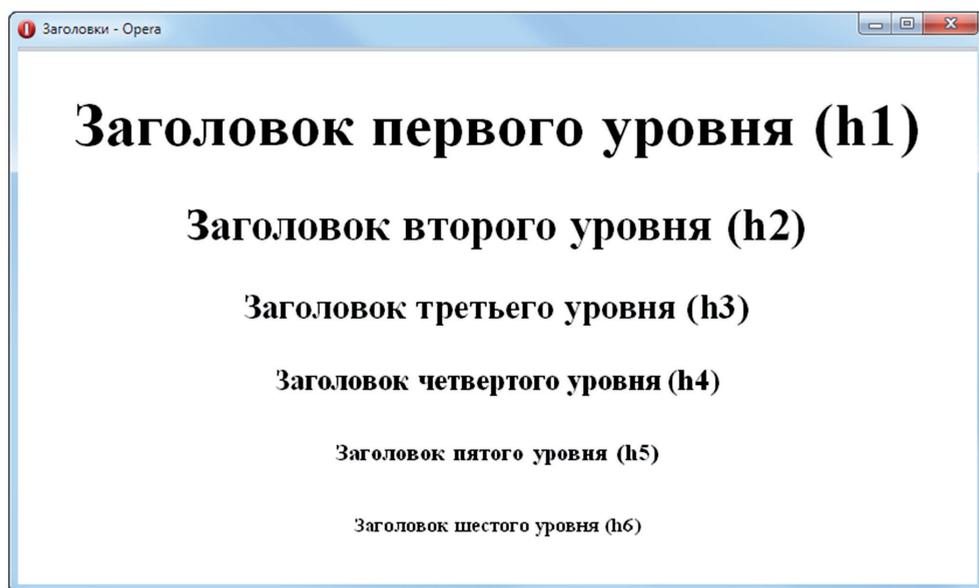


Рис. 3.47. Заголовки от первого до шестого уровня с выравниванием по центру

При работе с заголовками нужно следовать девизу: «На каждую веб-страницу — только по одному заголовку первого уровня (`h1`)». Это и будет заголовок целого документа. Обязательно нужно соблюдать иерархию заголовков. В документе не может быть заголовков третьего уровня (`h3`), если до этого не был использован заголовок второго уровня (`h2`). Иными словами, если провести аналогию с книгой, то можно сказать следующее: `h1` — это заголовок целой книги; `h2` — заголовок части; `h3` — главы; и т. д.

Заголовки мельче четвертого уровня (h5 и h6) нужно стараться использовать только в случае крайней необходимости.

3.2.6. Выключка и отступ

Выключка (англ. *justification*) — выравнивание строк текста относительно одного или двух его боковых краев (полей). Строки могут быть выровнены по левому краю при неровном правом (выключка влево), по правому краю при неровном левом (выключка вправо) или по обоим краям (полная выключка). Если строки выровнены таким образом, что середина каждой строки совпадает с центральной осью текстового блока или колонки набора при неровных боковых краях, то это выключка по центру (рис. 3.48).

При выборе полной выключки текста перестраиваются все атрибуты управления длиной строки: ширина текстового блока делится на количество символов в строке, а эти символы равномерно распределяются по длине строки, причем большая часть избыточных интервалов добавляется к межсловным пробелам. В результате первая буква строки выравнивается по левому, а последняя — по правому краю колонки (даже в том случае, если в строке всего две буквы), а остальные распределяются между ними. Читать такой текст трудно (см. рис. 3.48), поэтому полная выключка используется для сильных спецэффектов. Однако эти неудобства устраняются расстановкой переноса слов в тексте (так называемый «мягкий перенос»), приводящей к уменьшенной и равномерной ширине межсловных пробелов.

Выключка в веб-текстах задается с помощью стилевого свойства `text-align` со следующими значениями:

- `left` — выключка по левому краю (используется браузерами по умолчанию);
- `right` — выключка по правому краю;
- `center` — выключка по центру;
- `justify` — выравнивание текста по ширине.

При выравнивании по значению `justify` каждая строка растягивается на всю ширину блока так, чтобы текст выглядел ровно по краям. Однако при этом возможно появление «широких», мешающих читателям комфортно воспринимать текст междусловных пробелов, так как в браузерах отсутствует расстановка **мягких переносов** (мягкий перенос

Качество обслуживания (Quality-of-Service, QoS) Web-служб, предоставляемых поставщиками, играет решающую роль в привлечении и удержании пользователей. Например, разочарованные потребители покидают многие сайты электронной коммерции и более к ним не возвращаются, что приводит к потере доходов. Каким образом руководство компании определяет качество Web-служб?

Качество обслуживания (Quality-of-Service, QoS) Web-служб, предоставляемых поставщиками, играет решающую роль в привлечении и удержании пользователей. Например, разочарованные потребители покидают многие сайты электронной коммерции и более к ним не возвращаются, что приводит к потере доходов. Каким образом руководство компании определяет качество Web-служб?

Качество обслуживания (Quality-of-Service, QoS) Web-служб, предоставляемых поставщиками, играет решающую роль в привлечении и удержании пользователей. Например, разочарованные потребители покидают многие сайты электронной коммерции и более к ним не возвращаются, что приводит к потере доходов. Каким образом руководство компании определяет качество Web-служб?

Качество обслуживания (Quality-of-Service, QoS) Web-служб, предоставляемых поставщиками, играет решающую роль в привлечении и удержании пользователей. Например, разочарованные потребители покидают многие сайты электронной коммерции и более к ним не возвращаются, что приводит к потере доходов. Каким образом руководство компании определяет качество Web-служб?

Рис. 3.48. Примеры выключки строк одного текстового блока (слева направо, сверху вниз: влево; вправо; полная; по центру)

указывает место разрыва слова или словосочетания, когда оно попадает в конец строки). Поскольку на данный момент и в CSS нет свойств, позволяющих расставлять мягкие переносы, то можно прибегнуть к использованию скриптового языка JavaScript (в частности, можно подключить уже готовый скрипт `hyphenator.js` [40]).

Однако в HTML частичное решение проблемы расстановки мягких переносов возложено на тег `<wbr>`, который указывает браузеру место, где допускается делать перенос строки в тексте. Но при этом ни один браузер при переносе текста не добавляет символ дефиса. При необходимости дефис можно имитировать, заменив тег на символ мягкого переноса `­`. Запись «`­`» в коде HTML называется **мнемоникой**.

Использование мнемоник (подстановок) подробно рассматривается в следующем разделе.

В следующем примере показана установка выключки абзацев текста слева, справа, по центру и по ширине с применением имитации мягких переносов:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Выключка текста и мягкие переносы</title>
6     <style>
7       b { color: DarkGray; }
8     </style>
9   </head>
10  <body>
11    <p style="text-align: left;">
12      <b>Этот абзац имеет выравнивание
13      <i>по левому краю</i>:</b><br>
14      Простейшая формула может быть получена опытным
15      путем через определение соотношения химических
16      элементов в веществе с применением значений атомной
17      массы элементов. Так, простейшая формула воды будет
18       $H_2O$ .</p>
19    <p style="text-align: right;">
20      <b>Этот абзац имеет выравнивание
21      <i>по правому краю</i>:</b><br>
22      Истинная формула может быть получена, если известна
23      молекулярная масса вещества. Истинная формула воды
24       $H_2O$ , что совпадает с простейшей.</p>
25    <p style="text-align: center;">
26      <b>Этот абзац имеет выравнивание
27      <i>по центру</i>:</b><br>
28      В рациональных формулах выделяются группы атомов,
29      характерные для классов химических соединений.
30      Например, для спиртов выделяется группа -ОН.</p>
31    <p style="text-align: justify;">
32      <b>Этот абзац имеет выравнивание
33      <i>по ширине</i>:</b><br>
34      Структурная формула в графическом виде показывает
35      взаимное рас&shy;положение атомов в молекуле.
36      Химические связи между атомами обозна&shy;чаются
37      линиями. Разли&shy;чают двумерные
38      (2D) и трёхмерные (3D) формулы.</p>
39  </body>
40 </html>
```

В этом примере текст, выделенный полужирным начертанием, отображается именованным цветом `darkgray` (темносерый цвет), который определен в строке 7. В последнем абзаце, имеющем выключку по ширине, проставлен символ мягкого переноса в словах «расположение», «обозначаются» и «различают» (строки 35–37). На рис. 3.49 показано выполнение рассматриваемого примера в браузере Firefox. При этом слово «различают» не попало под перенос строки, поэтому дефис в нем не отображается.

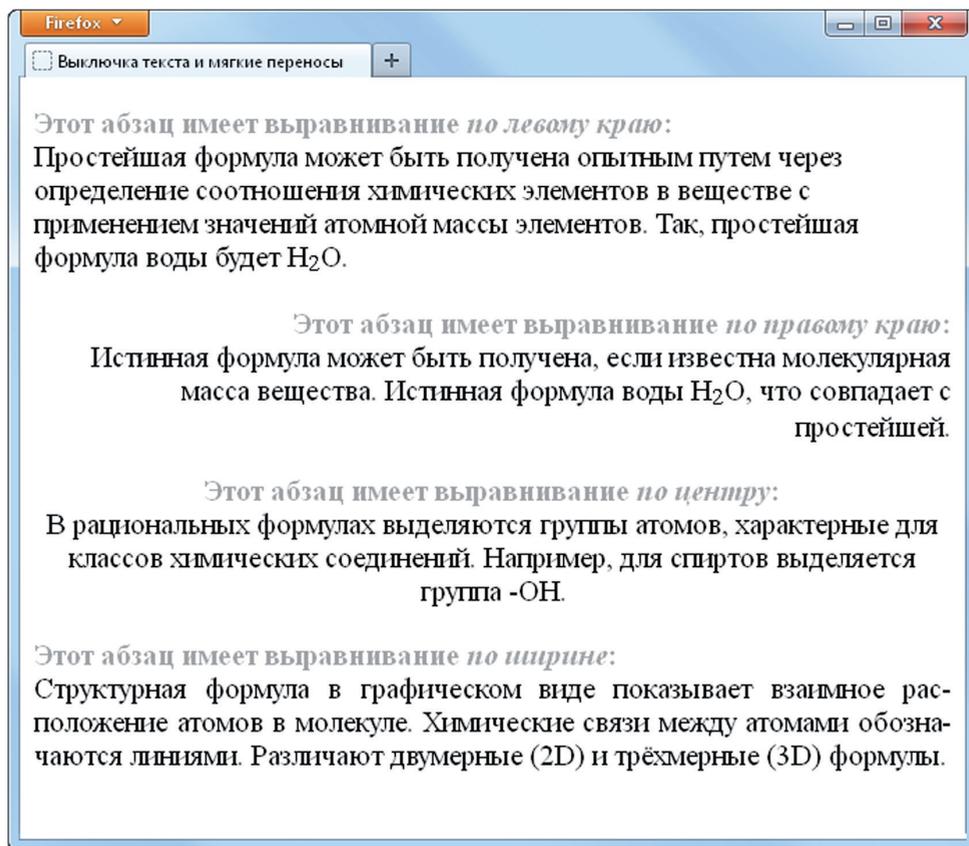


Рис. 3.49. Использование выключки абзацев текста с имитацией мягких переносов

Если для текста задано выравнивание через `text-align` со значением `justify` (выравнивание по ширине), то во избежание появления «длинных» межсловных пробелов можно воспользоваться стилевым свойством `word-spacing`, которое задает интервал между словами.

В качестве значений (которые могут быть и отрицательными) принимаются любые единицы длины, принятые в CSS, кроме процентной записи. Значение `normal` для свойства `text-align` позволяет браузерам устанавливать интервал между словами по умолчанию (т. е. как обычно).

* * * * *

Отступ (англ. *indent*) — отклонение от края текста одной или нескольких подряд идущих строк. Фактически отступы являются указателями для взгляда. Они и сигнализируют о новом абзаце, устанавливая иерархию блоков текста и определяют навигацию по тексту. Иногда их используют для того, чтобы обеспечить пространство для иллюстрации.

Можно выделить четыре основных вида отступов:

- Постоянный отступ** (англ. *running indent*), который сдвигает совокупность строк справа, слева или с обеих сторон;
- Абзацный отступ** (англ. *paragraph indent*), или «Красная строка» (англ. *firstline*), который влияет только на первую строку абзаца;
- Обратный отступ** (англ. *hanging indent*) — вариант постоянного отступа, который начинается после одной или нескольких строк, прижатых к границе левого поля (или, по крайней мере, расположенных левее всех остальных строк абзаца);
- Отступ, **выровненный по точке** (англ. *indent on point*) или по букве (англ. *indent on character*), — вид отступа относительно позиции определенного знака на предшествующей строке.

Когда текст обтекает графический или иной элемент, может получиться каскад строк, у каждой из которых свой собственный отступ.

Отступы текста веб-страницы реализуются, в основном, в виде **постоянных** и **абзацных** отступов. В CSS для этих отступов предусмотрены специальные стилевые свойства.

Поскольку втяжка текста фактически реализуется отступами с разных сторон, то для реализации постоянного отступа для текстового блока нужно использовать стилевое свойство `margin-left` (отступ слева).

В веб-документах **абзацный отступ** помечается специальным сдвигом первой строки текстового блока (например, для абзаца `<p>`), который устанавливает стилевой атрибут `text-indent`. В качестве значений для `text-indent` принимаются любые единицы длины, принятые в CSS: пиксели (px), дюймы (in), пункты (pt) и др. При задании значения в процентах, отступ первой строки вычисляется в зависимости

от ширины текстового блока. Величина отступа подбирается опытным путем исходя из размера шрифта, его типа и ширины блока текста. Для `text-indent` допустимо использовать и отрицательные значения, поэтому в сочетании с положительным значением для свойства `margin-left` можно организовать **обратный отступ** в текстовом блоке.

В следующем примере приведена реализация красной строки, постоянного и обратного отступов для абзацев текста в относительных единицах измерения `em`:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Отступы</title>
6     <style>
7       p {                               /* Выравнивание по ширине */
8         text-align: justify;
9       }
10      .const_margin { /* Постоянный отступ */
11        margin-left: 2em;
12      }
13      .red_line {     /* Отступ первой строки */
14        text-indent: 2em;
15      }
16      .outdent {     /* Обратный отступ */
17        margin-left: 2em;
18        text-indent: -2em;
19      }
20    </style>
21  </head>
22  <body>
23    <p class="red_line">
24      Толки и рассуждения о правах женщин, об отношениях
25      супругов, о свободе и правах их, хотя и не назывались
26      еще, как теперь, вопросами, были тогда точно такие
27      же, как и теперь; но эти вопросы не только не
28      интересовали Наташу, но она решительно не понимала
29      их.
30    </p>
31    <p class="const_margin">
32      Вопросы эти и тогда, как и теперь, существовали
33      только для тех людей, которые в браке видят одно
34      удовольствие, получаемое супругами друг от друга,
35      то есть одно начало брака, а не все его значение,
36      состоящее в семье.
37    </p>
```

```
38 <p class="outdent">
39     Рассуждения эти и теперешние вопросы, подобные
40     вопросам о том, каким образом получить как можно
41     более удовольствия от обеда, тогда, как и теперь,
42     не существуют для людей, для которых цель обеда есть
43     питание и цель супружества – семья.
44 </p>
45 </body>
46 </html>
```

Здесь первый абзац (строки кода 23–30) имеет отступ «красная строка», второй абзац (строки кода 31–37) имеет постоянный отступ, третий (строки кода 38–44) — обратный отступ. Результат выполнения данного примера приведен на рис. 3.50:

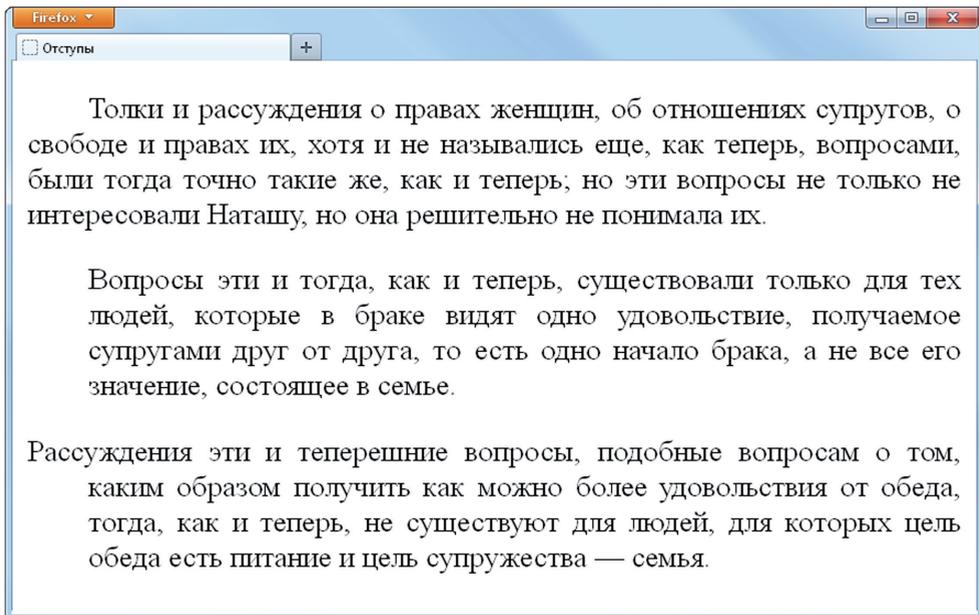


Рис. 3.50. Использование различных отступов для абзацев текста веб-страницы

3.2.7. Линейки

Линейкой в типографике называют вертикальную или горизонтальную линию. Линейки широко применяются для выделений фрагментов текста. Применение линеек характеризуется следующими параметрами:

- направление по отношению к тексту (вертикальное, горизонтальное, наклонное);
- положение по отношению к текстовому блоку (шире или уже текста);
- толщина линии (тонкая, полужирная, жирная);
- вид линии (сплошная, штриховая, штрих-пунктирная, разомкнутая, с изломами);
- состав линии (одинарная, двойная однородная, двойная неоднородная, тройная).

Также линейки различают по функциям:

- *графа* — линейка, разделяющая область текста на столбцы, полосы (или ярусы для адресного графления) и клетки;
- *колонлинейка* — линейка, отделяющая колонэлементы (колонтитулы, колонцифры и пр.) от текста;
- *регель* — линейка, отделяющая (обычно над втяжкой) нижний или средний ярус от основного текста (часто используется для отделения текста сноски от основного текста);
- *линейный средник* (дихазий) — вертикальная линейка, разделяющая колонки набора текста в многоколонном текстовом блоке.

Линейка в веб-типографике представляет собой вертикальную или горизонтальную линию. Одним из способов добавления горизонтальных линеек в веб-страницу является использование тега `<hr>`, который создает горизонтальную линию заданной высоты (параметр `size`) и ширины (параметр `width`). Ширина линейки может указываться в пикселях и процентах (процент от ширины области окна браузера для представления веб-страницы). В качестве размера `size` выступает любое целое положительное число (нулевое или отрицательное значение устанавливает толщину линии в один пиксель; по умолчанию значение `size` составляет 2 пикселя).

Кроме `size` и `width`, в теге `<hr>` определены следующие атрибуты:

- `align` — определяет выравнивание линии (`center` — по центру (значение по умолчанию), `left` — по левому краю, `right` — по правому краю);
- `color` — цвет линии;
- `noshade` — отображение линии без трехмерных эффектов.

Например, выполнение приведенных ниже тегов в браузере будет выглядеть так, как показано на рис. 3.51:

```
1 <hr noshade size="10" width="90%" color=black>  
2 <hr noshade size="8" width="85%">  
3 <hr noshade size="6" width="65%" color=black>  
4 <hr noshade size="3" width="45%">  
5 <hr noshade size="1" width="25%" color=black>
```

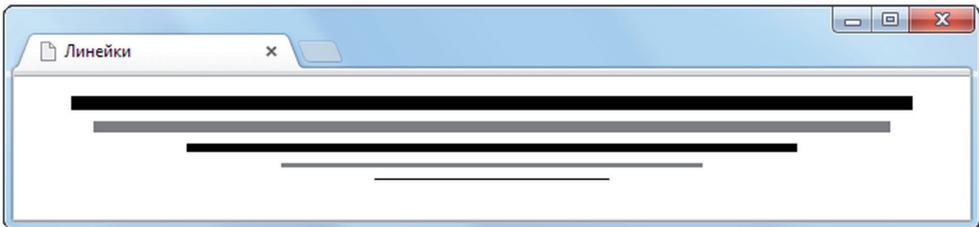


Рис. 3.51. Линейки, созданные при помощи тега `<hr>`

Стили CSS дают бóльшую свободу по созданию линеек и управлению их видом. При этом можно легко устанавливать как горизонтальные, так и вертикальные линии. Для этой цели применяются стилевые атрибуты `border-left`, `border-right`, `border-top` и `border-bottom`, которые, соответственно, задают линию слева, справа, сверху и снизу. Каждый из приведенных атрибутов одновременно определяет толщину линии (`border-width`), ее стиль (`border-style`) и цвет (`border-color`). Указанные атрибуты наследуют свойства универсального атрибута `border`, который позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента:

```
border: [border-width || border-style ||  
        || border-color]
```

Значения `border-width`, `border-style` и `border-color` могут указываться в любом порядке, разделяясь пробелом, — браузер сам определит, какое из них соответствует нужному свойству. Для управления видом линейки предоставляется несколько значений `border-style`, названия и результат действия которых представлен на рис. 3.52.

Например, чтобы добавить сплошную линейку красного цвета слева от абзаца, в стилях достаточно написать:

```
p { border-left: 3px solid #f00; }
```

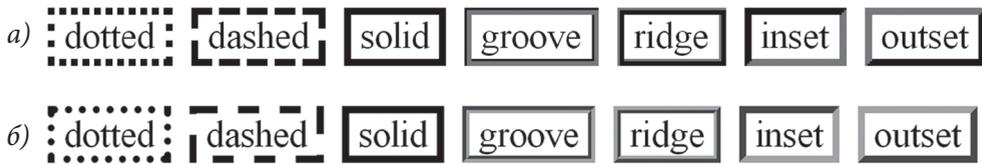


Рис. 3.52. 9.15. Стиль линеек, задаваемых атрибутом `border-style`. В разных браузерах выглядят по-разному (а — в браузере Opera, б — в браузере Firefox)

В качестве значения атрибута `border-left` вначале идет толщина в пикселях (3px), затем ключевое слово, обозначающее сплошную линию (`solid`), и заканчивается цветом линии в шестнадцатеричном формате (`#f00` — красный цвет).

В следующем примере показано добавление вертикальной и горизонтальной линеек к тексту на веб-странице:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>линейки</title>
6      <style>
7        div.leftLine {
8          /* Параметры вертикальной линейки */
9          border-left: 7px double red;
10         margin-left: 20px; /* Сдвиг блока вправо */
11         /* Расстояние между линейкой и текстом */
12         padding-left: 10px;
13       }
14       div.leftLine .bottomLine {
15         /* Параметры горизонтальной линейки */
16         border-bottom: 4px dotted blue;
17         /* Расстояние между линейкой и текстом */
18         padding-bottom: 5px;
19       }
20     </style>
21   </head>
22   <body>
23     <p>
24       идея работы портала – предоставление максимального
25       количества Интернет-сервисов для привлечения
26       наибольшего числа пользователей.
27     </p>
28     <div class="leftLine">

```

```
29 <p class="bottomLine">
30 <strong>Веб-портал для пользователей</strong></p>
31 <p>Сайт в компьютерной сети, который предоставляет
32 пользователю различные интерактивные сервисы
33 (Интернет-сервисы), работающие в рамках этого сайта.
34 Веб-портал может состоять из нескольких сайтов,
35 если они объединены под одним доменным именем.</p>
36 </div>
37 </body>
38 </html>
```

Результат данного примера показан на рис. 3.53. В коде вертикальная линейка отображается двойной сплошной линией, а горизонтальная — одинарной пунктирной. Ширина и высота линеек зависит от размера блоков (определенных тегами `<div>` и `<p>`) и подстраивается под них.

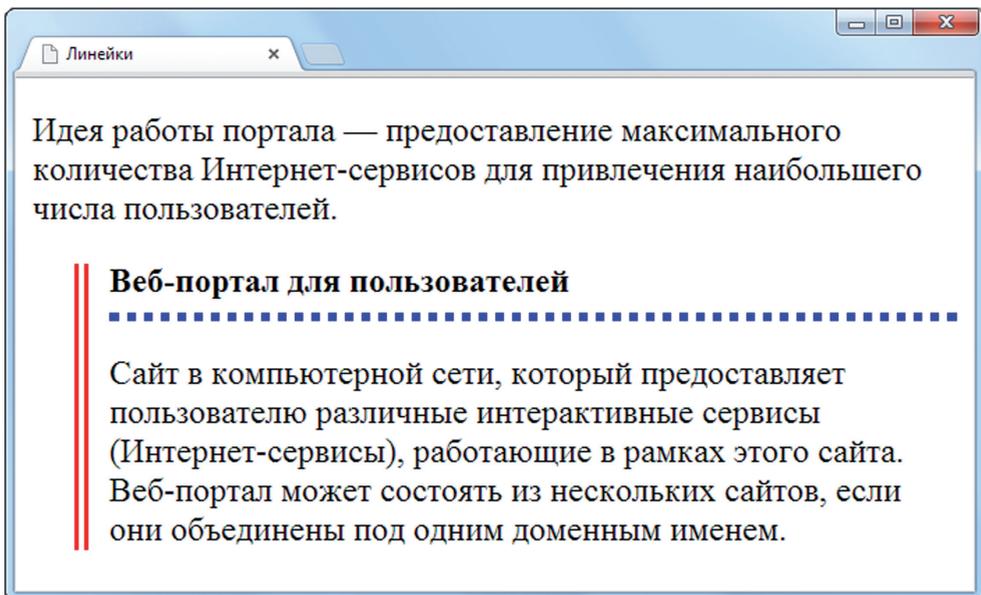


Рис. 3.53. Линейки, созданные с помощью стилей CSS

3.2.8. Способы выделения элементов текста

Выделения в тексте — набор букв, слов, фраз или частей текста отличным от набора основного текста способом. По **размеру выделения** различают **текстовые** (глобальные) и **внутритекстовые** (локальные) выделения.

По способам выделения различают **шрифтовые, нешрифтовые и комбинированные** выделения. Шрифтовое выделение отдельных слов или групп слов (локальное выделение) может быть выполнено следующими основными способами:

- полужирным прямым шрифтом гарнитуры основного текста;
- полужирным прямым шрифтом другой гарнитуры;
- светлым курсивным шрифтом гарнитуры основного текста;
- светлым курсивным шрифтом другой гарнитуры;
- полужирным курсивным шрифтом гарнитуры основного текста;
- полужирным курсивным шрифтом другой гарнитуры;
- капителью;
- светлым прямым шрифтом, набранным в *р а з р я д к у*;
- светлым курсивным шрифтом в *р а з р я д к у*.

При выделении части фразы знаки препинания, следующие за выделением и стоящие перед обычным текстом, набираются основным шрифтом без выделения. При выделении целой фразы знаки препинания (если они относятся к этой фразе) на стыке выделенной целиком фразы и невыделенного текста набираются выделительным шрифтом. Точка, стоящая в конце фразы, выделенной полужирным шрифтом, должна также иметь полужирное начертание.

Абзацы также являются текстовыми выделениями (в отличие от внутритекстовых), если они отличаются от основного текста одним или несколькими атрибутами:

- Шрифтовые способы выделения:
 - шрифтом повышенного кегля;
 - светлым шрифтом другой гарнитуры;
 - полужирным шрифтом своей или другой гарнитуры;
 - светлым курсивом своей гарнитуры;
 - светлым курсивом другой гарнитуры.
- Нешрифтовые способы выделения:
 - левосторонняя втяжка: абзац имеет отступ от левого края полосы набора основного текста;
 - двусторонняя втяжка: абзац имеет отступ и от правого, и от левого края полосы набора основного текста;
 - отчеркивание текста горизонтальными линейками;
 - отчеркивание текста вертикальной линейкой слева;
 - заключение текста в рамку.

К нешрифтовым выделениям отдельных слов или групп слов (локальное выделение) относится **подчеркивание** (реже **надчеркивание** и **перечеркивание**) одной или двумя чертами. При этом подчеркивается только выделенное слово или фраза, без подчеркивания пробелов, отделяющих выделенную часть от остальной. Следует, однако, отметить, что подчеркивание отдельных слов применяется в последнюю очередь, если уже использовано большое количество шрифтовых выделений.

Подчеркивание, надчеркивание и перечеркивание в CSS реализуются с помощью свойства `text-decoration`, добавляющего оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания. Одновременно можно применять несколько параметров, перечисляя их значения через пробел:

```
text-decoration: [ blink | line-through |  
                 | overline | underline ] | none
```

Используемое в `text-decoration` значение:

- `blink` — устанавливает мигающий текст, который периодически (примерно раз в секунду) исчезает, потом вновь появляется на прежнем месте (не поддерживается браузерами Opera и Firefox);
- `line-through` — создает перечеркнутый текст;
- `overline` — выделяет текст надчеркиванием;
- `underline` — устанавливает подчеркнутый текст;
- `none` — отменяет все эффекты (в том числе и подчеркивания у гиперссылок, которые заданы по умолчанию).

Линия, используемая для выделения в `text-decoration`, — сплошная тонкая и принимает цвет текста, к которому применяется. Для использования надчеркивания и подчеркивания другими типами линий можно воспользоваться рассмотренными выше свойствами `border-top` и `border-bottom`.

В следующем коде показаны различные варианты использования свойств `text-decoration`, `border-top` и `border-bottom`. Результат выполнения этого кода приведен на рис. 3.54:

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8">  
5 <title>Ссылки и подчеркивания</title>
```

```
6 <style>
7   /* убирается подчеркивание у посещенных ссылок */
8   a:visited {
9     text-decoration: none;
10  }
11
12   /* При наведении указателя мыши на ссылку */
13   a:hover {
14     /* убирается обычное подчеркивание */
15     text-decoration: none;
16     /* Двойная синяя линия под текстом */
17     border-bottom: 3px double red;
18   }
19
20   /* пунктирная серая линия под текстом */
21   .underLine {
22     border-bottom: 2px dashed gray;
23   }
24
25   /* Точечная линия над текстом */
26   .overLine {
27     border-top: 3px dotted firebrick;
28   }
29 </style>
30 </head>
31 <body>
32 <div><a href="http://scholar.google.com/">
33   Ссылка на Google Scholar</a></div>
34 <div><a href="http://scholar.google.com/">
35   Ссылка на Google Scholar</a></div>
36 <div><a href="http://scholar.google.com/">
37   Ссылка на Google Scholar</a></div>
38 <p>
39   <span class="overLine">Библиографические ссылки
40   Академии Google (Google Scholar)</span> позволяют
41   авторам следить за цитированием
42   <span style="text-decoration: line-through">чужих
43   публикаций</span> своих статей. Вы можете узнать,
44   кто ссылается на ваши публикации, создать диаграмму
45   цитирования и вычислить показатели этого процесса.
46   <span class="underLine">Кроме того, можно сделать
47   свой профиль общедоступным, чтобы он отображался
48   в результатах поиска по вашему имени.</span>
49 </p>
50 </body>
51 </html>
```

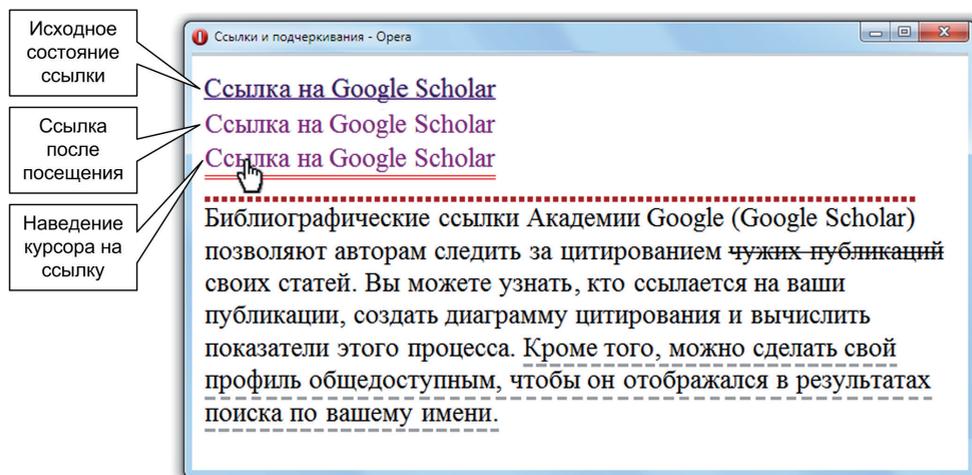


Рис. 3.54. Подчеркивание, надчеркивание и перечеркивание текста с помощью стилей CSS

3.2.9. Сервисы автоматического форматирования веб-текстов

Сервисы автоматического форматирования веб-текстов, так называемые **веб-типографы**, предназначены для подготовки текстов к размещению на веб-страницах т. е. приведения этих текстов к «правильному» виду с точки зрения веб-типографики и, в некоторых случаях, — орфографии и пунктуации, в котором их можно размещать в коде HTML.

В Интернете постоянно появляются новые «типографы», но все они обладают похожим функционалом. Поэтому в рамках данной книги, мы ограничимся знакомством только с одним из них — с «**Типографом**» студии **Артемия Лебедева** [41].

Мотивацию использования своего «Типографа» студия Артемия Лебедева объясняет так: «В жизни каждого порядочного веб-мастера наступает момент, когда он всерьез задумывается над экранной типографикой. С одной стороны, ему хочется, чтобы сверстанный текст хорошо смотрелся в окне браузера. А для этого нужно вместо знаков дюйма поставить нормальные кавычки («елочки» и „лапки“), поубивать лишние пробелы, в нужных местах поменять дефисы на тире, неразрывным пробелом «привязать» все короткие союзы и предлоги к следующим

за ними словам и произвести кучу других операций. С другой стороны, очень не хочется расставлять все эти значки, кавычки и неразрывные пробелы вручную» [42].

На рис. 3.55 приведен онлайн-сервис «Типограф» студии Артемия Лебедева с введенным текстом, требующим форматирования. В тексте преднамеренно нарушены правила типографики: добавлены лишние пробелы или, наоборот, убраны вовсе; заменены тире на дефисы; вставлены два подряд идущих дефиса вместо тире; и т. п. Кроме того, в тексте присутствуют слова на иностранном языке, в котором встречаются символы с диакритическими знаками.

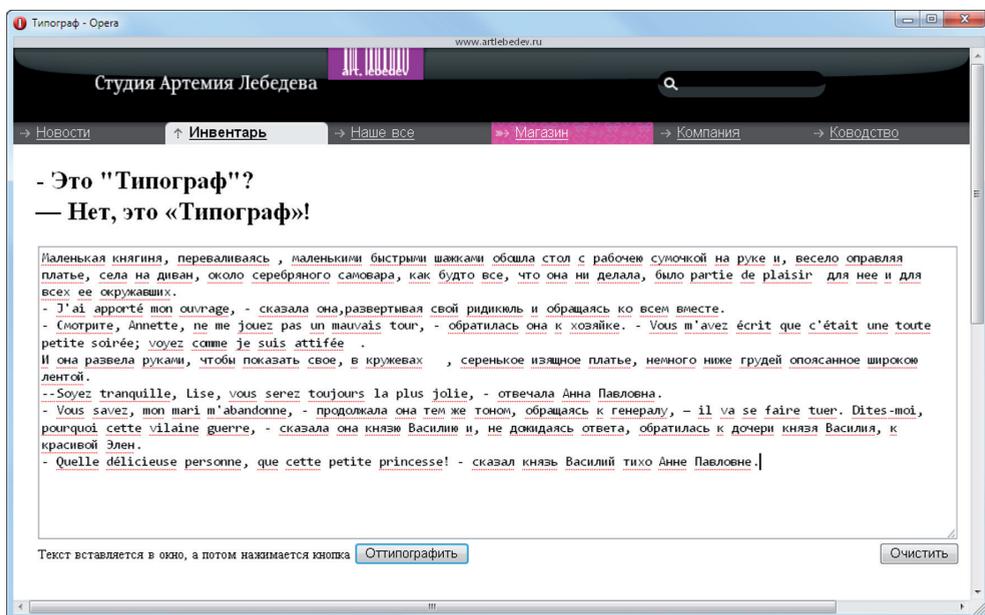


Рис. 3.55. «Типограф» студии Артемия Лебедева с текстом, требующим форматирования

На рис. 3.56 приведен результат выполнения форматирования текста рассматриваемым веб-типографом. При этом в окне ввода текст преобразовывается следующим образом:

1. Встраиваются теги HTML: в данном случае теги абзаца <p> и переноса строки
 (настраиваемая опция).
2. Дефисы меняются на тире (там где это нужно) и обрамляются пробелами.

3. Вставляются мнемоники неразрывных пробелов по правилам переноса строк, включая тире и короткие союзы и предлоги.
4. Символы с диакритическими знаками заменяются соответствующими мнемониками.

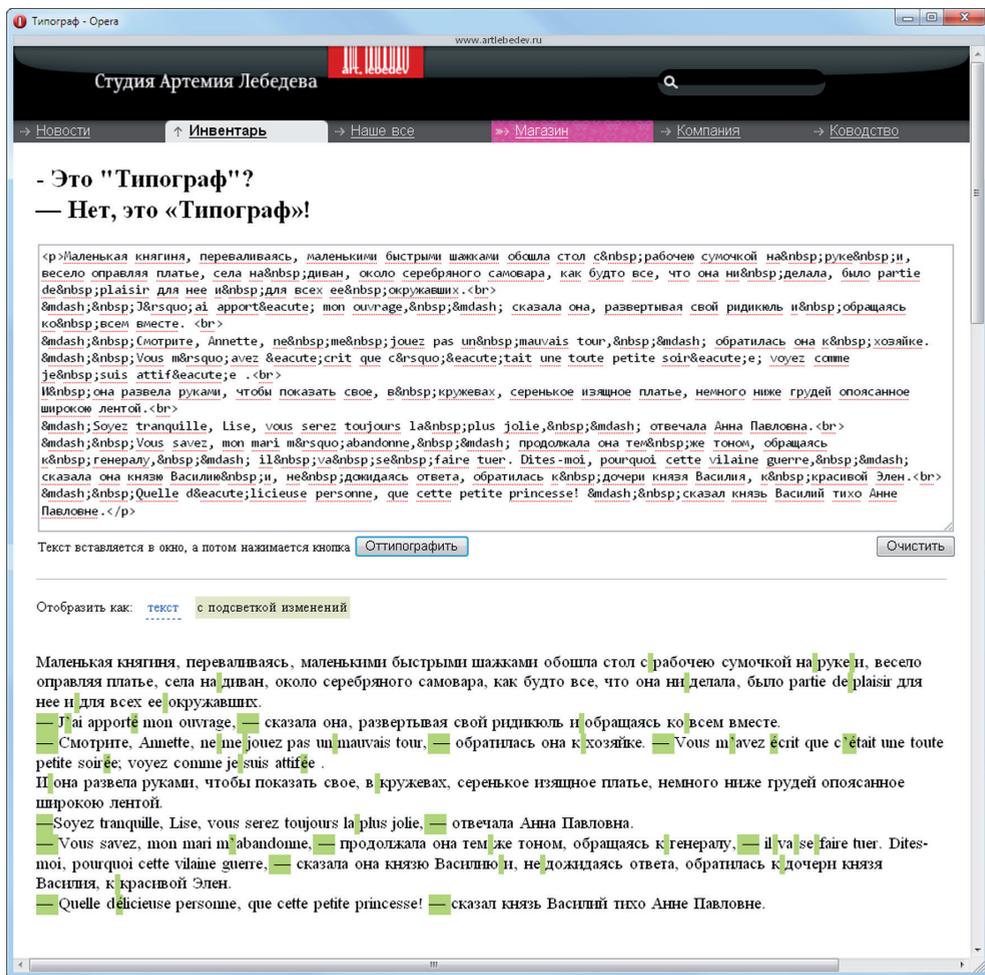


Рис. 3.56. Результат форматирования текста в веб-типографе студии Артемия Лебедева. Отформатированный материал отображается как код HTML и как текст в браузере (с включенной подсветкой изменений)

Кроме этого, «Типограф» умеет обрабатывать не только «чистый» текст, а и текст, представляющий собой коды HTML, CSS и JavaScript.

Эффективная работа данного «Типографа» зависит от того, как он настроен под определенные требования. Пользователь может выбрать

варианты отображения символов, установить символы кавычек и их вложенность, назначить теги для абзацев и подключить тег переноса строк.

С недавнего времени авторы «Типографа» дополнили его веб-сервисом, к которому могут обращаться клиентские приложения, написанные на любом языке. Примеры для разных языков программирования можно загрузить с сайта студии Артемия Лебедева [43].

Какому веб-типографу доверить свои тексты — каждый решает сам. Но положившись на один из них, не следует забывать, что веб-типограф — не панацея. Насколько бы хорошим он не был, ему не под силу исправить абсолютно все ошибки. Поэтому нужно обязательно перечитать текст еще раз, после форматирования веб-типографом и обязательно просмотреть полученный результат в браузере, в этом случае высока вероятность обнаружения ошибок форматирования, которые были пропущены веб-типографом. Кроме того, рассмотренный типограф поддерживает повторное типографирование текста: можно «доводить» форматирование текста прямо в поле ввода веб-типографа.

Контрольные вопросы

1. Что такое типографика и веб-типографика?
2. В чем отличие оформления текстов для сайта от подготовки текстов для печатной продукции?
3. Что такое текст, текстовый файл и текстовые данные?
4. Что подразумевается под понятием «форматирование текста»?
5. На какие категории делится текст по своей структуре?
6. Что такое абзац текста и какой тег HTML определяет текстовый абзац?
7. Что такое колонцифра и колонтитул?
8. Дайте определение категориям стандартных, нестандартных и безопасных шрифтов. Чем они отличаются друг от друга?
9. В каких единицах измерения задаются абсолютные размеры шрифта?

10. В каких единицах измерения задаются относительные размеры шрифта?
11. Какими значениями свойства `font-style` можно установить обычное, курсивное и наклонное шрифтовое начертание?
12. Какими значениями свойства `font-weight` можно установить полужирное, жирное, светлое и нормальное шрифтовое начертание?
13. Какие стилевые свойства текста можно задавать с помощью собирательного свойства `font`? В каком порядке они следуют?
14. Что такое капитель? Каким стилевым свойством CSS можно установить капитель для веб-текста?
15. Какими значениями свойства `text-transform` можно преобразовать символы веб-текста в строчные или прописные?
16. Что такое втяжка и отбивка текста? С помощью каких стилевых свойств можно управлять втяжкой и отбивкой текстов на веб-странице?
17. Что такое интерлиньяж? Как задавать интерлиньяж для текстов сайта с помощью стилевых свойств CSS?
18. Какие основные функции у заголовков текста?
19. Какие бывают заголовки по степени их содержательности, форме и составу?
20. Какими тегами HTML задаются заголовки на веб-страницах?
21. Что такое выключка и отступ? Какими стилевыми свойствами можно задавать различные виды выключки и отступа абзацев в веб-текстах?
22. Что такое текстовые линейки? Какой тег HTML размещает на веб-странице горизонтальную линейку?
23. Что такое шрифтовые и нешрифтовые способы текстовых выделений?
24. Какими стилями CSS можно реализовать подчеркивание, надчеркивание и перечеркивание веб-текста?
25. Что такое веб-типографы? Опишите принцип работы с веб-типографами на примере «Типографа» студии Артемия Лебедева.

3.3. Специальные символы в веб-текстах

Без использования типографских знаков невозможно представить ни один текст, поскольку в тексте, кроме буквенно-цифровых символов, присутствуют различные элементы. Эти элементы, во-первых, способствуют более **комфортному восприятию и пониманию текста** (наряду с текстовыми выделениями), во-вторых, **выполняют вспомогательные функции** разделения смысловых отрезков текста и, в-третьих, синтаксически оформляют текст.

В электронных текстах, естественно («какой же электронный текст не мечтает быть прочитанным!»), типографские знаки сохранили свою актуальность. Для одной части знаков назначены символы, которые можно ввести с клавиатуры: точка, запятая, двоеточие, дефис и др. Другая часть — это специальные символы («живущие» в наборе символов — таблице, задающей кодировку множества символов алфавита), которые вставляются в текст определенным образом.

3.3.1. Использование специальных символов

При компьютерном наборе к **специальным символам** относят все символы, которые невозможно ввести с клавиатуры. Их можно вставить в документ с помощью специальной **таблицы символов** (например, таблицы Юникода) или нажатия сочетания клавиш на клавиатуре (для Windows это обычно <Alt>+<код>, где <код> — цифровой код символа).

Специальные символы в HTML называют **мнемониками**, **подстановками**, **escape-последовательностями** (управляющими последовательностями) или **сущностями** (англ. *entities*). Мнемоники ссылаются на символ из соответствующего набора, который определен стандартом ISO 8859-1 [44]. Их можно вводить в код HTML одним из трех способов:

- **&имя;** — именованная мнемоника, объявленная по специальному (зарезервированному) имени;
- **&#dddd;** — десятичная числовая мнемоника, возвращающая специальный символ по коду dddd — коду символа в Юникоде в десятичной системе счисления;

- **&#xhhhh;** — шестнадцатеричная числовая мнемоника, возвращающая специальный символ по шестнадцатеричному коду hhhh.

Между мнемониками `&#dddd;` и `&#xhhhh;` справедливо равенство: $(dddd)_{16} = hhhh$ и $(hhhh)_{10} = dddd$, где коэффициенты 16 и 10 определяют перевод чисел в соответствующие системы счисления. Например, `©` и `©` соответствуют одному знаку (©) с кодом Юникода U+00A9 — знаку «авторского права», у которого также есть именованная мнемоника `©`.

Далеко не для всех символов существуют именованные мнемоники. Существует более универсальный способ вставить в документ произвольный символ из диапазона Юникода. Для этого надо узнать десятичный или шестнадцатеричный адрес нужного символа в таблице Юникода, и сделать из него числовую подстановку по принципу:

`&#<10-чный_адрес_символа>;` или

`&#x<16-ричный_адрес_символа>;`.

В следующем примере показан код с введенными в файл символами HTML из таблицы 3.2, а результат отображения этих символов в браузере показан на рис. 3.57.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="windows-1251">
5 <title>Псевдографические символы</title>
6 </head>
7 <body>
8 <p>
9 &#8224; &#8658; &#9730; &#9742; &#9757;
10 &#9762; &#9773; &#9822; &#9834;
11 </p>
12 </body>
13 </html>
```

Использование мнемоник в вебстранице иногда приводит к проблемам при просмотре страницы в браузере. Чтобы браузер мог корректно отобразить нужный символ, этот символ должен присутствовать хотя бы в одном из шрифтов, установленных на компьютере пользователя.

Таблица 3.2. Примеры некоторых графических символов

Юникод	Символ	Десятичная мнемоника
U+2020	†	†
U+21D2	⇒	⇒
U+2602	☂	☂
U+260E	☎	☎
U+261D	☞	☝
U+2622	☢	☢
U+262D	☭	☭
U+265E	🐎	♞
U+266A	🎵	♪

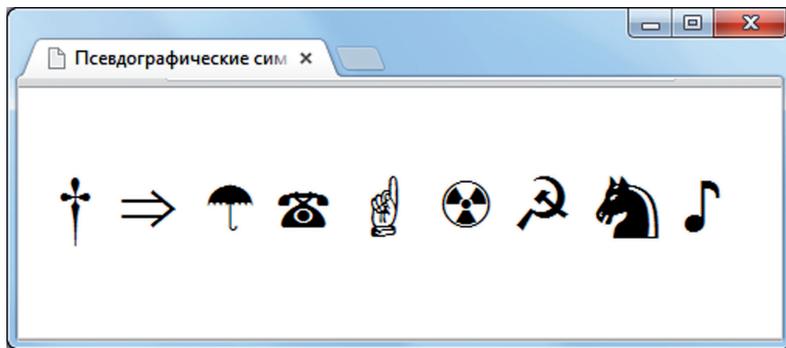


Рис. 3.57. Отображение специальных символов в браузере Google Chrome

3.3.2. Типографские знаки

Типографские знаки можно условно разделить на три категории:

- Диакритические знаки** — элементы письменности, модифицирующие начертание знаков и обычно набираемые отдельно (табл. 3.3).
- Знаки препинания** — элементы письменности для выделения смысловых отрезков текста, предложений, словосочетаний, слов или частей слова. Выполняют ряд функций, в том числе указывают на грамматические и логические отношения между словами, на коммуникативный тип предложения, его эмоциональную окраску и законченность.

- **Лигатуры** — знаки любой системы письма или фонетической транскрипции, образованные соединением двух и более единиц письменной речи (графем). Примеры лигатур: **Æ, Œ, Њ, fb, љ**.

Таблица 3.3. Примеры диакритических знаков

Название	Юникод	Мнемоника	Знак	Примеры
Акут (ударение)	U+0301	<i>&acute;</i> ; или <i>&#769;</i>	´	á É ú ó í
Гравис (обратное ударение)	U+0060	<i>&grave;</i> ; или <i>&#96;</i>	`	À à Ò ò ù È è
Кратка	U+02B8	<i>&#728;</i>	ˇ	Ŏ ŏ Ě ě
Гачек	U+02C7	<i>&caron;</i> или <i>&#711;</i>	ˇ	Ě ě Ď ě š š ě
Умляют	U+00A8	<i>&uml;</i> ; или <i>&#168;</i>	¨	Û ü ä ë Ÿ
Макрон	U+0304	<i>&#772;</i>	¯	Ē ē Ō ō Ī ī

В современной письменности русского языка можно также найти **диакритические знаки**:

- *ударение* — ставится только в редких случаях и не образует новых букв;
- *умляют* над «е» — образует новую букву «ё», но часто опускается (в украинском языке аналогичным образом образована буква «ї», при этом умляют никогда не опускается);
- *кратка* над «и» — образует новую букву «й» и никогда не опускается;
- *надчеркивание* и *подчеркивание* одинаково выглядящих при письме от руки букв «т» (*t*) и «ш» (*ш*);
- *хвостик* у «ш» — является неотъемлемой частью буквы «щ», но может быть воспринят как диакритический знак при формальном анализе и сравнении букв алфавита.

Символ с диакритическим знаком можно получить, если к исходному символу добавить код диакритического знака. Например, конструкция в коде HTML `ē` отобразится в браузере как символ «ē».

Знаки препинания используются в тексте повсеместно. Но помимо привычных точки, запятой, двоеточия, скобок, тире и др., существуют также специальные знаки препинания, которые могут быть использованы и в текстах веб-страниц. Некоторые из таких знаков препинания

сведены в таблицу 3.4 и относятся к знакам основной типографики, знакам интеллектуальной собственности и знакам «редкой» типографики.

Таблица 3.4. Специальные знаки препинания

Название	Знак	Юникод	Мнемоники		
Основная типографика					
Маркер списка (буллит)	•	U+2022	<i>&bull;</i>	<i>&#8226;</i>	<i>&#x2022;</i>
Знак градуса	°	U+00B0	<i>&deg;</i>	<i>&#176;</i>	<i>&#xB0;</i>
Многоточие	...	U+2026	<i>&hellip;</i>	<i>&#8230;</i>	<i>&#x2026;</i>
Знак деления (промежуток)	÷	U+00F7	<i>&divide;</i>	<i>&#247;</i>	<i>&#xF7;</i>
Промилле	‰	U+2030	–	<i>&#8240;</i>	<i>&#x2030;</i>
Абзац	¶	U+00B6	<i>&para;</i>	<i>&#182;</i>	<i>&#xB6;</i>
Знак параграфа	§	U+00A7	<i>&sect;</i>	<i>&#167;</i>	<i>&#xA7;</i>
Интеллектуальная собственность					
Знак охраны авторского права	©	U+00A9	<i>&copy;</i>	<i>&#169;</i>	<i>&#xA9;</i>
Знак охраны смежных прав	®	U+00AE	<i>&reg;</i>	<i>&#174;</i>	<i>&#xAE;</i>
Товарный знак	™	U+2122	<i>&trade;</i>	<i>&#8482;</i>	<i>&#x2122;</i>
«Редкая» типографика					
Астеризм	**	U+2042	–	<i>&#8258;</i>	<i>&#x2042;</i>
Интерробанг	‽	U+203D	–	<i>&#8253;</i>	<i>&#x203D;</i>
Иронический знак	‽	U+2E3E	–	<i>&#11822;</i>	<i>&#x2E3E;</i>

Применение знаков «редкой» типографики довольно специфическое и, как следует из названия, редко встречающееся. Например,

- *астеризм* применяется в типографике для привлечения внимания к повествованию или отделению подразделов либо эпизодов в тексте;
- *интерробанг* предназначается для обозначения риторических вопросов (в основном в англоязычных текстах и для американской типографики), большинство которых также является восклицаниями;
- *иронический знак* — знак препинания, выражающий иронию.

3.3.3. Пробельные символы

Пробел (англ. *space*) — функционально принадлежащий к знакам препинания символ, задающий интервал между буквами и обозначающий границы слов, — заслуживает отдельного рассмотрения.

Пробел является очень популярным символом, о чем, свидетельствует как высокий «частотный показатель» встречаемости пробельного символа в текстах, так и размер клавиши пробела на клавиатуре. Последний параметр обусловлен удобством (для нажатия большими пальцами обеих рук, не меняя их основного положения на клавиатуре) и частотой использования.

Говоря о пробеле как о символе, все же необходимо упомянуть и другие связанные с ним понятия. Ведь под пробелом также понимают любое пустое (незаполненное) место в тексте — рукописном, печатном или электронном (слово «пробел» в этом смысле применялось в предыдущих главах, посвященных форматированию текста).

Итак, по отношению к незаполненным участкам текста пробелы различают на:

- **спусковые** и **концевые** — большие вертикальные пропуски в полосе издания и пропуски в конце полосы;
- **абзацные** и **концевые абзацные** — отступы перед первой строкой и после последней строки абзаца от вертикальных границ текстового блока;
- **межстрочные** — промежутки между строками текста;
- **междусловные** — отступы между соседними словами в строке;
- **межбуквенные** — промежутки между буквами в слове.

Далее речь пойдет о пробельных символах, являющихся по определению междусловными пробелами и по функциональности — знаками препинания.

В таблице Юникода предусмотрен ряд символов для пробелов, которые сведены в табл. 3.5.

Обычный и неразрывный междусловные пробелы входят в состав шрифтов любой гарнитуры. Они способны «растягиваться» при выключке строк текста по ширине и правильно отображаются (в отношении пропорциональности к ширине символа) всеми текстовыми процессорами и браузерами (если не считать того факта, что у некоторых

Также начинающие веб-разработчики предпочитают использовать тег `<nobr>...</nobr>` (уведомляет браузер отображать текст без переносов), что тоже неграмотно, поскольку этого тега нет в официальной спецификации HTML, а его поддержка — частная инициатива разработчиков некоторых браузеров.

3.3.4. Дефис, тире и прочие черточки

При наборе современных компьютерных текстов часто возникают вопросы правильной расстановки дефиса и тире (длинного и/или короткого). Однако авторы текстов (или наборщики/верстальщики, а также веб-разработчики) подобные вопросы часто оставляют без внимания, довольствуясь присутствующим на клавиатуре непонятным символом-черточкой «-». Одни по ошибке считают, что это дефис, другие — что это знак минуса. В результате в электронных текстах, в частности, в текстах веб-страниц, налицо неграмотность причастных к этому людей. Такая ситуация прослеживается и в печатных изданиях — газетах, книгах, журналах, — что неудивительно, ведь современный печатный текст изначально набирался и форматировался на компьютере, т. е. ранее был электронным.

В русском языке определены как понятия дефиса и тире, так и правила (а также сфера) их применения. При этом дефис считается орфографическим знаком, а тире — знаком пунктуации, а точнее — одним из знаков препинания.

Чтобы определиться, когда и где использовать правильный знак-черточку, нужно рассмотреть весь арсенал подобных знаков. В таблице Юникода их **девять (!)**, и все они сведены в табл. 3.6.

Дефис-минус (который также называют дефисоминусом) — это тот самый символ, который вводится с клавиатуры. Это уже не дефис, но еще не минус (и ни в коем случае не тире!), а что-то среднее между ними. Использовать этот символ рекомендуется тогда, когда нет технической возможности употребить настоящий дефис или настоящий минус. Однако из-за удобства набора этот знак повсеместно встречается в веб-текстах.

Настоящего **дефиса** на клавиатуре нет и поэтому повсеместное его употребление представляется крайне проблематичным. На текущий момент выходом из этой ситуации может послужить использование

Таблица 3.6. Девять знаков-черточек

Название	Символ	Юникод	Мнемоника
Дефис-минус (Hyphen-minus)	-	U+002D	-
Дефис (Hyphen)	—	U+2010	‐
Знак минуса (Minus Sign)	—	U+2212	−
Цифровая черточка (Figure Dash)	—	U+2012	‒
Мягкий дефис (Soft Hyphen, символ переноса)	-	U+00AD	­
Черточный буллит (Hyphen Bullet)	-	U+2043	⁃
Короткое тире (En Dash)	—	U+2013	–
Длинное тире (Em Dash)	—	U+2014	—
Горизонтальная черта (Horizontal Bar)	—	U+2015	―

рассмотренных ранее программ/веб-сервисов автоматического форматирования веб-текстов перед публикацией («типографов») с включенным символом правильного дефиса.

Знак минуса используется в математических выражениях и имеет такую же ширину, как любая цифра. В отличие от тире и дефиса минус выравняется по высоте строчной буквы (поскольку цифры всегда «выше» прописных букв).

Цифровая черточка выглядит практически как минус (но минусом при этом не является) и используется, например, для обозначения цифровых промежутков: очевидна разница между правильным написанием промежутка «10–15» и неправильным (через дефис) «10-15».

Черточный буллит употребляется в качестве маркеров списков в основном для текстов на языках, использующих латиницу. В русскоязычных текстах предпочтение отдается традиционным кружкам с квадратиками, а также длинному или короткому тире.

При использовании тире возникает следующий вопрос: когда использовать длинное тире, а когда — короткое? В русской типографике используется исключительно **длинное тире** (черточка шириной с букву «М»), а в западной типографике — **короткое тире** (черточка шириной с букву «n»). Длинное тире также употребляется в русском языке в диалогах. На Западе же для этой цели часто используется **символ горизонтальной черты**.

Текст для веб-страницы, как правило, изначально набирается и правится в текстовых редакторах. Поэтому длинные и короткие тире можно расставлять и в процессе набора текста. В текстовых редакторах операционной системы Windows для вставки короткого тире используется сочетание клавиш `<Alt>+0150`, а для вставки длинного тире — `<Alt>+0151`. В популярном текстовом редакторе MS Word для тире зарезервированы «горячие клавиши»: `<Ctrl>+<->` — для вставки короткого тире; `<Ctrl>+<Alt>+<->` — для вставки длинного тире (где `<->` — клавиша «минус» цифрового блока клавиатуры).

На рис. 3.58 показано отображение некоторых знаков-черточек в браузере Opera.

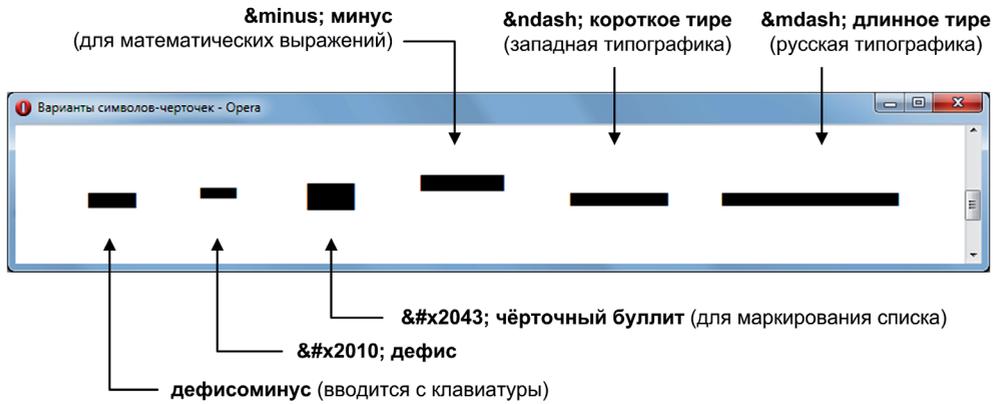


Рис. 3.58. Варианты символов-черточек. Хорошо просматриваются различия похожих знаков: дефиса и дефисоминуса, минуса и короткого тире

Использование **символа переноса** (`­`) подробно рассмотрено в предыдущем разделе. Стоит отметить, что символ переноса расставляется в словах между слогами и отображается лишь в том случае, если слово не помещается в строке. При этом часть слова с символом переноса остается на текущей строке, а другая часть — переносится на начало следующей строки.

3.3.5. Кавычки и штрихи

Кавычки — парный знак препинания, который употребляется для выделения прямой речи, цитат, отсылок, названий литературных произведений, газет, журналов, предприятий. Кавычками также «отбиваются»

отдельные слова, если они включаются в текст не в своем обычном значении, используются в ироническом смысле, предлагаются впервые или, наоборот, как устаревшие.

Использование «правильных» кавычек (в смысле их внешнего вида) — не менее запутанная тема, чем использование дефиса и тире. Это обусловлено наличием знака (") в таблице Юникода (Quotation Mark, U+0022) — так называемого **суррогата**, который не является ни кавычкой, ни знаком дюйма, ни знаком угловых секунд. Этот знак изначально использовался в программных кодах, однако любое его употребление в веб-текстах считается ошибочным.

По своему рисунку кавычки разделены на виды, которые представлены в табл. 3.7.

Таблица 3.7. Виды кавычек

Название	Вид	Юникод	Мнемоника	Комбинация клавиш
Французские кавычки (елочки)	«...»	U+00AB и U+00BB	<i>&laquo;</i> ; и <i>&raquo;</i> ;	<Alt>+0171 и <Alt>+0187
Немецкие кавычки (лапки)	„...“	U+201E и U+201C	<i>&bdquo;</i> ; и <i>&ldquo;</i> ;	<Alt>+0132 и <Alt>+0147
Английские двойные кавычки	“ ”	U+201C и U+201D	<i>&ldquo;</i> ; и <i>&rdquo;</i> ;	<Alt>+0147 и <Alt>+0148
Английские одинарные кавычки	‘ ’	U+2018 и U+2019	<i>&lsquo;</i> ; и <i>&rsquo;</i> ;	<Alt>+0145 и <Alt>+0146

Французские кавычки — традиционные кавычки-«елочки». Именно так должны выглядеть основные кавычки в русскоязычном текстовом наборе. **Немецкие „лапки“** в русскоязычном текстовом наборе используют как кавычки второго рисунка (кавычки внутри кавычек), а также в рукописных текстах. Если по техническим причинам невозможен набор кавычек другого рисунка, кавычки одного рисунка рядом не повторяются:

Правильно:	Неправильно:
«Сейчас я читаю книгу „Метрвые души“»	«Сейчас я читаю книгу «Метрвые души»»
«Сейчас я читаю книгу «Метрвые души»»	

Английские одинарные кавычки принадлежат к узкоспециализированным знакам и не используются в обычных текстах. Сфера их использования — филологические словари и справочники, а также иногда при переводе значения иноязычного слова (например, англ. *writer* — ‘писатель’).

Говоря об использовании кавычек в русскоязычных текстах, стоит упомянуть, что в текстах других языков существуют собственные правила применения этих символов. Например:

- В британском английском языке пользуются ‘английскими одинарными кавычками’ для кавычек первого уровня и “английскими двойными” для ‘кавычек “внутри” кавычек’. В американском английском языке — наоборот.
- В некоторых европейских языках (например, в Сербии, Черногории, Хорватии, Дании, Швеции и Австрии) открывающая кавычка выглядит как французская закрывающая, и наоборот. Пример: »**webbsida**«.
- В некоторых странах (например, в Финляндии) используются непарные кавычки. Пример: ”**painoasu**” или »**tyyografian**«.
- В украинском языке в качестве основных кавычек используются французские «елочки», но допускается альтернативный вариант — немецкие «лапки». Пример: „**Ще не вмерла Україна**“.

Следующий «проблемный» символ, требующий к себе пристальное внимание, — **апостроф**. Это небуквенный орфографический знак в виде надстрочной запятой (’), который употребляется в буквенном письме разных языков. Проблемность апострофа выражена наличием двух суррогатных символов («перекочевавших» в Юникод из таблицы ASCII), которые недопустимо использовать в качестве апострофа, одинарной кавычки и угловых минут: (’) — вертикальный «машинописный апостроф» (Apostrophe, U+0027); (˘) — «машинописный обратный апостроф» (Grave Accent, U+0060). Эти символы называют машинописными (которые уместны в машинописи, а не при компьютерном наборе), но самостоятельное значение они все еще имеют в некоторых языках программирования.

Для апострофа в качестве знака препинания рекомендуют использовать символ английской закрывающей кавычки — знак «Right Single Quotation Mark», а в качестве небуквенного орфографического символа — знак «Modifier Letter Apostrophe» (буква — модификатор апострофа):

Название	Вид	Юникод	Мнемоники	
Апостроф (знак препинания)	'	U+2019	<code>&rsquo;</code>	<code>&#8217;</code>
Апостроф (буква-модификатор)	'	U+02BC	–	<code>&#700;</code>

К типографским знакам также относятся **штрихи** (нем. *strich* — линия, черта, зарубка), которые представляют собой короткую, слегка наклоненную вправо, линию (одну — для одиночного, две — для двойного и три — для тройного штрихов). Штрихи обычно ставятся за правым верхним краем числового или буквенного обозначения:

Название	Вид	Юникод	Мнемоники	
Одинарный штрих	'	U+2032	<code>&prime;</code>	<code>&#8242;</code>
Двойной штрих	"	U+2033	<code>&Prime;</code>	<code>&#8243;</code>
Тройной штрих	'''	U+2034	–	<code>&#8244;</code>

Штрихи встречаются в научных и технических текстах. Ими обозначают:

- В геометрии и географии — угловые минуты (одинарный штрих) и секунды (двойной штрих): $20^{\circ} 15' 30''$ — 20 градусов 15 минут 30 секунд.
- В математическом анализе — первую, вторую и третью производные соответствующим числом штрихов: a' , b'' , c''' .
- В английской типографике — знаки фута (одинарный штрих) и дюйма (двойной штрих): $6' 10''$ — 6 футов 10 дюймов.

3.3.6. Обязательные мнемоники HTML

Как было сказано в начале этого раздела, мнемоники используются для вставки в документ символов, которые невозможно ввести с клавиатуры. Но существует несколько символов, которые присутствуют в английской раскладке клавиатуры, так как относятся к обычной ASCII, — но, в то же время, их нельзя непосредственно вставлять в текст веб-страницы, а только использовать соответствующие мнемоники.

Речь идет о **знаках «больше»** и **«меньше»**, а также об **амперсанде**. Для отображения этих знаков в компьютерных текстах используются следующие символы:

Название	Символ	Мнемоники	Назначение в HTML
Знак «меньше»	<	<code>&lt;</code> <code>&#60;</code>	Начало тега
Знак «больше»	>	<code>&gt;</code> <code>&#62;</code>	Конец тега
Амперсанд	&	<code>&amp;</code> <code>&#38;</code>	Начало мнемоники

Если знак «<» встречается в веб-странице, браузер попытается интерпретировать его как элемент разметки. Поэтому, чтобы все-таки вывести на страницу сам символ «<», нужно использовать именованную подстановку `<`. Из этих же соображений, чтобы вывести символ «>», применяется подстановка `>`.

К сожалению, не все разработчики знают о том, что сам символ «&» требует подстановки. Причина здесь такая же, как в случае с символами «<» и «>»: амперсанд в HTML обозначает начало подстановки, следовательно, если включать его в документ непосредственно, браузер может неправильно отобразить текст, особенно если после знака «&» стоит слово, завершающееся точкой с запятой.

Мнемоники этих символов также позволяют размещать тексты, описывающие назначения тегов и мнемоник для HTML (например, на тему «Справочники HTML и CSS»). Пренебрежение обязательными мнемониками приведет к искажению информации на веб-странице. Например, следующий код HTML

```

1  ...
2  <p>
3     Теги <b> и <strong>, а также <i> и <em> выполняют
4     одни и те же действия, однако теги <b> и <i> являются
5     тегами физического форматирования, а теги <strong>
6     и <em> – тегами логического форматирования.
7  </p>
8  <p>
9     Мнемоники &nbsp;, &ensp;, &ensp; и &thinsp;
10    используются для вставки пробельных символов.
11  </p>
12  ...

```

в браузере Chrome будет выглядеть так, как это показано на рис. 3.59.

Ниже представлен аналогичный код HTML, с использованием подстановок (на рис. 3.60 показано его выполнение в браузере Chrome):

```

1  ...
2  <p>
3      Теги <b> и <strong>, а также <i>
4      и <em> выполняют одни и те же действия, однако
5      теги <b> и <i> являются тегами
6      физического форматирования, а теги <strong>
7      и <em> — тегами логического форматирования.
8  </p>
9  <p>
10     Мнемоники &nbsp;, &ensp;, &ensp; и
11     &thinsp; используются для вставки пробельных
12     символов.
13 </p>
14 ...

```

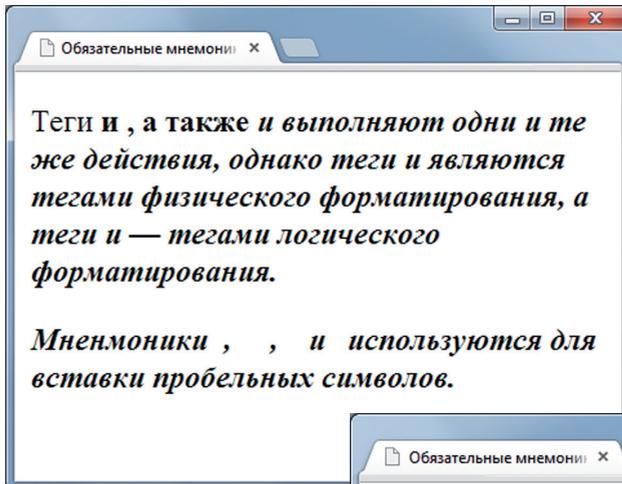


Рис. 3.59. Игнорирование обязательных мнемоник в коде HTML

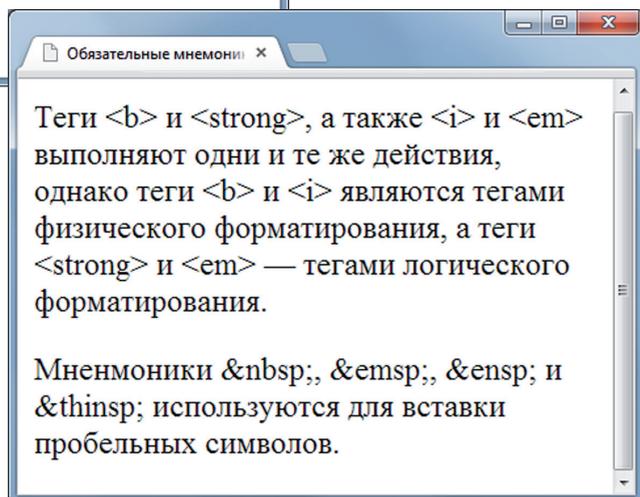


Рис. 3.60. Правильное использование обязательных мнемоник

Контрольные вопросы

1. Что такое мнемоники в текстовых блоках HTML?
2. Какими способами можно задавать специальные символы в веб-текстах?
3. На какие категории можно условно разделить типографские знаки?
4. Какие диакритические знаки существуют в современной письменности русского языка?
5. Какие бывают пробелы по отношению к незаполненным участкам текста?
6. Что такое неразрывный пробел? Какая мнемоника позволяет вставить неразрывный пробел в текст HTML?
7. Какие символы-черточки содержит таблица Юникода?
8. В каких случаях используется длинное и короткое тире?
9. Как длинное и короткое тире можно расставлять в процессе набора текста в текстовых редакторах операционной системы Windows?
10. Что такое кавычки? На какие виды по своему рисунку разделены кавычки?
11. В каких случаях используются английские, немецкие и французские кавычки?
12. Какая мнемоника позволяет вставить апостроф в текст HTML?
13. В каких случаях необходимо использовать одиночный, двойной и тройной штрихи? Какие мнемоники позволяют вставить штрихи в текст HTML?
14. Что такое обязательные мнемоники HTML? Для каких символов необходимо обязательно использовать мнемоники в текстах HTML?

3.4. Основы грамматики

Ни для кого не секрет, что грамотность текста в современном веб-пространстве находится далеко не на лучшем (и даже не на приемлемом) уровне.

Всемирная паутина широко открыла двери в плане свободы слова. Сейчас каждый может разместить в Сети любую информацию, которая незамедлительно станет доступна читателям всего мира. Даже закрыв глаза на достоверность информации (стоит упомянуть, что Интернет не защищен от недостоверной информации, как, впрочем, и от отсутствия культуры письменной речи), приходится наблюдать гнетущую и, естественно, нарастающую безграмотность текстового содержимого сайтов.

Помимо «вседозволенности» Интернета, нельзя не обратить внимание на современное состояние среднего общего образования, размывание моральных ограничений, потерю культурного наследия. Все это в сумме дает уже обретшее явную форму всепроникающее явление — нескрываемая безграмотность и неуважение к родному языку.

Мнение, что безграмотность в тексте угнетает и «убивает» грамотность читателя, высказано многими равнодушными к этому явлению людьми. В частности, В. В. Артюхин в книге «Реальность 2.0b. Современная история информационного общества» отмечает следующее: *«Совершение ошибок не является преступлением, всем людям присуще ошибаться <...>. Дело в нынешнем отношении к грамотности многих онлайн-авторов. Преступление — это игнорирование уже выявленных ошибок, как и самой необходимости прилагать усилия для грамотного, с точки зрения правил языка, изложения своих мыслей. <...> Что касается постов в блогах, <...> среди свежих публикаций невозможно найти ни одной, написанной без явных, хорошо заметных ошибок. Человек, даже будучи убежденно грамотным, регулярно читая тексты с ошибками, которые якобы «не влияют на понимание» (согласно множеству «крайне полезных» для языка исследований), и уже ожидая их увидеть, постепенно и неосознанно перестраивает свою систему ценностей, и сам перестает уделять внимание грамотности»* [46, с. 234–235].

Разработчики некоторых крупных сайтов привлекают профессиональных редакторов и корректоров для грамотной по отношению

к языку и типографике подачи текстового материала. Для большинства сайтов такая практика невозможна ни сейчас, ни в обозримом будущем по двум причинам. Во-первых, многие и не подозревают, что их тексты нуждаются в профессиональной обработке (тем более, что многие пользователи Интернета ведут блоги, общаются в форумах и социальных сетях, т. е. «стряпают» тексты онлайн), во-вторых, исходя из возможностей и разного рода убеждений, многие не готовы оплачивать труд профессионалов, если текст в виде «как есть» можно (безболезненно, безнаказанно) разместить на сайте, руководствуясь принципом «не нравится — не читай!».

Тем не менее, поднимать уровень грамотности текстов и их оформления для последующего размещения в веб-страницах можно (и нужно) самостоятельно — это, собственно, и является основной целью данного раздела, в котором и собраны некоторые правила русской и (совсем немного) английской грамматики, которые особенно часто нарушаются в веб-текстах.

Соблюдение правил грамматики является важным критерием грамотности. Но эти правила для текстов веб-страниц такие же, как и для любого другого текста, и с ними можно ознакомиться в различных справочных пособиях (например, в [47–49]).

3.4.1. Пробелы

Главное, что нужно помнить при вводе пробельного символа: пробел — признак нового слова. Тем не менее, в текстах веб-страниц встречаются ошибки, связанные с пробелами. Ниже приведены основные правила, соблюдение которых избавит пользователей от большей части подобных ошибок:

- Пробел ставится после сокращений. Так, распространенные сокращения *и т. д.*, *и т. п.*, *т. е.*, *т. н.*, *до н. э.*, *у. е.*, *в т. ч.* и им подобные нужно записывать с соответствующими пробелами.
- Между именными инициалами ставится соответствующий пробел: *И. И. Иванов*, *П. П. Петров*, *С. С. Сидоров*.
- Число и относящиеся к нему единицы измерения записываются с соответствующим пробелом: *50 км*, *2013 г.*, *XXI в.*
- Скобки и кавычки не отделяются пробелами от заключенного в них текста.

Лишние пробелы из текста удаляются. В тексте не может быть ни одного случая, когда необходимо ставить несколько подряд идущих пробелов. Между словами ставится один пробел¹!

Правильно:	Неправильно:
и_т_д.	и_т_д.
у_е.	у.е.
И_А_Крылов	И.А.Крылов, И.А._Крылов
1974_г.	1974г.
20_т	20т
рис._1.	рис.1.
Рис._1._Квадрат (текст в скобках)	Рис._1.Квадрат, Рис._1._Квадрат (_текст в скобках_)
[45]	[_45_]
«Война и мир»	«_Война и мир_»

В первых трех правилах используется фраза «соответствующий пробел». Дело в том, что пробел является потенциальным разделителем строк, а записи некоторых словесных конструкций не допускают переноса слов. Поэтому под фразой «соответствующий пробел» кроется необходимость уточнения, в каких ситуациях использовать обычный пробел, а в каких — неразрывный. Естественно, неразрывный пробел ставится в месте, где переносы слов запрещены.

Последнее правило (про лишние пробелы) носит, скорее всего, рекомендательный характер, поскольку браузер самостоятельно удаляет повторяющиеся пробельные символы (речь идет о простом пробеле, введенном с клавиатуры). Подряд идущие пробелы, включенные в веб-текст с помощью подстановок (мнемоник), браузером не удаляются.

Но лучше привыкать к набору текстов без лишних пробелов. Для удобного контроля за количеством пробелов в настройках текстовых

¹ Поскольку пробел это все-таки «пустой» символ, в некоторых примерах данного раздела использован специальный символ «_» для его обозначения. Например, если нужно подчеркнуть наличие пробела между словами, используется конструкция «слово1_слово2». Также появляется возможность акцентировать внимание на количестве подряд идущих пробелов: «слово1__слово2».

редакторов лучше включать отображение скрытых символов форматирования: при этом пробелы станут «видимыми» и будут обозначены символом (·).

3.4.2. Знаки препинания

В системе современной русской пунктуации наиболее часто используются следующие **знаки препинания**:

Название	Символ
точка	.
запятая	,
точка с запятой	;
многоточие	...
двоеточие	:
вопросительный знак	?
восклицательный знак	!
короткое и длинное тире	— —
дефис	-
скобки	([{ }])
кавычки	« “ ” »

Реже используются специальные и комбинированные знаки: !! ?! !.. и др. Необходимо помнить о важном правиле для некоторых знаков препинания, которое часто нарушается в веб-текстах:

Точка, запятая, точка с запятой, двоеточие, многоточие, восклицательный знак, вопросительный знак не отделяются пробелом от предшествующего символа.

Поскольку пробел является признаком нового слова, то слово, которое начинается после знака препинания, отделяется от него пробелом:

Правильно:	Неправильно:
постольку, _поскольку	постольку,поскольку
Ночь _Улица _Фонарь _Аптека...	Ночь_Улица_Фонарь.Аптека_...

Однако после знаков препинания пробел ставится не всегда. Например, когда после знака препинания новое слово не начинается.

Таким образом, пробел не нужен:

- Если после знака препинания следуют закрывающие скобка или кавычка. Например, «*Что же будет?*» — *подумал он, кто это? (или что это?)*.
- Если точка разделяет уровни нумерации, например, в заголовках или элементах перечня: 3.1.15. *Заключение* (здесь «Заключение» является новым словом, поэтому оно отделено пробелом от нумерации);
- Если запятая или точка разделяют целую и дробную части числа: 9,8, 3.1415.
- Если точка используется в адресах Интернета, именах файлов и других технических обозначениях: *www.user.ru, isv@mail.ru, file.txt*.
- Если указывается дата или время: *11.02.1974, 18.30, 15:44:59*.

Знаки препинания при скобках и кавычках ставятся согласно следующим правилам [49]:

- Точка, запятая, точка с запятой, двоеточие и тире не ставятся перед закрывающимися кавычками; все эти знаки могут стоять только после кавычек. Например: *Одни голосовали «за», другие — «против», но первых было явное большинство*.
- Вопросительный и восклицательный знаки и многоточие ставятся перед закрывающимися кавычками, если относятся к словам, заключенным в кавычки, например: *Роман «Что делать?» написан Н. Г. Чернышевским; Громкое «браво!» разнеслось по залу; «Я хотел бы поделиться с вами...» — так начал он свой рассказ*.
- Если же знаки вопросительный, восклицательный и многоточие относятся ко всему предложению вместе со словами, заключенными в кавычки, то названные знаки ставятся после закрывающих кавычек, например: *Где вы найдете больше, чем у нас, месторождений «черного золота»? Надоело мне ваше «За ответом приходите завтра»! Прежде чем начнете читать роман «Отцы и дети»...*
- Если перед закрывающими кавычками стоит вопросительный или восклицательный знак, то запятая после кавычек ставится только по требованию контекста, например: *В какой-то период жизни каждый из нас ставит перед собой вопрос «кем быть?», но отвечаем на него мы по-разному*.

- Если перед закрывающими кавычками стоит вопросительный или восклицательный знак, то тот же самый знак не повторяется после кавычек; неодинаковые же знаки, если они требуются по условиям контекста, ставятся перед закрывающими кавычками и после них. Например: *Читали ли вы статью в «Известиях» «Куда мы идем?»*; *Когда был выдвинут лозунг «Вся власть Советам!»?*
- Перед открывающей или закрывающей скобкой не ставятся запятая, точка с запятой, двоеточие и тире; все эти знаки ставятся только после закрывающей скобки, например: *У него было три дочери (он их даже специально так назвал): Вера, Надежда и Любовь.*
- Точка, вопросительный и восклицательный знаки ставятся перед закрывающей скобкой, если относятся к словам, заключенным в скобки, например: *Прощай, сестрица! (Целуется с Варварой.) Прощай, Глаша! (Целуется с Глашей.) Прощайте, маменька! (Кланяется.); Осторожно ступая по свежевыструганным, еще не выкрашенным (не хватило краски!) половицам, Доронин миновал сени и вошел в светлую большую комнату.*
- После закрывающей скобки ставится знак препинания, требуемый условиями контекста, независимо от того, какой знак стоит перед закрывающей скобкой, например: *Быть может (лестная надежда!), укажет будущий невежда на мой прославленный портрет; Используются клиентские технологии (HTML, CSS, JavaScript и т. д.).*

3.4.3. Дефис и тире

Дефис соединяет части сложных слов, например:

- Наречия с приставкой **по-**, оканчивающиеся на **-ому**, **-ему**, **-ки**, **-ни**, **-ьи**, например: **по-новому**; **по-вашему**; **по-дружески**; **по-прежнему**; **по-латыни**.
- Наречия с приставкой **в-** (**во-**), образованные от порядковых числительных, например: **во-первых**; **в-седьмых**; а также **в-последних**.
- Неопределенные наречия с частицами **-то**, **-либо**, **-нибудь**, **кое-**, **-таки**, например: **когда-то**; **откуда-либо**; **как-нибудь**; **кое-где**; **все-таки**.

- Наречия, образованные повторением того же самого слова или связанных по ассоциации слов, например: *едва-едва; чуть-чуть; как-никак; крест-накрест; туго-натуго; мало-помалу; неожиданно-негаданно.*
- Частицу **-таки**, например: *верно-таки; прямо-таки; все-таки; неужели-таки; настоял-таки; ушел-таки.*
- Сложные междометия, например: *ей-богу; ей-же-ей; о-го-го; ой-ой-ой; ха-ха-ха; динь-динь-динь; кис-кис; мяу-мяу.*
- Так называемый «**висячий**» **дефис** употребляется при сочетании нескольких сложных слов с одинаковой второй частью: *радио- и телетрансляция; двух-, трех- и четырехместные палатки.*

При **расстановке дефисов** следует помнить, что:

- во всех случаях, где правила русского языка предписывают употреблять дефис, **не следует ставить вместо него тире или какой-нибудь другой знак.**
- дефис **не отделяется пробелами** от тех частей слова, которые он соединяет.

Тире употребляется для разделения разных слов. Например: *твои речи — будто острый нож; курить — здоровью вредить!; Киев — мать городов русских.*

В компьютерных текстах веб-типографика предполагает использование двух видов тире — **короткого** и **длинного**.

Короткое тире употребляется:

- При обозначении промежутков:
 - временных, например: *1799–1837 гг.; XIX–XX вв.; 12–15 мая;*
 - пространственных, например: *поезд Москва – Петербург;*
 - количественных, например: *5–6 кг; 23,5–25 км; 7–10 °С.*
- В названиях физических законов, открытых несколькими учеными, например: *закон Джоуля – Ленца; закон Бугера – Ламберта – Бера.*

В приведенных правилах употребления короткого тире можно заметить, что **короткое тире не отделяется пробелами, если с обеих сторон от него стоят цифры.** Во всех остальных случаях необходимо отделять короткое тире пробелами с двух сторон (табл. 3.8).

Однако **в составных номерах между группами цифр нужно ставить дефис, а не тире**, поскольку это один целый номер, а не несколько разных. К таким номерам относятся, например:

- телефонные номера: *050-123-45-67*;
- номера ISBN и ISSN: *ISBN 978-966-8248-20-7; ISSN 1816-2681*;
- номера ГОСТов: *ГОСТ 7.56-2002*;
- обозначение даты: *2014-08-14*.

Длинное тире употребляется во всех остальных случаях, где правила русского языка предписывают использовать тире. **Длинное тире всегда отделяется пробелами с обеих сторон** (табл. 3.8).

Таблица 3.8. Примеры вставки дефиса и тире

Правильно:	Неправильно:
все-таки	все_таки все–таки все—таки все_–_таки
1939–2001	1939-2001 1939_–_2001 1939—2001
999-88-77	999–88–77
1938-01-25	1938–01–25
закон Генри_–_Дальтона	закон Генри-Дальтона закон Генри–Дальтона закон Генри—Дальтона
Куренье_—_вред	Куренье—вред Куренье–вред Куренье-вред Куренье_–_вред

3.4.4. Числа и математические знаки

Целые числа в цифровой форме, в которых больше четырех разрядов, **разбиваются пробелами по разрядам** (группы по три цифры) справа налево. Не разбиваются числа, стоящие после знака номера (№), в обозначениях марок машин и в номерах документов:

Правильно:	Неправильно:
1_234_567_890	1234567890
6245	6_245
№_1042554	№_1_042_554

Для группировки чисел по разрядам можно использовать символ обычного пробела (с точки зрения ширины символа), однако нужно позаботиться о том, чтобы число не разрывалось при переносе строк. В этом случае можно использовать символ неразрывного пробела или тонкой шпации с запретом разрыва строки. Так, число 1 234 567 890 можно вставить в код HTML двумя способами:

```
1 ...
2 <style>
3   span.noLineBreak { white-space: nowrap; }
4 </style>
5 ...
6
7   <!-- с использованием неразрывного пробела -->
8 <p>1&nbsp;234&nbsp;567&nbsp;890</p>
9
10  <!-- с использованием тонкой шпации -->
11 <p>
12   <span class="noLineBreak">
13     1&thinsp;234&thinsp;567&thinsp;890
14   </span>
15 </p>
16 ...
```

В **десятичных дробях** между целой и дробной частью ставится запятая (после которой не должно быть пробела). Многие разработчики сайтов, знакомые с программированием, привычно ставят не запятую, а точку — и таким образом совершают ошибку (табл. 3.9).

Натуральные дроби в тексте записываются через дробную черту (символ «/» таблицы Юникода, U+2044), причем числитель указывается в верхнем индексе, а знаменатель — в нижнем. Если дробное число имеет целую часть, она пишется слитно с дробью (табл. 3.9). Для некоторых натуральных дробей в таблице Юникода можно найти соответствующие символы, например, $\frac{1}{3}$, $\frac{5}{8}$, $\frac{2}{3}$.

Таблица 3.9. Примеры записи десятичных и натуральных дробей

Правильно:	Неправильно:
2,71828	2.71828
$\frac{1}{3}$ (U+2153)	1/3
$\frac{1}{5}$	1/5
$3\frac{5}{7}$	3_ ⁵ / ₇

В **обозначениях даты и времени** числа записываются следующим образом:

- **2015-08-20**, что обозначает «20 августа 2015 года» (формат YYYY-MM-DD согласно международному стандарту ISO 8601:2004 [50], где Y, M и D — цифры года, месяца и дня соответственно).
- **20:19:45**, что соответствует «20 часов 19 минут 45 секунд» (24-часовой формат HH:MM:SS согласно международному стандарту ISO 8601:2004, где H, M и S — цифры часов, минут и секунд соответственно).

Такие форматы записи даты и времени характерны для технических устройств и специализированных текстов.

В текстах, предназначенных для широкого круга читателей, в русской типографике принято употреблять традиционные, не противоречащие правилам русского языка, записи, например: *20.02.2013*; *20.19*.

В английском языке для представления времени иногда используется «**британская**» **12часовая система**, в которой используются сокращения «*a. m.*» (от лат. *ante meridiem* — до полудня) и «*p. m.*» (от лат. *post meridiem* — после полудня). Например: *7:30 a. m.* (7.30 — половина восьмого утра); *9:00 p. m.* (21.00).

Словесная форма даты в английском языке также отличается от соответствующей записи на русском языке:

- русский язык: *20 февраля 2013, среда*;
- английский язык: *Wednesday, February 20, 2013*.

Знаки математических операций (+, −, ×, ·, ÷ и др.) и **соотношений** (<, ≤, >, ≥, ≡, ≈, ⊂, ⊄, ⊆, ⊃, ⊇ и др.) в выражениях отделяются от окружающих чисел или переменных пробелами:

Правильно:	Неправильно:
3_+_2	$3+2$
$P_=_3,22$	$P=3,22$
$x_>_20$	$x\geq 20$
$A_ \subseteq _B$	$A\subseteq B$
$125_:_4_=_31,25$	$125:4=31,25$ $125:4_=_31,25$

Большинство знаков математических операций и соотношений вставляется в код HTML с помощью соответствующих мнемоник.

При использовании математических знаков в текстах веб-страниц часто нарушаются следующие правила:

- В качестве **знака умножения** используется точка (\cdot) или крестик (\times). Грубейшей ошибкой считается использование в качестве знака умножения буквы «х»:

Правильно:	Неправильно:
$3 \cdot 2 = 6$	$3 \times 2 = 6$
$3 \times 2 = 6$	$3 X 2 = 6$

- При **указании габаритных размеров** используется крестик (\times), который не отделяется от чисел пробелами. Например: *ручная кладь 30×40×50 см; разрешение экрана: 1680×1050.*
- Если **математический символ употребляется перед одиночным числом** (например, чтобы показать его знак или приращение), пробел между символом и числом не ставится:

Правильно:	Неправильно:
+20	+_20
-5	_5
+36,6 °C	+_36,6 °C

- Числа в верхнем и нижнем индексе** не отделяются пробелом от предшествующего символа:

Правильно:	Неправильно:
x^3	x_3
$Na(CO)_2$	$Na(CO)_2$
a_{10}	a_10

3.4.5. Специальные знаки

- Знаки номера (№) и параграфа (§)** без стоящих за ними чисел не употребляются и отделяются от них пробелом:

Правильно:	Неправильно:
дом №_191	дом №191
см. §_3	см. §3
Второй параграф прочитан	Второй § прочитан
Срочно в номер!	Срочно в №!

Стоит отметить, что знак номера (№) в английском языке не употребляется. Вместо него используется решетка (#) или сокращение «No.». И решетка, и сокращение отделяются от числа пробелом. Например: # 7; No. 15. При этом в русском языке не следует использовать вместо знака «№» решетку «#» или сокращение «No.».

- Знак охраны авторского права** состоит из трех частей и записывается именно в такой последовательности: символ «©», имя или наименование обладателя исключительных авторских прав, год первого опубликования. Между именем и годом ставится запятая, а символ «©» отделяется пробелом от последующего слова:

Правильно	Неправильно
©	(с), (С)
© «Урал-Пресс», 1996	© «Урал-Пресс» 1996 © 1996
© Иван Иванов, 2012	© Иван Иванов

- Знаки единиц измерений**, записанные в верхнем индексе (', ", ", °), не отделяются пробелом от предшествующего числа:

Правильно:	Неправильно:
10°	10_°
25'	25_'
10°_25'_45"	10_° 25_' 45_"

В английском языке применяются специальные знаки денежных единиц — например, доллар (\$), евро (€), фунт стерлингов (£). Эти знаки записываются перед цифрой и не отделяются от нее пробелом. Например: \$999.99; €500. **Знаки денежных единиц в тексте на русском языке не употребляются.**

3.4.6. Сокращения

В печатных изданиях к сокращению прибегают в основном для экономии бумаги. Повсеместное применение сокращений — справочная литература, словари и энциклопедии. В электронных текстах вопросы экономии и «сжатия» текста не актуальны. Здесь встают вопросы удобочитаемости, а «злоупотребление» сокращениями неизменно сказывается на легкости и скорости чтения. Поэтому в веб-текстах рекомендуется избегать принятые в книжных текстах сокращения. Даже если тексты сайта носят «справочный» характер, нужно расшифровывать все сокращения.

Тем не менее, бывают ситуации, когда сокращения в тексте на веб-странице целесообразны. При этом у многих веб-авторов возникают сложности, связанные с правильным сокращением слов.

При сокращении слов следует выполнять следующие общие рекомендации [51]:

- Сокращения должны быть понятны читателю. Большое число необщепринятых сокращений затрудняет чтение текста. При усечении слова оставшаяся часть должна позволять легко и безошибочно восстанавливать полное слово, например: *филос.*, *филол.* (но не *фил.*).
- Нежелательны сокращения, совпадающие по написанию с другими. Такие сокращения допустимы только в том случае, если контекст подсказывает, какое именно слово или словосочетание сокращено.
- Сокращения должны быть единообразными. Принцип единообразия выдерживается, когда сокращаются (или не сокращаются) все однотипные слова. Форма сокращения при этом должна быть одинаковой.

Точку после сокращения обычно ставят, если слово с отсеченной конечной частью при чтении вслух произносится в полной, а не сокра-

щенной форме, например: *г.* — год, но *КПД* — коэффициент полезного действия (при чтении так и произносится: «кэпэдэ»). Точку не ставят при обозначении стандартных физических величин: *м*, *мм*, *кг* и т. п.

Также стоит рассмотреть некоторые часто нарушаемые правила сокращений с подробными объяснениями и рекомендациями:

- **Слова «год» и «век», следующие за числом**, принято сокращать до одной буквы с точкой после нее: *1884 г.*; *IV в. до н. э.* Во множественном числе «года» и «века» после чисел сокращаются соответственно до «гг.» и «вв.»: *XIX–XX вв.*; *30е гг.* Во фрагментах текста, где важна краткость (заголовки, пункты меню), слово «год» можно опустить, но только если понятно из контекста, о чем идет речь: *11 февраля 1974; в 2012.*
- **Падежное окончание порядковых числительных**, обозначенных арабскими цифрами, должно быть однобуквенным, если предпоследняя буква в слове — гласная, и двухбуквенным — если согласная. Например, в слове «*третьего*» предпоследняя буква «г» — согласная, следовательно, сокращение будет «*3-го*». Еще несколько примеров: *1-й* (первый); *2-му* (второму); *4-го* (четвертого); *20-ми* (двадцатыми). Не рекомендуется использовать падежные окончания там, где грамматическая форма числительного может быть непонятна из контекста, например: *7-м* (седьмым или седьмом?); *20-й* (двадцатый или двадцатой?).
- **Слова «миллион» и «миллиард»** сокращаются до «млн» и «млрд» соответственно. Точка после этих сокращений не ставится: *2 млн туристов*; *35 млрд клеток*. В сокращении слова «**тысяча**» — «*тыс.*» точка ставится: *10 тыс. экземпляров*.
- **Сокращения единиц измерения** употребляются только вместе с числом и записываются по правилам метрической системы, без точки: *г* (грамм), *м* (метр), *с* (секунда) и т. д. Если число отсутствует, то единицы измерения записываются полностью: *несколько сантиметров*; *измеряется в граммах*.
- **Единицы измерения, образованные от фамилий**, в сокращенной форме обычно пишутся с прописной буквы: *Н* (ньютон), *Па* (паскаль), *Гц* (герц), но в полной форме — только со строчной: *сотни герц*; *ток в два ампера*.
- **Десятичные приставки** записываются по правилам метрической системы: в частности, приставки «*микро-*», «*милли-*»

и «кило-» сокращаются строчными буквами «мк», «м» и «к» (мкФ; мВт; кг), а приставки «мега-», «гига-», «тера-» — прописными буквами «М», «Г» и «Т» (МПа; ГГц; Тбайт).

3.4.7. Переносы слов и строк

В русском языке целый свод правил посвящен тому, как нужно переносить слова и разбивать строки. При этом для веб-текстов более актуальны правила переноса строк, чем правила переноса слов, поскольку браузеры не поддерживают автоматическую расстановку мягких переносов. Те, кто намерен осуществить расстановку мягких переносов для слов текста вручную (например, с использованием мнемоники `­`), могут воспользоваться исчерпывающими правилами переноса слов из соответствующей справочной литературы, например, из [47–49; 51;52].

Далее приведен «**минимальный комплект**» **правил переноса слов** в русском языке (но необходимый и достаточный для большинства случаев):

- При переносе слов нельзя ни оставлять в конце строки, ни переносить на другую сторону часть слова, не составляющую слога. Например, нельзя переносить: *просмо-тр*; *ст-рах*.
- Нельзя отделять согласную букву от следующей за ней гласной (табл. 3.10, № 1).
- Если после приставки стоит буква «ы», то переносить часть слова, начинающуюся с «ы», не разрешается (табл. 3.10, № 2).
- Нельзя отрывать буквы «ъ» и «ь» от предшествующей согласной (табл. 3.10, № 3).
- Нельзя отрывать букву «й» от предшествующей гласной (табл. 3.10, № 4).
- Нельзя оставлять в конце строки или переносить на другую строку одну букву (табл. 3.10, № 5).
- Нельзя разбивать переносом односложную часть сложносокращенного слова (табл. 3.10, № 6).
- Нельзя делать переносов, искажающих смысл слова, неблагозвучных или создающих двусмысленные толкования (табл. 3.10, № 7).
- Нельзя разбивать переносом буквенные аббревиатуры, как пишущиеся одними прописными, так и пишущиеся частью строч-

ными, частью прописными или прописными с цифрами, например: *МИД, Кзот, ТУ-104*.

Из всей совокупности правил переноса следует, что многие слова можно переносить различными способами. Однако необходимо предпочитать такие переносы, при которых не разбиваются значащие части слова.

Таблица 3.10. Примеры переноса слов

№	Правильно	Неправильно
1	лю-бовь дя-денька, дядень-ка ре-бята, ребя-та па-стух, пас-тух	люб-овь дяд-енька реб-ята паст-ух
2	ра-зыскать, разыс-кать ро-зыгрыш, розыг-рыш	раз-ыскать роз-ыгрыш
3	подъ-езд боль-шой бу-льон	под-ъезд бол-ьшой бул-ьон, буль-он
4	вой-на стой-кий фей-ерверк, фейер-верк май-ор	во-йна сто-йкий фе-йерверк ма-йор
5	ака-ция	а-кация, акаци-я
6	спец-одежда	спе-цодежда
7	брига-ды про-цедура, процеду-ра побе-да	бри-гады проце-дура по-беда

Правила переноса строк более востребованы в веб-текстах. Не смотря на то, что браузеры сами следят за тем, чтобы не разрывать строку посередине слова или числа, чтобы не отрывать знак препинания от предшествующего слова и т. д., во многих случаях они все-таки разрывают строку в неподходящих местах. Разработчик сайта должен позаботиться о том, чтобы строки разрывались только в местах, не противоречащих правилам переноса строк:

- Нельзя разрывать строку там, где не ставится пробел. В частности, не может быть перевода строки перед любым знаком препинания, после открывающей или перед закрывающей скобкой и кавычкой, перед дефисом или после него, перед коротким тире (между числами) или после него.

- Нельзя разрывать строку перед длинным тире, но после него разрыв строки допускается.
- Нельзя разрывать сокращенные выражения *и т. д.*, *и т. п.*, *т. е.* и им подобные.
- При переносе нельзя отрывать инициалы от фамилии и друг от друга.
- Нельзя также отрывать имена собственные от предшествующих сокращений, например: *тов. Сидоров; г. Донецк.*
- Нельзя разрывать число и относящееся к нему сокращенное наименование: *50 км; 2015 г.*
- Нельзя отрывать знаки номера (№) и параграфа (§) от относящихся к ним чисел.
- нельзя оставлять в конце строки предлоги, союзы и частицы, начинающие предложение. Нельзя также оставлять в конце строки однобуквенные предлоги и союзы.
- нельзя отрывать частицы «же», «бы», «ли» от связанного с ними слова.
- Нельзя разрывать многоразрядные числа, разряды которых разбиваются пробелами.

Ниже приведены примеры, показывающие места, в которых запрещено переносить строки (знак § показывает место запрета переноса, но это не всегда место пробела):

- и § т. § д.; и § т. § п.; т. § е.;
- когда § - §нибудь; скатерть § - §самобранка;
Петров § - §Водкин;
- 1799 § - §1837 § гг.; маршрут Донецк § - §Львов;
миру § — мир!;
- В. § В. § Маяковский; Дж. § Р. § Толкин;
- проф. § Бушуев; пос. § Первомайский; ул. § Куйбышева;
- п. § 2.1; № § 17; § § 3; 15 § августа; 10 § м²; 500 § Гб;
- в § парке...; на § обочине...; и § дольше века...;
- так § же; то § ли; как § бы;
- 1 § -е; 2 § -го; 1 § 234 § 567 § 890.

3.4.8. Текстовые выделения

О видах выделения и их употреблении на сайте было рассказано ранее. Здесь пойдет речь о синтаксическом оформлении некоторых видов выделений [52]:

- **Заголовок в подбор.** Это тот самый заголовок, с которого начинается эта строка. После него ставится точка, которая выделяется тем же способом, что и сам заголовок (в данном случае — полужирным шрифтом).
- Если предложение внутри текста выделено целиком, то завершающий его знак препинания также выделяется. Например: *Думай по-другому!* (Восклицательный знак также выделяется курсивом).
- Если несколько слов или словосочетаний подряд выделены как независимые объекты, то разделяющие их знаки препинания не выделяются. Например:

Выберите правильные *заголовок*, *заголовочные надписи*, *названия пунктов навигационного меню*, чтобы посетитель сразу уловил тематическое направление вашего сайта.

Здесь каждое слово в отдельности выделено курсивом, но запятые между ними имеют прямое начертание.

- Если словосочетание выделено как единый объект, то знаки препинания внутри него также выделяются, а знаки препинания, стоящие после выделенного фрагмента, — не выделяются:

Правильно:	Неправильно:
<i>Музыка</i> , как особая эстетическая ценность...	<i>Музыка</i> , как особая эстетическая ценность... (запятая выделена курсивом)
... фундаментальный уровень истории музыки.	... фундаментальный уровень истории музыки. (точка выделена полужирным шрифтом)

- Если выделенное слово, словосочетание или предложение окружено скобками, сами скобки не выделяются. Кавычки в этом случае, наоборот, выделяются так же, как заключенный в них фрагмент:

Правильно:	Неправильно:
Сетевая литература (<i>сетература</i>) в последнее время...	Сетевая литература (<i>сетература</i>) в последнее время...
литературное общество « Арзамас »	литературное общество « Арзамас »
сказка Сент-Экзюпери « <u>Маленький принц</u> »	сказка Сент-Экзюпери « <u>Маленький принц</u> »

В веб-текстах следует также учитывать, что **гиперссылка** — один из видов выделения, и на нее распространяются все правила синтаксического оформления текстовых выделений:

Правильно:	Неправильно:
E-mail: <u>ijsv74@mail.ru</u> , <u>ivanitsa-serg@rambler.ru</u> .	E-mail: <u>ijsv74@mail.ru</u> , <u>ivanitsa-serg@rambler.ru</u> . (запятая и точка выделены подчеркиванием)
В «Википедии» (<u>www.ru.wikipedia.org</u>) можно найти...	В «Википедии» (<u>www.ru.wikipedia.org</u>) можно найти... (скобки и пробельный символ выделены подчеркиванием)

Правила типографики, как и правила грамматики, в каждом языке свои. Например, в английском языке для разделения уровней нумерации может использоваться не точка, а, например, короткое тире или номер заголовка может не отделяться точкой. Такие моменты необходимо учитывать при переводе. Например:

- английский оригинал: *Table 2-5. HTML Basic Tags*
3.1 Introduction to CSS3
- перевод на русский язык: *Таблица 2.5. Основные теги HTML*
3.1. Введение в CSS3

* * * * *

Говоря о грамматике для текстов, стоит упомянуть и о языковой культуре. Языки всех народов мира постоянно развиваются. При этом правила и нормы, зафиксированные в нормативной и справочной литературе, не управляют развитием языка, а, наоборот, закрепляют уже

сложившиеся принципы и традиции. В каждом языке находится место и специальной терминологии, и жаргону, и ненормативной лексике.

Язык отражает культуру и духовное богатство человека. Поэтому посетители сайта будут судить о его разработчике по стилю и грамотности текста, по его типографскому и художественному оформлению. Следовательно, качественный и грамотно оформленный текст — залог хорошего отношения пользователей и к самому сайту, и к его разработчику.

Контрольные вопросы

1. В каких случаях необходима обязательная вставка пробела?
2. Какие знаки препинания используются в системе современной русской пунктуации?
3. В каких случаях необходимо использовать дефис?
4. Когда необходимо употреблять длинное и короткое тире?
5. Как правильно записываются целые числа в цифровой форме?
6. Как необходимо записывать десятичные дроби в веб-текстах?
7. Как в тексте правильно записывать натуральные дроби?
8. Как правильно записываются числа при обозначении даты и времени?
9. Как в текстах правильно использовать знаки математических операций?
10. Как правильно сокращать слова? Назовите основные правила и приведите примеры сокращения слов.
11. В чем заключается особенность переноса строк в веб-текстах?
12. Как синтаксически правильно оформлять текстовые выделения в веб-текстах?

Список литературы к главе 3

1. Бородаев Д. В. Web-сайт как объект графического дизайна. Монография. /Д. В. Бородаев. — Х.: «Септима ЛТД», 2006. — 288 с.
2. Иваница С. В. Веб-типографика. Искусство оформления текстов для Интернета. / С. В. Иваница. — Донецк: ДонНТУ, УНИТЕХ, 2013. — 384 с.: ил.
3. Font Problem. Все о шрифтах [Электронный ресурс]. — Режим доступа: <http://fontproblem.narod.ru/srift.htm>. — Загл. с экрана.
4. FontExplorer® X — Professional font management software. [Электронный ресурс]. — Режим доступа: <http://www.fontexplorerx.com>. — Загл. с экрана. — Яз. англ.
5. Opcion Font Viewer — Free font viewer for Windows, Mac, and Linux. [Электронный ресурс]. — Режим доступа: <http://opcion.sourceforge.net>. — Загл. с экрана. — Яз. англ.
6. Загрузка Java для всех операционных систем [Электронный ресурс]. — Режим доступа: <http://www.java.com/ru/download/manual.jsp>. — Загл. с экрана.
7. FontFrenzy. CD Software. [Электронный ресурс]. — Режим доступа: <http://www.sdsoftware.org/tag/fontfrenzy/>. — Загл. с экрана. — Яз. англ.
8. Adobe — Adobe Type Manager Light [Электронный ресурс]. — Режим доступа: <http://www.adobe.com/products/atmlight/>. — Загл. с экрана. — Яз. англ.
9. Font Xplorer — ultimate font viewing and management software [Электронный ресурс]. — Режим доступа: <http://www.moonsoftware.com/xfplorer.asp>. — Загл. с экрана. — Яз. англ.
10. AMP Font Viewer — Complete font manager: view, organize, install, delete and print fonts lists (TrueType, OpenType and Type 1) — AMPsoft [Электронный ресурс]. — Режим доступа: <http://www.ampsoft.net/utilities/FontViewer.php>. — Загл. с экрана. — Яз. англ.

11. Сайт XFont.RU — сайт, где можно бесплатно скачать шрифты [Электронный ресурс]. — Режим доступа: <http://www.xfont.ru>.
12. Портал шрифтов RuFONT.ru. [Электронный ресурс]. — Режим доступа: <http://www.rufont.ru>. — Загл. с экрана.
13. Сайт для скачивания шрифтов FontRiver.com. [Электронный ресурс]. — Режим доступа: <http://ru.fontriver.com>.
14. MyFonts. The world's largest collection of fonts — WhatTheFont [Электронный ресурс]. — Режим доступа: <http://www.myfonts.com/WhatTheFont/>.
15. Identifont — Identify fonts by appearance, find fonts by name. [Электронный ресурс]. — Режим доступа: <http://www.identifont.com/index.html>. — Загл. с экрана. — Яз. англ.
16. Google Web Fonts [Электронный ресурс]. — Режим доступа: <http://www.google.com/webfonts>. — Загл. с экрана. — Яз. англ.
17. Видимое излучение [Электронный ресурс] // Википедия. Свободная инциклопедия — Режим доступа: http://ru.wikipedia.org/wiki/Видимое_излучение. — Загл. с экрана.
18. Самара Т. Типографика цвета. Практикум. / Тимоти Самара. — РИП-Холдинг, Rockport, 2006. — 256 с.
19. Психология цвета [Электронный ресурс]. — Режим доступа: http://www.ejevichka.holm.ru/psicho_cvet.htm.
20. Дом Солнца® — Психология цвета [Электронный ресурс]. — Режим доступа: <http://www.sunhome.ru/psychology/>.
21. Психология цвета. Символика цвета. Цвет и характер. Цвет и работоспособность [Электронный ресурс]. — Режим доступа: <http://psyfactor.org/color.htm>.
22. Астропсихологический центр «aquarun». Клар Г. Тест Люшера. Психология цвета [Электронный ресурс]. — Режим доступа: <http://www.aquarun.ru/psih/ct/ct4.html>.
23. Иттен И. Искусство цвета. / Иоханнес Иттен. — М.: Издатель Д. Аронов, 2000. — 95 с.
24. Поисковая система Яндекс [Электронный ресурс]. — Режим доступа: <http://www.yandex.ru>.
25. Adobe Kuler [Электронный ресурс]. — Режим доступа: <https://kuler.adobe.com>. — Загл. с экрана. — Яз. англ.

26. ColorScheme.Ru. Инструмент для подбора цветов и генерации цветовых схем [Электронный ресурс]. — Режим доступа: <http://colorscheme.ru>. — Загл. с экрана.
27. Color Scheme Designer 3 [Электронный ресурс]. — Режим доступа: <http://colorschemedesigner.com>. — Загл. с экрана. — Яз. англ.
28. ColoRotate [Электронный ресурс]. — Режим доступа: <http://web.colorotate.org>. — Загл. с экрана. — Яз. англ.
29. Color Palette Generator [Электронный ресурс]. — Режим доступа: <http://www.degraeve.com/color-palette/index.php>. — Яз. англ.
30. ColorMania. Advanced Color Picker Utility [Электронный ресурс]. — Режим доступа: <http://www.blacksunsoftware.com/colormaniaman.html>. — Загл. с экрана. — Яз. англ.
31. Typetester — Compare fonts for the screen [Электронный ресурс]. — Режим доступа: <http://www.typetester.org>. — Загл. с экрана.
32. Феличи Д. Типографика: шрифт, верстка, дизайн / Д. Феличи. — СПб.: БХВ-Петербург, 2004. — 496 с: ил.
33. Типографика. Что ты знаешь о ней? — Удобочитаемость [Электронный ресурс]. — Режим доступа: <http://aboutfonts.ru/2009/01/udobochitaemost/>. — Загл. с экрана.
34. Кричевский. В. Типографика в терминах и образах. / В. Кричевский. — В 2-х тт. — М.: Слово/Slovo, 2000. — Т.1. — 144 с.
35. Искусство создания и продвижения сайтов — Список стандартных шрифтов Windows [Электронный ресурс]. — Режим доступа: <http://www.fortress-design.com/spisok-standartnyh-shriftov-windows/>.
36. Microsoft typography. List of fonts supplied [Электронный ресурс]. — Режим доступа: <http://www.microsoft.com/typography/fonts/product.aspx>. — Загл. с экрана. — Яз. англ.
37. Microsoft typography. Fonts supplied with Mac OS [Электронный ресурс]. — Режим доступа: <http://www.microsoft.com/typography/fonts/mac.htm/>. — Загл. с экрана. — Яз. англ.
38. Code Style: Most common fonts for Linux and Unix. — Linux and Unix family font survey results [Электронный ресурс]. — Режим доступа: <http://www.codestyle.org/css/font-family/sampler-UnixResults.shtml>.
39. An easy way to install Microsoft's TrueType core fonts on linux [Электронный ресурс]. — Режим доступа: <http://corefonts.sourceforge.net/>. — Загл. с экрана. — Яз. англ.

40. Hyphenator — merge + pack. [Электронный ресурс]. — Режим доступа: <http://hyphenator.googlecode.com/svn/trunk/mergeAndPack.html>. — Загл. с экрана. — Яз. англ.
41. Студия Артемия Лебедева — Типограф. [Электронный ресурс]. — Режим доступа: <http://www.artlebedev.ru/tools/typograf/>.
42. Студия Артемия Лебедева — О программе «Типограф». [Электронный ресурс]. — Режим доступа: <http://www.artlebedev.ru/tools/typograf/about/>.
43. Студия Артемия Лебедева — Веб-сервис «Типограф». [Электронный ресурс]. — Режим доступа: <http://www.artlebedev.ru/tools/typograf/webservice/>.
44. Официальная таблица кодировки ISO 8859-1. [Электронный ресурс]. — Режим доступа: <http://www.unicode.org/Public/MAPPINGS/ISO8859/8859-1.txt>.
45. Unicode Standard Annex #14 — Unicode Line Breaking Algorithm. [Электронный ресурс]. — Режим доступа: <http://www.unicode.org/reports/tr14/tr14-22.html#Introduction>. — Загл. с экрана.
46. Артюхин В. В. Реальность 2.0b. Современная история информационного общества / В. В. Артюхин. М., 2011. — 432 с.
47. Правила русской орфографии и пунктуации. Полный академический справочник / Под ред. В. В. Лопатина. М.: Эксмо, 2007. — 480 с.
48. Розенталь Д. Э. Справочник по правописанию и литературной правке. / Д. Э. Розенталь. — М., Айрис-пресс, 2012, 16-е изд. — 368 с.
49. Розенталь Д. Э. Справочник по правописанию, произношению, литературному редактированию / Д. Э. Розенталь, Е. В. Джанджакова, Н. П. Кабанова. — М.: ЧеРо, 1999. [Электронный ресурс]. — Режим доступа: <http://www.eartist.narod.ru/text1/20.htm>.
50. International standard ISO 8601 Data elements and interchange formats — Information interchange — Representation of dates and times / Third edition 2004-12-01. [Электронный ресурс]. — Режим доступа: http://dotat.at/tmp/ISO_8601-2004_E.pdf. — Яз. англ.
51. ГРАМОТА.РУ — справочно-информационный интернет-портал «Русский язык». [Электронный ресурс]. — Режим доступа: <http://www.gramota.ru>. — Загл. с экрана.
52. Чебыкин Р. Разработка и оформление текстового содержания сайтов / Р. И. Чебыкин. — СПб.: БХВ-Петербург, 2004. — 528 с.: ил.



ЗАКЛЮЧЕНИЕ

Итак, дорогие читатели, вы дочитали книгу до конца! Если вы ее внимательно читали и параллельно практиковались на основе изученной информации (надеюсь, в большинстве случаев — успешно!), отвечали на контрольные вопросы в конце каждого раздела, просматривали рекомендуемую литературу, то можно вас поздравить — вы максимально эффективно поработали с этой книгой.

Коллектив авторов признателен каждому читателю за изучение изложенного в книге материала, и уверен, что полученная информация и знания помогут каждому в достижении своих целей.

Работая над курсом «Интернет-технологии», авторы ежедневно сталкиваются с все более ускоряющимся развитием мировой IT-сферы: возникновением новых технологий, постоянным усовершенствованием специализированного программного обеспечения, развитием и автоматизацией многих интернет-процессов. Поэтому в книге изложены незыблемые (по крайней мере, в ближайшее пятилетие) фундаментальные основы, которые являются отправной точкой как в ознакомлении с интернет-технологиями, так и в изучении одноименного курса.

Данная книга рассматривает важные вопросы курса, но охватывает всего лишь треть того, что авторы хотят и могут донести до вас. Остается пожелать вам терпения в ожидании новых работ коллектива авторов в ключе развития темы «Интернет-технологии».

Успехов вам, дорогие читатели!

Для заметок



Монография 1

А. Я. Аноприенко, С. В. Иваница

ПОСТБИНАРНЫЙ КОМПЬЮТИНГ И ИНТЕРВАЛЬНЫЕ ВЫЧИСЛЕНИЯ В КОНТЕКСТЕ КОДО-ЛОГИЧЕСКОЙ ЭВОЛЮЦИИ

Донецк: ДонНТУ, УНИТЕХ, 2011. — 248 с.
ISBN 978-966-8248-20-7

Монография посвящена рассмотрению закономерностей кодо-логической эволюции средств и методов компьютеринга и особенностей перехода к постбинарному компьютерингу, рассматриваемому в качестве следующего этапа в развитии средств и методов вычислений. Детально рассмотрены интервальные вычисления и особенности их постбинарной реализации.

<http://ea.donntu.org:8080/jspui/handle/123456789/7544>



Монография 2

А. Я. Аноприенко, С. В. Иваница

ТЕТРАЛОГИКА, ТЕТРАВЫЧИСЛЕНИЯ И НООКОМПЬЮТИНГ. ИССЛЕДОВАНИЯ 2010–2012

Донецк: ДонНТУ, УНИТЕХ, 2012. — 304 с.
ISBN 978-966-8248-40-5

В монографии представлены результаты исследований и разработок в области теоретического обоснования и практической реализации тетралогии как наиболее эффективного варианта реализации постбинарной логики и тетравычислений и как наиболее перспективного варианта постбинарных арифметических операций. Рассмотрены также особенности перехода к ноокомпьютерингу, раскрывающегося в качестве следующего этапа в развитии информационно-компьютерных технологий.

<http://ea.donntu.org:8080/jspui/handle/123456789/19453>



masters.donntu.org/typography

ТЕХНОПАРК ДОННТУ УНИТЕХ
В РАМКАХ КУРСА ИНТЕРНЕТ-ТЕХНОЛОГИИ

ПРЕДСТАВЛЯЕТ

для всех желающих освоить приемы профессиональной работы с текстовым содержанием сайтов!

Веб-ТИПОГРАФИКА

СЕРГЕЙ ИВАНИЦА

Искусство оформления текстов для Интернета



Веб-типографика представляет собой процесс оформления веб-текста таким образом, чтобы сделать его заметным для пользователя, акцентировать внимание на важных моментах, сделать его чтение приятным и комфортным!

Как повысить эффективность сайта и сделать его привлекательным для пользователей?

Как сделать, чтобы сайт находился в первых строчках поисковиков и приносил прибыль?

Начните с разработки качественного веб-текста — текстового содержимого сайта!

Из книги вы узнаете:

- Как профессионально оформлять веб-тексты
- Как правильно работать с веб-шрифтами и «шрифтовыми» веб-сервисами
- Как использовать цветовые модели для идеального сочетания цветов и формировать собственные цветовые схемы
- Как грамотно использовать типографские знаки и специальные символы для веб-типографики
- А также много другой полезной информации!

Книга содержит множество практических примеров кода HTML и CSS, описание работы с популярными веб-сервисами и бесплатными приложениями, примеры сайтов с профессиональной веб-типографикой.

Технологии:



Браузеры:



Операционные системы:



По вопросам получения книги обращаться по тел.

+38 050 258 71 18

или по адресу:

**ул. Артема, 58
каб. 1.311, 4.11**

Иванец Сергей Васильевич

преподаватель кафедры компьютерной инженерии Донецкого национального технического университета

web-typography@rambler.ru
sergey.ivanitsa@cs.donntu.edu.ua



АНОПРИЕНКО Александр Яковлевич
ИВАНИЦА Сергей Васильевич
ЗАВАДСКАЯ Татьяна Владимировна

ИНТЕРНЕТ-ТЕХНОЛОГИИ ДЛЯ СТУДЕНТОВ И ПРЕПОДАВАТЕЛЕЙ

Учебное пособие

Книга первая

*Редакционно-техническое оформление,
компьютерная верстка: С. В. Иваница*
Дизайн обложки: С. В. Иваница

Подписано к печати 28.12.2015 г.
Формат 70×100¹/₁₆. Бумага мелованная.
Гарнитура «Minion Pro». Печать — лазерная.
Уч.-изд. л. 22,58. Ус. печ. л. 21,17.
Заказ № 1215. Тираж 500 экз.

**Отпечатано в типографии
издательства «Донецкая политехника»
на цифровом лазерном издательском комплексе
Xerox DocuColor 2060
Тел.: +380 (62) 304-60-82**



АНОПРИЕНКО АЛЕКСАНДР ЯКОВЛЕВИЧ

Профессор кафедры «Компьютерная инженерия» Донецкого национального технического университета (ДонНТУ, г. Донецк). Автор курса «Интернет-технологии» и основатель портала магистров ДонНТУ (masters.donntu.org). Читает лекции по курсу «Интернет-технологии» с 2000 года.



ИВАНИЦА СЕРГЕЙ ВАСИЛЬЕВИЧ

Ассистент кафедры «Компьютерная инженерия» Донецкого национального технического университета (г. Донецк). Автор учебного пособия «Веб-типографика. Искусство оформления текстов для Интернета». С 2010 года ведет практические занятия, а с 2014 года — читает лекции по курсу «Интернет-технологии».



ЗАВАДСКАЯ ТАТЬЯНА ВЛАДИМИРОВНА

Доцент кафедры «Компьютерная инженерия» Донецкого национального технического университета (г. Донецк). С 2007 года читает лекции и ведет практические занятия по курсу «Интернет-технологии».

Книга основана на материалах курса «Интернет-технологии» для магистрантов, читаемого в Донецком национальном техническом университете с 2001 года.

Эта книга — первая из серии книг «Интернет-технологии для студентов и преподавателей», в которую вошел материал о веб-дизайне, веб-типографике, о работе с графической информацией в Интернете и программном обеспечении для работы с сайтами.

ISBN 978-966-8248-55-9



*Издание приурочено к 95-летию
Донецкого национального технического
университета*

Сайт ДонНТУ: donntu.org

Портал магистров ДонНТУ: masters.donntu.org

ДЛЯ ШИРОКОГО КРУГА ЧИТАТЕЛЕЙ:
СТАРШИЕ ШКОЛЬНИКИ И СТУДЕНТЫ, МАГИСТРЫ И АСПИРАНТЫ,
МОЛОДЫЕ УЧЕНЫЕ, ПРЕПОДАВАТЕЛИ ВСЕХ КВАЛИФИКАЦИОННЫХ УРОВНЕЙ