

**А. А. Дуванов**

**WEB-КОНСТРУИРОВАНИЕ.  
DHTML**

Санкт-Петербург  
«БХВ-Петербург»

2003

УДК 681.3.06  
ББК 32.973.26-018.2  
Д79

**Дуванов А. А.**

Д79 Web-конструирование. DHTML. — СПб.: БХВ-Петербург, 2003. — 512 с.: ил.

ISBN 5-94157-334-0

Это вторая книга серии "Web-конструирование". Она пополняет навыки создания гипертекстовых страниц на базе "чистого" HTML правилами построения каскадных стилевых таблиц (CSS) и основами программирования сценариев на языке JavaScript.

Подробно рассматривается объектная модель браузера, на конкретных примерах объясняется методика построения собственных программных объектов, интегрируемых в общую среду "браузер-гипертекстовое приложение". Большое внимание уделяется структуре программного кода, дисциплине программирования, тестированию и отладке сценариев.

*Для учащихся 10—11 классов и преподавателей школ*

УДК 681.3.06  
ББК 32.973.26-018.2

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Михальчук</i>
Компьютерная верстка	<i>Татьяны Олоновой</i>
Корректор	<i>Вера Александрова</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.06.03.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 41,28.

Тираж экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов  
в Академической типографии "Наука" РАН  
199034, Санкт-Петербург, 9 линия, 12.

# Содержание

<b>Предисловие</b> .....	<b>1</b>
Напутствие читателю .....	1
Гипертекстовые учебники Роботландского университета .....	2
Поддержка книги (коды примеров) .....	2
Инструментарий занятий .....	3
Устройство книги .....	4
Благодарности .....	6
<b>Введение</b> .....	<b>7</b>
Что такое DHTML? .....	7
Что такое CSS? .....	8
<b>ЧАСТЬ I. ОСНОВЫ CSS</b> .....	<b>11</b>
<b>Урок 1. CSS в HTML-коде</b> .....	<b>13</b>
Стиль для отдельного тега .....	13
Стиль для отдельного HTML-файла .....	14
Стиль для нескольких HTML-файлов .....	17
Комбинирование стилей .....	17
За что мы любим CSS? .....	21
Зачетный класс .....	21
Задания .....	22
<b>Урок 2. Обзор свойств CSS</b> .....	<b>24</b>
Единицы измерения .....	24
Шрифт .....	25
Цвет .....	28
Текст .....	29
Поля и рамки .....	31
Вид .....	32

Зачетный класс .....	33
Задания .....	34
<b>Урок 3. Основы построения CSS .....</b>	<b>35</b>
Наследование .....	35
Контекстные селекторы .....	37
Классы .....	38
Теги <i>&lt;DIV&gt;</i> и <i>&lt;SPAN&gt;</i> .....	42
Абсолютное позиционирование .....	45
<i>Z-index</i> .....	47
Каскадирование .....	48
Зачетный класс .....	49
Задания .....	52
<b>Урок 4. Позиционирование, <i>z-index</i> .....</b>	<b>54</b>
Позиционирование .....	54
Иерархия кода страницы .....	54
Абсолютное позиционирование .....	55
Относительное позиционирование .....	59
Смешанное позиционирование .....	61
<i>Z-index</i> .....	69
Зачетный класс .....	70
Задания .....	71
<b>ЧАСТЬ II. ОСНОВЫ JavaScript .....</b>	<b>75</b>
<b>Урок 5. Первый скрипт .....</b>	<b>77</b>
Классический HTML болен статичностью .....	77
JavaScript? Что это такое? .....	77
Как скрипт «монтируется» в HTML? .....	78
Зачетный класс .....	81
Задания .....	82
Дополнительный материал .....	82
<b>Урок 6. Переменные, константы, выражения .....</b>	<b>87</b>
Переменные и константы .....	87
Описание переменных .....	88
Выражения .....	89
Команды .....	89
Зачетный класс .....	91
Задания .....	94
Дополнительный материал .....	95

<b>Урок 7. Ветвления</b> .....	<b>105</b>
Ввод информации.....	105
Условная команда.....	106
Переключатель.....	108
Логические операции.....	110
Блок.....	110
Конструируем небольшой экзамен.....	111
Арифметические и строковые выражения как условия.....	113
Значения логического типа.....	114
Трудные условия.....	114
Ввод альтернативы.....	115
Зачетный класс.....	117
Задания.....	120
Дополнительный материал.....	120
<b>Урок 8. Повторения</b> .....	<b>125</b>
Команды цикла.....	125
Команды <i>break</i> и <i>continue</i> .....	127
Зачетный класс.....	129
Задания.....	131
Дополнительный материал.....	132
<b>Урок 9. Функции</b> .....	<b>136</b>
Описание и вызов функции.....	136
Проектирование «сверху вниз».....	138
Оборонительное программирование.....	145
Как браузер разбирается в типах переменных?.....	147
Зачетный класс.....	148
Задания.....	151
Дополнительный материал.....	152
<b>Урок 10. Объект = методы + свойства</b> .....	<b>159</b>
Понятие объекта.....	159
Доступ к свойствам и методам.....	161
Встроенные объекты и объекты пользователя.....	161
Зачетный класс.....	167
Дополнительный материал.....	169
<b>ЧАСТЬ III. ОБЪЕКТНАЯ МОДЕЛЬ БРАУЗЕРА</b> .....	<b>193</b>
<b>Урок 11. Объекты и события браузера</b> .....	<b>195</b>
Объект <i>window</i> .....	195
Объект <i>document</i> .....	198

События.....	200
Зачетный класс.....	205
Задания.....	205
Дополнительный материал.....	207
<b>Урок 12. Формы (кнопки, строки ввода).....</b>	<b>212</b>
Принцип программного управления.....	212
Блок <i>&lt;Form&gt;</i> .....	213
Элемент <i>&lt;INPUT&gt;</i> .....	216
Зачетный класс.....	221
Задания.....	223
Дополнительный материал.....	225
<b>Урок 13. Формы (флажки, радиокнопки).....</b>	<b>231</b>
Флажки.....	231
Радиокнопки.....	237
Зачетный класс.....	241
Задания.....	243
Дополнительный материал.....	245
<b>Урок 14. Формы (меню, многострочное поле).....</b>	<b>248</b>
Меню <i>&lt;SELECT&gt;</i> .....	248
Многострочное поле <i>&lt;TEXTAREA&gt;</i> .....	254
Зачетный класс.....	259
Задания.....	265
<b>ЧАСТЬ IV. ПЕРВЫЙ ПРАКТИКУМ НА «КУХНЕ» СИДОРОВА.....</b>	<b>267</b>
<b>Урок 15. Конструирование объектов.....</b>	<b>269</b>
Объект для хранения данных.....	269
Построение сложного объекта.....	277
Задания.....	288
<b>Урок 16. Меню.....</b>	<b>289</b>
Меню ссылок на UL.....	290
Меню ссылок на <i>&lt;SELECT&gt;</i> .....	292
Меню на стандартных кнопках.....	298
Меню на графических кнопках.....	300
Задания.....	305
<b>Урок 17. Роботландский Плюсик.....</b>	<b>307</b>
Плюсик.....	307
Техническое задание на гипертекстового Плюсика.....	309

Программирование стека .....	311
Протокол. Откатка. Накатка .....	316
Плюсик работает! .....	327
<b>ЧАСТЬ V. CSS + JAVASCRIPT .....</b>	<b>341</b>
<b>Урок 18. Программное управление стилями .....</b>	<b>343</b>
Принцип программного управления стилями .....	343
Доступ к элементам стиля .....	343
Построение динамического меню .....	347
Движение на экране .....	351
Задания .....	354
<b>Урок 19. Управление содержимым страницы .....</b>	<b>355</b>
Свойство <i>innerText</i> .....	355
Свойство <i>outerText</i> .....	356
Свойство <i>innerHTML</i> .....	356
Свойство <i>outerHTML</i> .....	357
«Крестики-нолики» .....	357
Модель _«плавающего»_ фрейма .....	363
Задания .....	366
<b>ЧАСТЬ VI. ВТОРОЙ ПРАКТИКУМ НА «КУХНЕ» СИДОРОВА .....</b>	<b>369</b>
<b>Урок 20. Второй практикум на «кухне» Сидорова: Конюх .....</b>	<b>371</b>
Техническое задание .....	372
<b>Урок 21. Стилиевые решения .....</b>	<b>376</b>
<b>Урок 22. Скрипты .....</b>	<b>389</b>
Перетаскивание элементов по экрану .....	389
Модальные окна .....	398
Полный код Конюха .....	405
<b>ЧАСТЬ VII. ПРИЛОЖЕНИЯ .....</b>	<b>407</b>
<b>Приложение 1. Справочник CSS .....</b>	<b>409</b>
Единицы измерения .....	409
Структура справочных таблиц .....	410

---

<b>Приложение 2. Справочник JavaScript .....</b>	<b>424</b>
Зарезервированные слова .....	424
Конструкции .....	424
Операции .....	435
Встроенные функции .....	444
Встроенные объекты .....	446
<b>Приложение 3. Справочник. Объекты и события браузера.....</b>	<b>467</b>
<i>window</i> .....	467
Примеры работы с объектом <i>window</i> .....	469
<i>document</i> .....	475
<i>event</i> .....	482
<i>history</i> .....	485
<i>location</i> .....	486
<i>navigator</i> .....	489
<i>screen</i> .....	491
События браузера .....	493
<b>Приложение 4. Коды испытателей.....</b>	<b>495</b>
Испытатель 1. Сравнения ( <i>Урок 9</i> ).....	495
Испытатель 2. Открытие окна ( <i>Урок 11</i> ) .....	496
Испытатель 3. Кнопка с секретом ( <i>Урок 11</i> ).....	498
Испытатель 4. <i>value</i> и <i>defaultValue</i> ( <i>Урок 12</i> ).....	500
Испытатель 5. Угадай пароль ( <i>Урок 12</i> ).....	502
<b>Литература .....</b>	<b>504</b>



Посвящается Александру Домнину, моему верному другу со школьных лет, и Олегу Какалину в память о первых совместных работах

# Предисловие

Это вторая книга серии «Web-конструирование»<sup>1</sup>. Она дополняет навыки создания гипертекстовых страниц на базе «чистого» HTML (HyperText Markup Language — язык разметки гипертекста) правилами построения каскадных стилевых таблиц (CSS, Cascading Style Sheets — каскадные таблицы стилей) и основами программирования сценариев на языке JavaScript.

Технология построения динамических гипертекстов, описанная в книге, условно изображается формулой:

**DHTML = HTML 4.0 + CSS + JavaScript + Internet Explorer (4 и старше)**

Расшифровывается это так: внедрение в гипертекстовые документы (HTML, версия 4.0) стилевых свойств (CSS) и сценариев (JavaScript). В качестве программы просмотра документов предполагается браузер Internet Explorer (версии 4 и старше).

Эта книга, как и первая, отражает многолетний опыт Роботландского университета — сетевой школы, в которой обучаются команды (группы школьников во главе с руководителем — школьным учителем) из разных регионов России и ближнего зарубежья.

## Напутствие читателю

Вам надоели статичные гипертекстовые документы? Хочется, чтобы ваша страница следила за мышью, клавиатурой, меняла экран в ответ на действия пользователя? Может быть, вы даже мечтаете создать в гипертексте игру или другую полезную программу?

Книга вам поможет. Ведь DHTML означает динамический HTML, т. е. речь пойдет о технологии, которая позволяет создавать «живые» гипертексты, практически неотличимые от обычных компьютерных программ.

Но эта книга для вас, если только вы:

- построили несколько гипертекстовых страниц, честно записывая теги HTML в текстовом редакторе;
- написали пару программ на каком-нибудь языке программирования и поняли, что это занятие вам по душе.

---

<sup>1</sup> Первая: Web-конструирование. HTML. СПб.: «БХВ-Петербург», 2003.

Что означают эти требования?

1. Если вы построили сайт в визуальном режиме, используя Word, FrontPage или другой визуальный редактор, книга вам не поможет. Для изучения DHTML нужно владеть «анатомией» гипертекстового конструирования: понимать структуру HTML-документа, правила записи тегов и их атрибутов. Можете считать себя вполне подготовленным в этом вопросе, если уже поработали с первой книгой этой серии «Web-конструирование. HTML».
2. Динамика гипертекстовых страниц и способность их к активной работе с пользователем появляются благодаря программным кодам, которые дополняют обычную разметку HTML. Книга не опирается на ваш опыт программирования, но предполагает склонность к этому роду деятельности. Если вы по принуждению написали несколько программных строк для отчета, мало что в них поняли, а главное — не почувствовали никакого душевного трепета от этих «магических заклинаний», «оживляющих» железный ящик с экраном на вашем столе, — книга не для вас. Вы только потеряете деньги и время. Посоветуйте книгу своим увлеченным друзьям.

## Гипертекстовые учебники Роботландского университета

Эта книга не просто отражает опыт сетевой школы Роботландии, она является бумажной версией двух гипертекстовых учебников.

Учебники университета особенные: они больше похожи на электронные лаборатории. На их страницах можно «дергать за веревочки» многочисленных испытателей, сдавать экзамен в Зачетном классе, запускать собственный код.

Бумажная книга, конечно, лишена интерактивности, зато читать ее гораздо комфортнее, чем тексты с экрана компьютера.

Хотя бумажная книга и построена самодостаточным образом, идеальным представляется вариант, при котором в распоряжении пользователя окажутся обе версии. Бумажный носитель вы уже держите в руках, а электронные учебники можно заказать на сайте [www.botik.ru/~robot](http://www.botik.ru/~robot) или в письме автору по адресу [kurs@robotland.botik.ru](mailto:kurs@robotland.botik.ru).

Теперь можно читать, удобно расположившись на диване, а компьютер включать только для выполнения заданий, работы с Испытателями, Зачетными классами и для создания собственных проектов.

## Поддержка книги (коды примеров)

Исходные коды примеров и испытательных стендов этой книги можно скопировать с адреса: <ftp://ftp.botik.ru/rented/robot/univer/dbook.zip>.

# Инструментарий занятий

## Браузер: Microsoft Internet Explorer (4 или старше)

Прежде всего для практической работы необходим браузер. В нем происходит отладка программ и интерпретация построенных приложений.

Когда создается сайт для Интернета, разработчику нужно иметь под рукой как можно больше разных браузеров и уж обязательно два основных: Microsoft Internet Explorer (IE) и Netscape Navigator (NN). Ведь пользователь сети может «заехать» к нам на любом транспорте! И нужно знать, что он увидит.

К сожалению (и большой головной боли Web-программистов), браузеры разных фирм интерпретируют HTML-коды, программы на JavaScript и стили CSS по-разному. Принципы построения динамических документов остаются практически одинаковыми, но в деталях появляются досадные различия.

Разветвлять начальный курс по особенностям разных браузеров кажется не слишком правильным, так же как и ограничивать изложения одними общими идеями. Поэтому автором в качестве «рабочей лошадки» выбран браузер Microsoft Internet Explorer (версии 4 или старше).

Такой выбор сделан, исходя из следующих соображений:

- браузер IE самый популярный сегодня;
- IE не надо устанавливать пользователям Windows (а таких пользователей большинство);
- с точки зрения программирования IE (как конструктор объектной модели) устроен более логично, чем, например, NN, и тем самым на нем легче учить новичка;
- если приложение создается не для Интернета, а для работы на отдельном компьютере или в локальной сети, то его можно ориентировать только на один браузер, прописав это в сопроводительной документации;
- используя полученные в книге навыки, можно легко перейти к созданию приложений, ориентированных на другие браузеры.

## Текстовый редактор

Для написания программ необходим какой-нибудь текстовый редактор. Вообще говоря, годится стандартный Блокнот Windows. Сам автор предпочитает МикроМир, а многие его коллеги с удовольствием работают в текстовом редакторе, встроенном в популярную оболочку FAR.

## Устройство книги

Материал книги изложен в шести частях.

Сначала в книге излагаются стилевые свойства тегов (*Часть I. Основы CSS*), затем основы языка JavaScript (*Часть II. Основы JavaScript*), наконец, эти два механизма суммируются, что позволяет разработчику активно воздействовать на объектную модель документа и достигать необходимого уровня динамичности и интерактивности гипертекстового приложения.

Рассматривается объектная модель браузера, на конкретных примерах объясняется методика построения собственных программных объектов, интегрируемых в общую среду «браузер-гипертекстовое приложение». Большое внимание уделяется структуре программного кода, дисциплине программирования, тестированию и отладке сценариев.

Детально, начиная с технического задания, конструируются избранные исполнители системы «Роботландия» (курс информатики для малышей). Определяется дизайн изделия, строятся стилевые таблицы, разрабатываются сценарии.

Материал книги разбит на уроки. Урок — это одна тема. Он, как правило, завершается Зачетным классом (набором контрольных вопросов по изученной теме) и пакетом заданий для выполнения практических работ. Изучение книжного урока может занять один или несколько школьных уроков (если работать с книгой в школе): это зависит от уровня подготовки школьников и времени, отводимого на обучение. В некоторых уроках кроме основного (минимального) слоя присутствует дополнительный материал (выделяемый в отдельный подраздел). Он более подробно раскрывает тему и затрагивает методические вопросы. Дополнительный материал ориентирован в первую очередь на учителя, но будет полезен и любому другому заинтересованному читателю.

*Часть I. Основы CSS* содержит четыре урока, в которых рассматриваются стилевые свойства тегов, способы построения и внедрения стилевых определений.

Это самый простой раздел книги. В нем нет программ на JavaScript, обращений к объектной модели документа, но и этот простой стилиевой инструментарий открывает разработчику новые возможности визуального представления документа на экране, которые несвойственны «чистому» HTML.

*Часть II. Основы JavaScript* включает шесть уроков, в которых излагаются основы программирования и правила записи программ на языке JavaScript.

*Часть III. Объектная модель браузера* содержит четыре урока. Вы познакомитесь с программной моделью, которую строит браузер для каждого гипер-

текстового документа, и изучите способы воздействия на эту модель при помощи сценариев на JavaScript.

Главной иллюстрирующей темой здесь выступают способы управления элементами гипертекстовых форм (кнопки, меню, поля ввода).

*Часть IV. Первый практикум на «кухне» Сидорова* состоит из трех уроков, в которых молодой программист Иван Сидоров вместе со своими коллегами излагает принципы ООП (объектно-ориентированного программирования). Теория подкреплена многочисленными примерами, которые имеют практическую ценность и могут быть использованы при создании динамических гипертекстовых приложений. Разговор на «кухне» завершается построением Плюсика — роботландского исполнителя. Это достаточно сложное приложение. Подобные примеры авторы пособий обычно «анатомируют», показывая устройство готового продукта. Сидоров нарушает эту традицию. Он выступает не в роли лектора, а в роли партнера читателя. Проект создается с чистого листа, на который сначала записывается техническое задание. Показаны все этапы работы проектировщика, в том числе построение специальных отладочных стендов и интерфейсные улучшения продукта на завершающей стадии работы.

*Часть V. CSS + JavaScript* включает рассмотрение в двух уроках способов программного управления стилевыми свойствами тегов и содержимым гипертекстовой страницы.

*Часть VI. Второй практикум на «кухне» Сидорова* состоит из трех уроков, в которых детально, начиная с технического задания, конструируется роботландский исполнитель Конюх. Определяется дизайн изделия, строятся стилевые таблицы, разрабатываются сценарии. Отладка продукта выполняется на испытательных стендах, специально построенных для этой цели.

*Приложения 1, 2, 3* содержат достаточно подробные (но далеко не полные) справочники по свойствам CSS, синтаксису и объектам языка JavaScript, объектной модели браузера. Помимо справочного материала, в разделе собраны многочисленные примеры, которые помогают понять смысл описываемых конструкций.

В *Приложение 4 (Коды испытателей)* собраны коды испытателей, на которые есть ссылки со страниц книги. Рабочие файлы испытателей (вместе с файлами примеров) можно скопировать с адреса: **<ftp://ftp.botik.ru/rented/robot/univer/dbook.zip>**.

В списке *Литературы* приводятся ссылки на дополнительные источники для тех, кто собирается продолжить практику конструирования динамических гипертекстов.

## Благодарности

Автор с радостью вспоминает тот счастливый миг, когда его посетила идея попросить Якова Наумовича Зайдельмана, своего давнего товарища, прочесть рукопись с «критическим пером» в руке. Острота этого «фирменного» инструмента известна еще со времен совместной работы над школьным курсом «Роботландия».

Правда, количество замечаний и претензий оказалось больше, чем автор ожидал, и ему пришлось заново переписывать отдельные фрагменты книги, придумывать новые учебные примеры, переделывать программные коды.

Одним словом, Яков Наумович нагрузил автора дополнительной работой (с которой автор едва справился к сроку сдачи книги в издательство), и читатели должны почувствовать большую симпатию к этому человеку, ибо он непримиримо стоял на страже их интересов.

# Введение

## Что такое DHTML?

Суть построения «живых» гипертекстовых приложений достаточно проста. Браузер как интерпретатор HTML-кода реализует две функции: визуальную и моделирующую. Визуальное действие очень заметно: браузер отображает документ на экране. Но на этом работа не заканчивается. Браузер строит в памяти компьютера программную модель документа, в которой каждому тегу из HTML-файла соответствует отдельный объект.

Посмотрите на схему (рис. 1). Построение объектной модели документа (она носит иерархический характер) браузер начинает с объекта `window`. В этом объекте описаны свойства окна, в котором показан документ. В состав объекта `window` входит объект `document`, который соответствует контейнеру `<HTML>...</HTML>`. Объект `document` включает в себя (среди прочих) объект-массив `images`, который содержит информацию обо всех тегах `<IMG>` исходного HTML-файла. В частности, первая картинка описывается в этом массиве объектом `images[0]`, вторая — объектом `images[1]` и так далее.

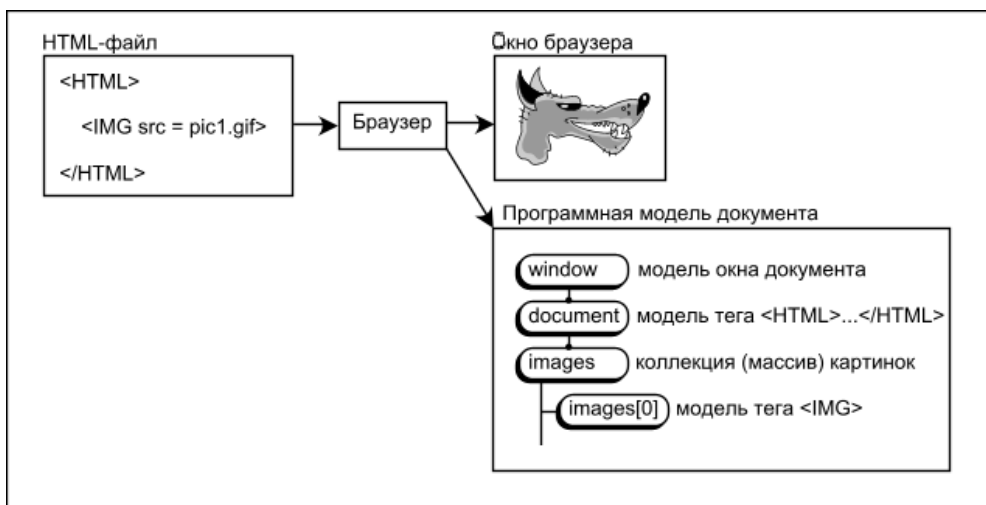
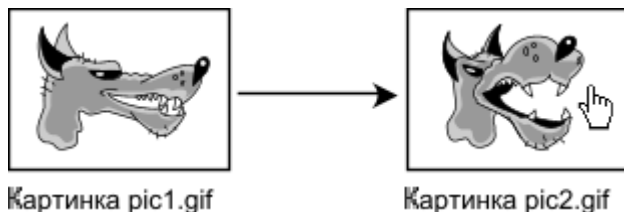


Рис. 1. Браузер строит изображение и объектную модель документа

Документ на экране станет «живым», если в HTML-коде предусмотреть не только описание тегов, но и воздействия на объектную модель при возникновении того или иного события. Можно, например, при наступлении события «курсор мыши над картинкой» изменить в объекте `images[0]` свойство `src` с `pic1.gif` на `pic2.gif` (то есть заменить одну картинку другой). И мы увидим, как волк сердится на компьютерную мышь (рис. 2).



**Рис. 2.** Замена изображения во время просмотра документа

Программные коды, оживляющие гипертекст (их называют «скриптами», или «сценариями»), записываются на специальных языках программирования. Самым популярным языком, коды которого можно встраивать в HTML-тексты, является JavaScript. Именно этот язык рассмотрен в книге и именно он использован в ней как инструмент создания динамических интерактивных приложений.

## Что такое CSS?

CSS — Cascading Style Sheets (каскадные таблицы стилей) — это средство, позволяющее отделить содержание гипертекстовых страниц от формы их представления на экране, а в паре со сценариями — достичь такой динамичности и интерактивности, при которых гипертекстовое приложение становится практически неотличимо от мультимедийного продукта, созданного в профессиональных программистских оболочках.

Технология CSS допускает, например, абсолютное позиционирование, т. е. позволяет выводить элементы на экран, используя реальные координаты. Программное изменение координат в объектной модели вызывает изменение положения элемента на экране.

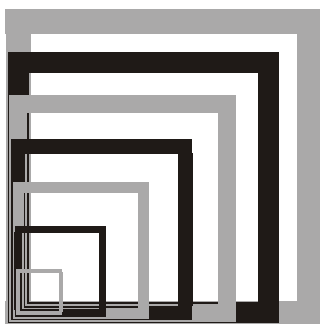
Наряду с двумя координатами  $x$  и  $y$  при помощи CSS можно задавать еще и третью координату —  $z$ -индекс. Третья координата определяет номер слоя, в который помещается элемент. Таким образом, при движении в многослойном пространстве одни элементы могут проходить над или под другими.



Но даже при проектировании статичных гипертекстов технология CSS весьма полезна:

- CSS позволяет сократить объем HTML-файла за счет исключения из кода страницы тегов и атрибутов, ответственных за визуальное представление элементов на экране (теги: `<FONT>`, `<CENTER>`, `<B>`, `<I>`, `<S>`, `<U>` и т. д. ; атрибуты: `align`, `color`, `bgcolor`, `size`, `width`, `height` и т. д.);
- CSS предлагает гораздо больше средств для управления внешним видом элемента, чем обычные теги и атрибуты HTML;
- разработчик получает возможность менять внешний вид сотни гипертекстовых страниц, внося правки только в один файл с описаниями стилевых свойств.





# Часть I

## ОСНОВЫ CSS

**Урок 1.** CSS в HTML-коде

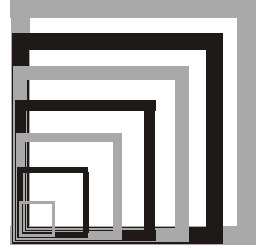
**Урок 2.** Обзор свойств CSS

**Урок 3.** Основы построения CSS

**Урок 4.** Позиционирование, *z-index*



# Урок 1



## CSS в HTML-коде

### Стиль для отдельного тега

CSS — Cascading Style Sheets (каскадные таблицы стилей) — это средство, позволяющее задавать различные визуальные свойства HTML-тегам.

Можно указать, например, как должен выглядеть на экране конкретный абзац `<P>`:

```
<P style="font-size:1.5cm; color:green">
```

К этому абзацу применено стилевое определение.

Стиль задается атрибутом `style`. В приведенном примере браузеру дано указание вывести абзац зелеными буквами размером в 1.5 сантиметра. Стилиевые определения задаются при помощи указаний (рис. 1.1)



Рис. 1.1. Формат стилиевых определений

Указания отделяются друг от друга символом «;».

Атрибут `style` можно использовать практически в каждом теге, задавая его специфические свойства. Посмотрите еще один пример внедрения стилей в теги.

Страница на рис. 1.2 построена таким кодом (листинг 1.1).

#### Листинг 1.1

```
<HTML>  
<HEAD>  
<META http-equiv="Content-Type"
```

```

        content="text/html; charset=windows-1251">
<TITLE>Стиль для отдельного тега</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
<H2>Стиль для отдельного тега</H2>
<HR>
<UL>
  <LI>Это обычный текст.
  <LI style="color:red;font-size:1cm;font-style:italic">
    Красные наклонные буквы высотой в 1 сантиметр.
  <LI style="margin-left:2cm">
    Отступ слева в 2 сантиметра.
</UL>
</BODY>
</HTML>

```

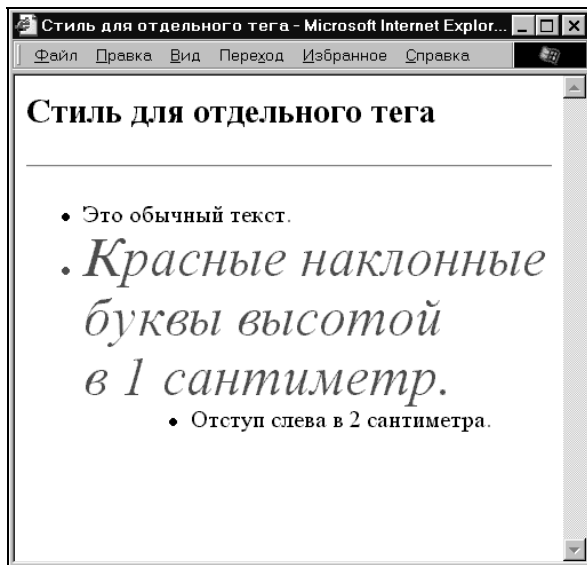


Рис. 1.2. Стиль для одного тега

## Стиль для отдельного HTML-файла

Можно задавать стиль для тега или группы тегов так, чтобы определение работало на протяжении всего HTML-документа. Например, можно указать вид всех заголовков. Для этого достаточно написать определение (листинг 1.2) в головной части документа.

**Листинг 1.2**

```
<HTML>
<HEAD>
  <META http-equiv="Content-Type"
    content="text/html; charset=windows-1251">
  <TITLE>Стиль для отдельного файла</TITLE>
  <STYLE type="text/css">
    <!--
      H1, H2, H3, H4, H5, H6
      {
        text-align:right;
        color:maroon;
        font-family: "Arial Cyr", Geneva, Helvetica, sans-serif;
      }
    -->
  </STYLE>
</HEAD>
<BODY bgcolor=#DFFD05 text=black>
  <H2>Стиль для отдельного файла</H2>
  <HR>
  <P>Это обычный текст
  <H3>Это заголовок</H3>
  <P>Это снова обычный текст
</BODY>
</HTML>
```

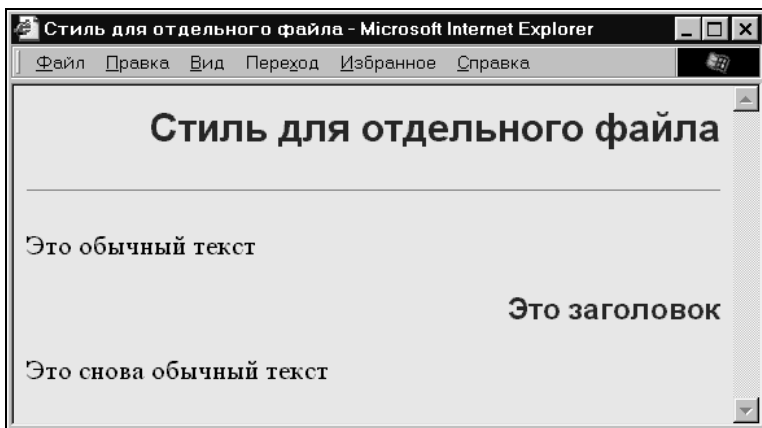
Браузер отображает заголовки в этом документе рубленным шрифтом цвета maroon и выравнивает их по правому краю экрана (рис. 1.3).

Такое поведение браузера соответствует инструкции (листинг 1.3).

**Листинг 1.3**

```
<STYLE type="text/css">
  <!--
    H1, H2, H3, H4, H5, H6
    {
      text-align:right;
      color:maroon;
      font-family: "Arial Cyr", Geneva, Helvetica, sans-serif;
    }
  -->
```

```
-->
</STYLE>
```



**Рис. 1.3.** Стиль для отдельного файла

Стилевые определения (*селекторы*) располагаются внутри блока `<STYLE>...</STYLE>` и «запаковываются» в HTML-комментарий (для браузеров, которые не поддерживают CSS).

Стилевое определение имеет вид:

Имя тега (или имена тегов через запятые)

```
{
  характеристика: величина;
  ...
  характеристика: величина;
}
```

В приведенном выше примере использованы три характеристики:

<code>text-align:right;</code>	—	задает выравнивание по левому краю;
<code>color:maroon;</code>	—	задает цвет maroon;
<code>font-family:"Arial Cyr", Geneva, Helvetica,sans-serif;</code>	—	задает рубленый шрифт.

Заголовки будут выводиться шрифтом Arial Cyr, если, конечно, этот шрифт есть в компьютере пользователя. Если шрифта нет, браузер последовательно ищет шрифты Geneva, Helvetica или, в конце концов, какой-нибудь рубленый шрифт (указание «sans-serif»). В случае полной неудачи браузер выведет текст «шрифтом по умолчанию», т. е., как правило, шрифтом Times New Roman.



## Стиль для нескольких HTML-файлов

Обычной практикой является указание стилей в отдельном файле. Для таких файлов используют расширение `css`. Например, можно в файле `style.css` записать:

```
BODY {margin-left:40px;}
H1,H2,H3,H4,H5,H6
{
    text-align:right;
    color:maroon;
    font-family:"Arial Cyr",Geneva,Helvetica,sans-serif;
}
```

Для подключения этих указаний в разделе `<HEAD>...</HEAD>` HTML-файла нужно поместить ссылку:

```
<LINK rel=stylesheet type="text/css" href=style.css>
```

Такой способ расположения стилевых определений очень удобен. На один и тот же стилиевой файл могут ссылаться многие HTML-документы. Изменения в этом единственном файле скажутся на внешнем виде десятка (а то и сотни) страниц.

Обратите внимание на стилевое указание `BODY {margin-left:40px;}`.

Оно задает экранный отступ слева в 40 пикселей для всего документа в целом. В силу такого определения элементы не будут «наползать» на вертикальную фоновую полосу, моделирующую левое страничное поле. Как видите, для задания страничного отступа можно обойтись без таблиц и «распорок», о которых говорилось в книге «Web-конструирование. HTML».

## Комбинирование стилей

Были показаны три способа внедрения стилей в HTML-коды:

- указания в отдельном теге;
- указания в заголовке HTML-файла;
- указания в отдельном CSS-файле.

А что если комбинировать эти способы? Какой из них окажется самым «сильным» для конкретного тега?

Проделаем серию опытов.

## Документ без CSS-указаний

В примере, код которого приводится ниже (листинг 1.4, рис. 1,4), заголовки выводятся черным цветом по белому.

### Листинг 1.4

```
<HTML>
  <HEAD>
    <TITLE>Пример 1</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H1>Заголовок 1</H1>
    <H2>Заголовок 2</H2>
    <H3>Заголовок 3</H3>
  </BODY>
</HTML>
```

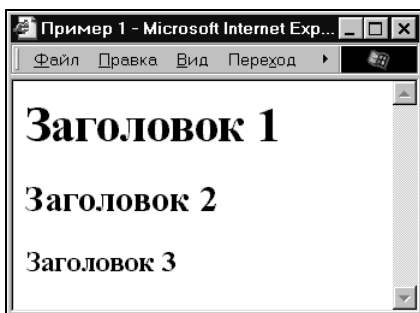


Рис. 1.4. Документ без CSS-указаний

## CSS-указания в отдельном теге

Этот код (листинг 1.5) отобразит первые два заголовка черным цветом, а последний — красным.

### Листинг 1.5

```
<HTML>
  <HEAD>
    <TITLE>Пример 2</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H1>Заголовок 1</H1>
```

```
<H2>Заголовок 2</H2>
<H3 style="color:red">Заголовок 3</H3>
</BODY>
</HTML>
```

## CSS-указания в <HEAD>

Здесь (листинг 1.6) первые два заголовка выводятся синим цветом, а последний — красным.

### Листинг 1.6

```
<HTML>
<HEAD>
  <STYLE type="text/css">
    <!--
      H1,H2,H3
      {
        color:blue;
      }
    -->
  </STYLE>
  <TITLE>Пример 3</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
  <H1>Заголовок 1</H1>
  <H2>Заголовок 2</H2>
  <H3 style="color:red">Заголовок 3</H3>
</BODY>
</HTML>
```

## CSS-указания в CSS-файле

Файл prim.css содержит:

```
H1,H2,H3
{
  color:green;
}
```

HTML-файл имеет вид (листинг 1.7).

### Листинг 1.7

```
<HTML>
  <HEAD>
    <!-- Подключение файла prim.css. -->
    <LINK rel=stylesheet type="text/css" href=prim.css>
    <STYLE type="text/css">
      <!--
        H1, H2, H3
        {
          color:blue;
        }
      -->
    </STYLE>
    <TITLE>Пример 4</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H1>Заголовок 1</H1>
    <H2>Заголовок 2</H2>
    <H3 style="color:red">Заголовок 3</H3>
  </BODY>
</HTML>
```

Код отобразит первые два заголовка синим цветом, а последний — красным. Рассмотрим другой порядок следования тегов `<STYLE>` и `<LINK>` (листинг 1.8).

### Листинг 1.8

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      <!--
        H1, H2, H3
        {
          color:blue;
        }
      -->
    </STYLE>
```

```
<!-- Подключение файла prim.css. -->
<LINK rel=stylesheet type="text/css" href=prim.css>
<TITLE>Пример 5</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
  <H1>Заголовок 1</H1>
  <H2>Заголовок 2</H2>
  <H3 style="color:red">Заголовок 3</H3>
</BODY>
</HTML>
```

Код отобразит первые два заголовка зеленым цветом, а последний — красным.

## За что мы любим CSS?

Используя CSS можно:

- задавать поля, отступы, размер и тип шрифта, цвета текста и фона для отдельных элементов страницы (абзацев, слов, букв), оформлять красные строки, буквицы. В CSS представлен ряд свойств, с помощью которых можно создавать выпадающие меню, накладывать один элемент на другой, заставлять текст «отбрасывать тень», проявляться в «рентгеновских лучах» и использовать другие эффекты. Можно выводить элементы на экран браузера с точностью до одного пиксела, динамически перемещать их по экрану в разных слоях (одни элементы будут проходить над или под другими). Иными словами, CSS вместе с JavaScript позволяет монтировать на экране браузера приложение, практически ничем не отличающееся от компьютерной программы, написанной на профессиональном языке программирования, таком, например, как Си;
- изменять оформление целого сайта, состоящего из сотен файлов, не прикасаясь к HTML-коду, редактируя всего лишь один CSS-файл;
- уменьшать количество тегов в HTML-документе, отделяя информационное наполнение HTML-файла от его внешнего представления на экране браузера.

## Зачетный класс

1. Как расшифровывается аббревиатура CSS?
2. Опишите назначение технологии CSS.
3. Каким атрибутом можно задать стиль в теге?

4. Какие из нижеприведенных записей задают красный цвет текста в абзаце?
  - а) `<P color=red>`;
  - б) `<P color=#FF0000>`;
  - в) `<P color=rgb(255,0,0)>`;
  - г) `<P style="color:red">`;
  - д) `<P style="color:#FF0000">`;
  - е) `<P style="color:rgb(255,0,0)">`.
5. В каком специальном HTML-блоке `<?????>...</?????>` размещают стилевые описания?
6. Какие из нижеприведенных стилевых описаний не содержат ошибок?
  - а) `P {color:red; font-size=1cm};`
  - б) `P {color:red, font-size=1cm};`
  - в) `P {color:red font-size:1cm};`
  - г) `P {color:red, font-size:1cm};`
  - д) `P {color:red; font-size:1cm}.`
7. В каком теге записывается ссылка на файл со стилевыми определениями, каким атрибутом она задается? Запишите эти имена вместо вопросительных знаков: `<???? rel=stylesheet type="text/css" ????=file.css>`
8. Какие три способа задания стилевых описаний вам известны? Сравните эти способы и укажите назначение каждого из них.

## Задания

1. После внимательного изучения примеров урока ответьте на следующие вопросы:
  - Какое CSS-указание главнее: описанное в отдельном теге или размещенное в разделе `<HEAD>` HTML-программы?
  - Какое указание главнее: описанное в разделе `<HEAD>` HTML-программы или размещенное в отдельном CSS-файле и связанное с документом при помощи тега `<LINK>`? Зависит ли результат от порядка следования этих предписаний?
2. Используя стили, постройте документ, изображенный на рис. 1.5.  
В этом документе:
  - основные цвета: черный текст на белом фоне;
  - отступ всех элементов на странице слева и справа равен 2 см;