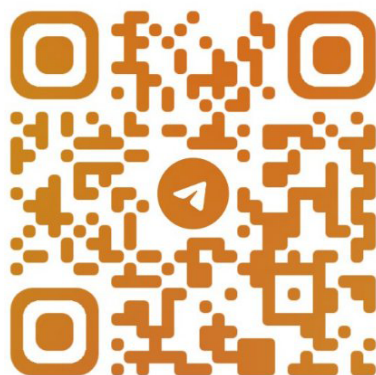
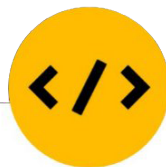


JavaScript[®]

ДЛЯ
ЧАЙНИКОВ[®]



@CODELIBRARY_IT

JavaScript®

FOR
DUMMIES®

A Wiley Brand

by Chris Minnick and Eva Holland

FOR
DUMMIES®
A Wiley Brand

JavaScript[®]

ДЛЯ
ЧАЙНИКОВ[®]

Крис Минник, Ева Холланд



ДИАЛЕКТИКА

Москва • Санкт-Петербург • Киев
2017

ББК 32.973.26-018.2.75

М62

УДК 681.3.07

Компьютерное издательство “Диалектика”

Главный редактор *С.Н. Тригуб*

Зав. редакцией *В.Р. Гинзбург*

Перевод с английского и редакция канд. хим. наук *А.Г. Гузиковича*

По общим вопросам обращайтесь в издательство “Диалектика” по адресу:

info@dialektika.com, <http://www.dialektika.com>

Минник, Крис, Холланд, Ева.

М62 JavaScript для чайников. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2017. — 320 с.
: ил. — Парал. тит. англ.

ISBN 978-5-8459-2036-2 (рус.)

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства John Wiley & Sons, Inc.

Copyright © 2017 by Dialektika Computer Publishing.

Original English edition Copyright © 2015 by John Wiley & Sons, Inc.

All rights reserved including the right of reproduction in whole or in part in any form. This translation is published by arrangement with John Wiley & Sons, Inc.

Научно-популярное издание

Крис Минник, Ева Холланд

JavaScript для чайников

В издании использованы карикатуры Банка изображений Cartoonbank.ru

Литературный редактор *И.А. Попова*

Верстка *Л.В. Чернокозинская*

Художественный редактор *Е.П. Дынник*

Корректор *Л.А. Гордиенко*

Подписано в печать 15.03.2017. Формат 70х100/16

Усл. печ. л. 25,8. Уч.-изд. л. 14,98

Доп. тираж 700 экз. Заказ № 1827.

Отпечатано в АО «Первая Образцовая типография»

Филиал «Чеховский Печатный Двор»

142300, Московская область, г. Чехов, ул. Полиграфистов, д. 1

Сайт: www.chpd.ru, E-mail: sales@chpd.ru, тел. 8(499)270-73-59

ООО “И. Д. Вильямс”, 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-2036-2 (рус.)

ISBN 978-1-119-05607-2 (англ.)

© 2017, Компьютерное изд-во “Диалектика”,
перевод, оформление, макетирование

© 2015 by John Wiley & Sons, Inc.

Оглавление

Об авторах	15
Введение	17
Часть I. Приступаем к программированию на JavaScript	21
Глава 1. Самый неправильно понятый язык программирования в мире	23
Глава 2. Моя первая программа на JavaScript	35
Глава 3. Работа с переменными	55
Глава 4. Массивы	69
Глава 5. Операторы, выражения, инструкции	79
Глава 6. Циклы и ветвление кода	92
Часть II. Организация программ на JavaScript	105
Глава 7. Приобретаем функциональность	107
Глава 8. Создание и использование объектов	124
Часть III. JavaScript в Интернете	137
Глава 9. Управление браузером с помощью объекта Window	139
Глава 10. Манипулирование документами с помощью DOM	152
Глава 11. Использование событий в JavaScript	169
Глава 12. Интеграция ввода и вывода данных	179
Глава 13. Работа с CSS и графикой	192
Часть IV. Дополнительные темы	207
Глава 14. Поиск с использованием регулярных выражений	209
Глава 15. Функции обратного вызова и замыкания	220
Глава 16. Приветствуем AJAX и JSON	230
Часть V. JavaScript и HTML5	245
Глава 17. Программные интерфейсы HTML5	247
Глава 18. Библиотека jQuery	262
Часть VI. Великолепные десятки	279
Глава 19. Десять JavaScript-фреймворков и библиотек, которые вам следует изучить	281
Глава 20. Десять самых распространенных ошибок в JavaScript-программах и как их избежать	292
Глава 21. Десять онлайн-инструментов, которые улучшат качество создаваемых вами программ на JavaScript	301
Предметный указатель	310

Содержание

Об авторах	15
Введение	17
О чем эта книга	17
Исходные предположения	18
Пиктограммы, используемые в книге	19
Дополнительные материалы	19
Что дальше	19
Ждем ваших отзывов!	20
Часть I. Приступаем к программированию на JavaScript	21
Глава 1. Самый неправильно понятый язык программирования в мире	23
Что такое JavaScript?	23
И тут пришел Эйх...	24
Сначала был Mocha	24
Даешь больше эффектов!	25
JavaScript взрослеет	26
Динамический язык сценариев	26
Что можно делать с помощью JavaScript?	28
Почему именно JavaScript?	28
Язык JavaScript легок в изучении	29
Где работает JavaScript? JavaScript работает везде!	30
JavaScript — мощный язык	33
JavaScript — востребованный язык	34
Глава 2. Моя первая программа на JavaScript	35
Настройка среды разработки	35
Загрузка и установка браузера Chrome	36
Загрузка и установка редактора исходного кода	37
Оформление JavaScript-кода	44
Выполнение JavaScript в окне браузера	45
Использование JavaScript в атрибутах событий HTML-элементов	45
Использование JavaScript в элементе <code>script</code>	46
Включение внешних JavaScript-файлов	48
Использование консоли разработчика JavaScript	51
Комментирование кода	52

Глава 3. Работа с переменными	55
Понятие переменной	55
Объявление переменных	57
Глобальные и локальные области видимости	58
Именованние переменных	60
Создание констант с помощью ключевого слова <code>const</code>	62
Работа с типами данных	62
Числовой тип данных	62
Строковый тип данных	65
Булев тип данных	67
Тип данных <code>NaN</code>	68
Тип данных <code>undefined</code>	68
Глава 4. Массивы	69
Создание списка	69
Основные сведения о массивах	70
Отсчет индексов в массивах ведется от нуля	71
В массивах могут храниться данные любого типа	72
Создание массивов	72
Использование ключевого слова <code>new</code>	72
Литеральное определение массива	72
Заполнение массивов значениями	73
Многомерные массивы	73
Доступ к элементам массива	75
Перемещение по элементам массива в цикле	76
Свойства массивов	76
Методы для работы с массивами	77
Использование методов для работы с массивами	77
Глава 5. Операторы, выражения, инструкции	79
Выражения	79
Знакомство с операторами	79
Приоритет операторов	80
Типы операторов	83
Операторы присваивания	83
Операторы сравнения	84
Арифметические операторы	84
Строковый оператор	86
Поразрядные операторы	86
Логические операторы	88
Специальные операторы	89
Объединение операторов	90

Глава 6. Циклы и ветвление кода	92
Ветвление кода	92
if...else	92
Инструкция switch	94
Циклы	96
for	96
for...in	98
Цикл while	101
Цикл do...while	101
break и continue	102
 Часть II. Организация программ на JavaScript	105
Глава 7. Приобретаем функциональность	107
Роль функций	107
Терминология функций	108
Определение функции	109
Заголовок функции	109
Тело функции	109
Вызов функции	109
Определение параметров и передача аргументов	109
Возврат значения	110
Преимущества использования функций	110
Написание функций	113
Возврат значений	114
Передача и использование аргументов	115
Передача аргументов по значению	116
Передача аргументов по ссылке	117
Вызов функции с неполным числом аргументов	117
Аргументы по умолчанию	117
Вызов функции с количеством аргументов, превышающим количество параметров	118
Получение значений аргументов с помощью объекта arguments	118
Область видимости функции	119
Анонимные функции	120
Различия между анонимными и именованными функциями	120
Самовыполняющиеся анонимные функции	120
Сделайте это снова с помощью рекурсии	121
Функции, объявленные в других функциях	122
 Глава 8. Создание и использование объектов	124
Объект моих желаний	124
Создание объектов	125

Определение объектов с помощью объектных литералов	125
Определение объектов с помощью конструктора <code>Object()</code>	126
Получение и установка свойств объектов	127
Точечная нотация	127
Скобочная нотация	128
Удаление свойств	129
Работа с методами	129
Использование ключевого слова <code>this</code>	131
Объектно-ориентированный способ разбогатеть: наследование	132
Создание объектов с помощью конструкторов	133
Видоизменение объектного типа	134
Создание объектов с помощью метода <code>Object.create()</code>	135
Часть III. JavaScript в Интернете	137
Глава 9. Управление браузером с помощью объекта <code>Window</code>	139
Браузерная среда	139
Пользовательский интерфейс	140
Загрузчик	140
Синтаксический анализ HTML-документа	142
Синтаксический анализ CSS-стилей	142
Синтаксический анализ JavaScript	142
Компоновка и визуализация	142
Взаимодействие с BOM	143
Объект <code>Navigator</code>	143
Объект <code>Window</code>	145
Использование методов объекта <code>window</code>	150
Глава 10. Манипулирование документами с помощью DOM	152
Что такое DOM	152
Отношения между узлами	154
Использование свойств и методов объекта <code>Document</code>	158
Использование свойств и методов объекта <code>Element</code>	159
Работа с содержимым элементов	162
Свойство <code>innerHTML</code>	163
Установка значений атрибутов	163
Получение элемента по его идентификатору, имени тега или классу	164
Метод <code>getElementById()</code>	164
Метод <code>getElementsByTagName()</code>	165
Метод <code>getElementsByClassName()</code>	165
Использование свойств объекта <code>Attribute</code>	166
Создание и присоединение элементов	167
Удаление элементов	167

Глава 11. Использование событий в JavaScript	169
События	169
Обработка событий	171
Встроенные обработчики событий	171
Обработка событий с использованием свойств элементов	172
Обработка событий с использованием метода <code>addEventListener()</code>	173
Отмена распространения событий	177
Глава 12. Интеграция ввода и вывода данных	179
HTML-формы	179
Элемент <code>form</code>	179
Элемент <code>label</code>	181
Элемент <code>input</code>	181
Элемент <code>select</code>	183
Элемент <code>textarea</code>	183
Элемент <code>button</code>	184
Работа с объектом <code>Form</code>	184
Использование свойств объекта <code>Form</code>	184
Использование методов объекта <code>Form</code>	186
Доступ к элементам формы	187
Получение и установка значений элементов формы	188
Проверка пользовательского ввода	189
Глава 13. Работа с CSS и графикой	192
Использование объекта <code>Style</code>	192
Получение текущего стиля элемента	193
Установка стилевых свойств	196
Анимация элементов с помощью объекта <code>Style</code>	196
Работа с изображениями	199
Использование объекта <code>Image</code>	200
Создание трансформируемых кнопок	200
Увеличение размеров изображения при наведении на него указателя мыши	201
Создание слайд-шоу	202
Использование анимационных свойств объекта <code>Style</code>	204
Часть IV. Дополнительные темы	207
Глава 14. Поиск с использованием регулярных выражений	209
Поиск текста с помощью регулярных выражений	209
Создание регулярных выражений	211
Использование объекта <code>RegExp</code>	211
Литеральные регулярные выражения	213
Тестирование регулярных выражений	214
Специальные символы в регулярных выражениях	214

Использование модификаторов	216
Использование регулярных выражений в коде	216
Глава 15. Функции обратного вызова и замыкания	220
Что такое функции обратного вызова	220
Функции в роли аргументов	220
Написание функций, использующих функции обратного вызова	221
Использование именованных функций обратного вызова	222
Замыкания	224
Использование замыканий	228
Глава 16. Приветствуем AJAX и JSON	230
Закулисная работа AJAX	230
Примеры применения AJAX	231
Детальное ознакомление с работой AJAX	232
Использование объекта XMLHttpRequest	236
Политика одинакового источника	238
CORS — серебряная пуля AJAX-запросов	240
Передача объектов с помощью JSON	241
Часть V. JavaScript и HTML5	245
Глава 17. Программные интерфейсы HTML5	247
Как работают API	247
Проверка браузерной поддержки программных интерфейсов HTML5	248
Знакомство с программными интерфейсами HTML5	249
Использование HTML5 Geolocation API	251
Что такое геолокация	251
Как работает геолокация	251
Применение геолокации	252
Сочетание геолокации с картами Google	254
Работа со звуком и видео	258
Глава 18. Библиотека jQuery	262
Меньше кода, больше дела	262
Приступаем к работе с jQuery	262
Объект jQuery	264
Готов ли документ к работе?	265
Использование селекторов jQuery	265
Изменение документа с помощью jQuery	266
Получение и установка значений атрибутов	266
Изменение стилей CSS	266
Манипулирование элементами в DOM	267
События	268

Использование метода <code>on()</code> для подключения событий	268
Открепление событий с помощью метода <code>off()</code>	270
Привязка событий к еще не существующим элементам	270
Другие методы для работы с событиями	271
Эффекты	271
Базовые эффекты	272
Эффекты затухания	272
Эффекты скольжения	272
Задание аргументов анимационных методов	273
Создание пользовательских анимационных эффектов с помощью метода <code>animate()</code>	273
Пример выполнения анимации средствами jQuery	274
AJAX	275
Использование метода <code>ajax()</code>	275
Сокращенные методы для работы с AJAX	276
Часть VI. Великолепные десятки	279
Глава 19. Десять JavaScript-фреймворков и библиотек, которые вам следует изучить	281
AngularJS	281
Backbone.js	283
Ember.js	284
Famo.us	285
Knockout	285
QUnit	286
Underscore.js	287
Modernizr	288
Handlebars.js	289
jQuery	290
Глава 20. Десять самых распространенных ошибок в JavaScript-программах и как их избежать	292
Путаница с оператором сравнения	293
Избегайте неправильного использования оператора присваивания	293
Как избежать подводных рифов сравнений	293
Непарные скобки	294
Несоответствие кавычек	295
Отсутствующие скобки	295
Отсутствие точки с запятой	296
Ошибки, связанные с неправильным использованием регистра букв	296
Ссылки на код, не успевший загрузиться	296
Плохие имена переменных	298

Ошибки, связанные с неправильным использованием областей видимости переменных	299
Пропуск параметров при вызове функций	299
Подсчитываем ошибки: забывчивость в отношении отсчета индексов от нуля	299
Глава 21. Десять онлайн-инструментов, которые улучшат качество создаваемых вами программ на JavaScript	301
JSLint	301
JSFiddle.net	302
JSBin	303
javascriptcompressor.com	303
jsbeautifier.org	305
Генератор регулярных выражений JavaScript RegEx	306
JSONformatter	306
jshint.com	307
Сайт Mozilla Development Network	308
Дуглас Крокфорд	309
Предметный указатель	310

Об авторах

Крис Минник — писатель, преподаватель и веб-разработчик, автор книги *HTML5 и CSS3 для чайников*. Прежде чем основать компанию WatzThis?, Крис восемнадцать лет проработал исполнительным директором компании Minnick Web Services, где под его руководством и с его участием были разработаны сотни проектов, выполненных как по заказу небольших компаний, так и компаний с мировым именем. Крис обучил HTML, JavaScript, CSS и мобильной разработке не одну тысячу студентов.

Ева Холланд — опытный писатель, преподаватель и соучредитель компании WatzThis?. Она превосходно умеет излагать самые сложные концепции на языке, понятном для новичков с любым уровнем подготовки. Ева занималась писательской, дизайнерской и преподавательской деятельностью в онлайн-режиме, персонально и посредством видеокурсов. Ею созданы учебные программы по веб-дизайну, мобильной разработке и поисковой оптимизации (SEO). До учреждения компании WatzThis? Ева занимала должность операционного директора компании MWS, где зарекомендовала себя как умелый руководитель, способный видеть перспективу и обеспечивать достижение стоящих перед компанией задач.

Введение

JavaScript сейчас на пике популярности. Когда-то это был наскоро скроенный язык для одного из первых браузеров, теперь же он стал самым популярным в мире языком программирования. Спрос на JavaScript-программистов как никогда высокий и продолжает неуклонно расти.

Эта книга — ваш ключ к овладению основными концепциями JavaScript. Независимо от того, стремитесь ли вы оказаться на высокооплачиваемой должности программиста или хотите создать собственный интерактивный сайт, смеем вас заверить, что содержание книги и описанные в ней методики полностью соответствуют самым последним стандартам JavaScript и наилучшей практике программирования. Каждая глава содержит примеры реального кода, которые вы сможете протестировать в браузере в домашних условиях.

Точно так же, как для музыканта единственный способ оказаться на сцене Карнеги-Холла — это репетировать, репетировать и еще раз репетировать, единственный способ стать лучшим программистом — кодировать, кодировать и еще раз кодировать!

О чем эта книга

Эта книга представляет собой доступное руководство, предназначенное для тех, кто только учится писать код на JavaScript. По сравнению с другими языками программирования JavaScript отличается простотой, и вам будет нетрудно начать его применять. Ввиду легкости освоения этого языка многие из тех, кто начинал свою деятельность как веб-дизайнер, вскоре обнаруживали, что в состоянии самостоятельно изменять и писать JavaScript-код. Если вы относитесь к этой категории людей, то книга поможет вам быстро и без лишних сложностей войти в курс дела.

Независимо от того, знакомы ли вы с JavaScript-кодом или никогда в жизни его не видели, эта книга научит вас самостоятельному созданию корректного кода.

В частности, в книге рассматриваются следующие темы:

- ✓ базовая структура JavaScript-программ;
- ✓ интеграция JavaScript с HTML и CSS;
- ✓ структурирование программ за счет использования функций;
- ✓ работа с объектами JavaScript;
- ✓ использование передовых методик JavaScript, таких как AJAX, функции обратного вызова и замыкания;
- ✓ основы jQuery.

Освоение языка JavaScript не сводится к изучению одного только синтаксиса. Вы также должны знать, как получить доступ к инструментам разработчика и присоединиться к сообществу, успевшему сформироваться вокруг этого языка. За всю длительную и увлекательную историю развития JavaScript профессиональные программисты идеально отшлифовали инструментальные средства и методики, применяемые для создания программ на этом языке. В книге мы постоянно обращаем ваше внимание на наилучшие из существующих подходов и средств, предназначенных для тестирования и документирования кода и повышения эффективности всего процесса разработки в целом.

Сделаем несколько замечаний, которые облегчат вам чтение книги.

- ✓ Весь JavaScript-код, а также разметка HTML и CSS выделяются в тексте моноширинным шрифтом, например:

```
document.write("Привет!");
```

- ✓ Поскольку размеры книжных страниц не совпадают с размерами экранов компьютерных мониторов, длинные цепочки HTML-, CSS- и JavaScript-кода могут продолжаться на нескольких строках. Помните, что компьютер рассматривает подобный код так, будто это одна строка. Мы указываем на то, что код в действительности представляет собой одну строку, разрывая его на знаке препинания или пробеле и располагая остальную часть на следующей строке с отступом примерно так, как показано ниже.

```
document.getElementById("anElementInTheDocument").  
    addEventListener("click",doSomething,false);
```

- ✓ HTML и CSS не особо чувствительны к регистру букв, чего нельзя сказать о JavaScript, где это очень важно. Чтобы получать правильные результаты, используйте прописные и строчные буквы в строгом соответствии с текстом примеров.

Исходные предположения

Чтобы разбираться в программировании, вовсе не требуется быть программным экспертом или хакером. Доскональное знание работы компьютера для этого также не требуется, и вы даже не обязаны уметь считать в двоичной системе.

И все же нам придется сделать некоторые предположения относительно вас. Мы предполагаем, что вы умеете включать компьютер, знаете, как пользоваться мышью и клавиатурой, имеете подключение к Интернету и на вашем компьютере установлен браузер. Если вам уже известно кое-что о том, как создавать веб-страницы (это не так уж и сложно!), то считайте, что у вас есть небольшая фора.

Все остальные детали, которые потребуются вам для того, чтобы начать писать и выполнять программы на JavaScript, изложены в этой книге. А внимание к деталям, в чем вы сами убедитесь, — это непреложная истина программирования.

Пиктограммы, используемые в книге

Ниже приведены пиктограммы, используемые в книге для выделения фрагментов, на которые мы хотели бы особо обратить ваше внимание.



Эта пиктограмма обозначает полезные советы, подсказки и приемы, позволяющие экономить время и усилия.



Концентрируйте свое внимание всякий раз, когда вам встречается эта пиктограмма. Она говорит о том, что обозначенную ею информацию стоит запомнить.



Будьте внимательны, очень внимательны! Эта пиктограмма предупреждает о разного рода ловушках, которых следует избегать.



Эта пиктограмма обозначает информацию технического характера, которая может быть интересной для вас. Можете смело пропускать такую информацию, но если хотите узнать технические подробности, то чтение этих абзацев доставит вам удовольствие.

Дополнительные материалы

Изложенный в книге основной материал дополняют следующие информационные ресурсы.

- ✓ **Примеры.** Коды примеров, используемых в книге, можно загрузить по следующему адресу:

<http://www.codingjsfordummies.com/code/>

- ✓ **Упражнения.** Читателям, владеющим английским языком, будет полезно проделать упражнения, подготовленные для книги на сайте www.codecademy.com. Чтобы получить доступ к упражнениям, посетите следующий сайт:

<http://www.dummies.com/go/codingwithjavascript>

Что дальше

Написание программ на JavaScript доставляет большое удовольствие, и стоит вам овладеть хотя бы минимальным багажом знаний в этой области, как перед вами откроется удивительный мир интерактивных веб-приложений. Поэтому будьте смелее! Мы надеемся, что книга станет хорошим подспорьем в ваших начинаниях.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@dialektika.com

WWW: www.dialektika.com

Наши почтовые адреса:

в России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

в Украине: 03150, Киев, а/я 152

Часть I

Приступаем к программированию на JavaScript



© Иванов Владимир/cartoonbank.ru

В этой части...

- ✓ Написание первой программы на JavaScript
- ✓ Работа с переменными и массивами
- ✓ Операторы, выражения и инструкции
- ✓ Циклы и ветвление кода в программах на JavaScript

Глава 1

Самый неправильно понятый язык программирования в мире

В этой главе...

- Знакомство с JavaScript
- Возможности JavaScript
- Преимущества JavaScript

“Люди настолько плохо понимают меня, что даже мои переживания по поводу того, что меня не понимают, им непонятны”.

Серен Кьеркегор

Язык программирования JavaScript не всегда пользовался такой большой популярностью, как в наши дни. Одни считали его лучшим языком программирования в мире, другие — наихудшим. За последние несколько лет как в методику программирования на JavaScript, так и в интерпретаторы этого языка были внесены многочисленные усовершенствования. В результате современный язык JavaScript значительно улучшился по сравнению с ранними версиями.

В этой главе вы узнаете о том, что собой представляет JavaScript, и познакомитесь с краткой историей развития и становления этого языка программирования. Кроме того, вы получите общее представление о возможностях JavaScript и сферах его применения.

Что такое JavaScript?

На заре Интернета браузеры представляли собой простые программы для чтения веб-страниц (рис. 1.1). Они практически не располагали собственными функциональными возможностями, в лучшем случае позволяя лишь изменять размер шрифта отображаемого текста. Как только компания Microsoft выпустила браузер Internet Explorer, разгорелась так называемая *война браузеров* — жесткое конкурентное сражение за господство на рынке, резко ускорившее разработку новых средств. В браузерах одна за одной стали предлагаться такие возможности, как отображение графики, использование шрифтов различных видов, мерцающий текст, бегущая строка и многие другие.

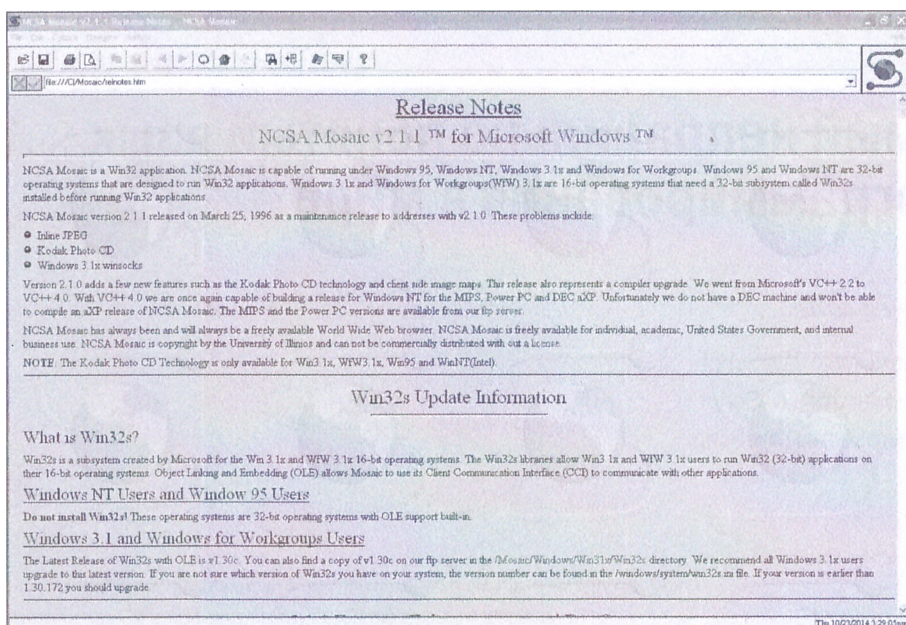


Рис. 1.1. Неприятный внешний вид веб-страницы в окне одного из первых браузеров

Прошло совсем немного времени, пока одного человека не осенила мысль, что браузер может не только быть программой для отображения текста в красивом виде, но и самостоятельно выполнять массу полезных вещей.

И тут пришел Эйх...

Проект JavaScript стартовал в компании Netscape в далеком 1995 году. Брендан Эйх, создатель языка JavaScript, написал его в рекордно короткие сроки (говорят, что для этого ему понадобилось всего лишь десять дней!), заимствуя все лучшее из ряда других языков программирования. Из-за спешки, вызванной стремлением как можно быстрее выйти на рынок с готовым продуктом, в язык вкрались некоторые, скажем так, “странности” (а если говорить без обиняков — ошибки). Результат представлял собой своеобразный язык наподобие эсперанто, который обманчиво воспринимался людьми, имеющими опыт работы с другими языками программирования, как язык, с которым они уже знакомы.

Сначала был Mocha

Язык JavaScript первоначально назывался Mocha. С выпуском первой бета-версии браузера Netscape Navigator он был переименован в LiveScript, но в версию браузера Netscape 2, выпущенную в 1995 г., он был встроен уже как язык JavaScript. Оперативно воспользовавшись методами обратного инжиниринга (обратной разработки), компания Microsoft встроила в свой браузер Internet Explorer точную копию JavaScript, назвав ее Jscript, чтобы избежать проблем, связанных с незаконным использованием торговой марки.

Компания Netscape представила язык JavaScript в организацию Ecma International, занимающуюся стандартизацией информационных и коммуникационных технологий, где он был утвержден в качестве стандарта ECMAScript в 1997 г.



Брендан Эйх, создатель JavaScript, отпустил по поводу названия стандартизированного языка нашумевшее замечание, в котором охарактеризовал ECMAScript как “неудачное название торговой марки, звучащее подобно названию кожного заболевания”.



Признавая ECMAScript неудачным названием для языка программирования, можно сказать то же самое и в отношении названия JavaScript, которое было дано языку компанией Netscape и используется большинством тех, кто с ним работает. Если вы уже знаете, как писать программы на языке Java, или планируете этому научиться, то должны отдавать себе отчет в том, что, несмотря на определенное сходство между ними, в действительности Java и JavaScript — это два совершенно разных языка.

Даешь больше эффектов!

Сразу же после своего дебюта язык JavaScript быстро приобрел популярность в качестве средства, позволяющего делать веб-страницы более динамичными. Одним из первых результатов встраивания JavaScript в браузеры стало появление так называемого динамического HTML (DHTML) — языка разметки веб-страниц, обеспечившего реализацию не только всевозможных забавных эффектов, таких как падающие снежинки (рис. 1.2), всплывающие окна и фигурные уголки страниц, но и более полезных вещей, например раскрывающихся меню и средств валидации (проверки корректности данных) веб-страниц.

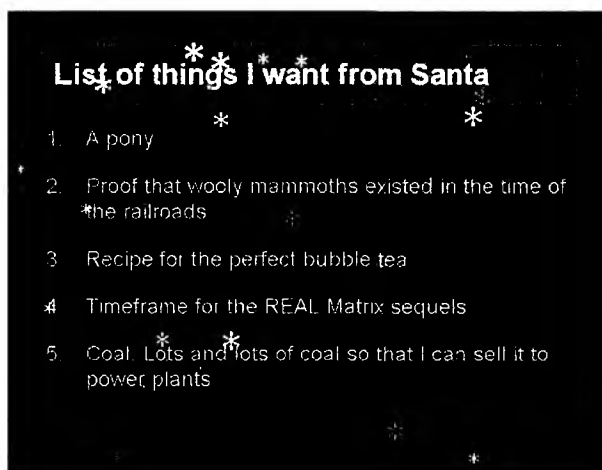


Рис. 1.2. Эффект падающих снежинок на веб-странице, созданный с помощью JavaScript

JavaScript взрослеет

Разменяв третий десяток лет, JavaScript стал наиболее широко используемым языком программирования в мире, и практически на каждом из эксплуатируемых в настоящее время персональных компьютеров установлен по крайней мере один браузер, способный выполнять JavaScript-код.

Язык JavaScript достаточно гибок для того, чтобы быть пригодным для использования непрограммистами, и одновременно располагает достаточно мощными возможностями, которых профессиональным программистам вполне хватает для того, чтобы поддерживать функциональность практически любого современного сайта в Интернете, начиная с одностраничных сайтов и кончая такими “монстрами”, как сайты Google, Amazon, Facebook и многие другие.

Динамический язык сценариев

JavaScript часто описывают как *динамический сценарный язык*, т.е. язык для написания сценариев (жарг. *скриптовый язык*). Чтобы смысл этого стал вам понятен, потребуется сначала определить пару терминов и разобраться с контекстом применения языка.

Распространенные заблуждения относительно JavaScript

За долгие годы существования JavaScript в его адрес было высказано немало острых замечаний. Некоторые из ходящих о нем слухов довольно интересны, однако не всегда справедливы. Ниже перечислены некоторые распространенные заблуждения, касающиеся JavaScript.

✓ **Миф:** JavaScript не является настоящим языком программирования. **Реальность:** JavaScript часто используют для выполнения простейших задач в браузерах, но это никоим образом не лишает его статуса полноценного языка программирования. В действительности многие передовые характеристики JavaScript установили высокую планку для языков программирования и теперь имитируются во многих языках, включая PHP, C++ и даже Java.

✓ **Миф:** JavaScript родствен Java. **Реальность:** вовсе нет. Название JavaScript было выбрано исключительно из маркетинговых соображений, поскольку на момент выхода JavaScript язык Java пользовался невероятной популярностью.

✓ **Миф:** JavaScript — новый язык. **Реальность:** некоторые из известных нам профессиональных программистов на JavaScript к моменту создания этого языка даже не успели появиться на свет.

✓ **Миф:** В движке JavaScript имеются ошибки, и в разных браузерах он выполняется по-разному. **Реальность:** хотя в некоторых случаях это действительно является правдой, производители браузеров давным-давно приняли решение поддерживать стандартизированную версию JavaScript. В наше время JavaScript выполняется одинаково во всех браузерах.

Компьютерные программы — это наборы инструкций, с помощью которых можно заставить компьютер выполнять определенные действия. Каждый компьютерный язык программирования располагает собственным набором инструкций и правилами их записи, которыми должны руководствоваться программисты. Компьютер неспособен понимать эти инструкции в непосредственном виде. Чтобы компьютер мог понимать

язык программирования, требуется предварительный процесс преобразования инструкций, в ходе которого они транслируются (переводятся) из формы, удобной для чтения (и записи) человеком, в машинный язык. В зависимости от того, когда именно осуществляется процесс трансляции инструкций, языки программирования делятся на два основных типа: компилируемые и интерпретируемые (рис. 1.3).

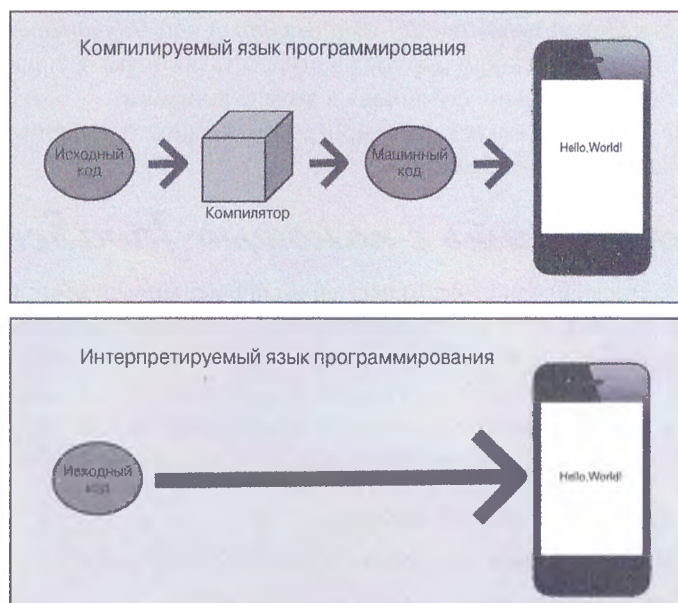


Рис. 1.3. Классификация языков программирования в зависимости от того, когда именно осуществляется компиляция программы

Компилируемые языки программирования

Компилируемые языки программирования — это языки, которые требуют, чтобы написанный программистом код был обработан специальной программой, так называемым *компилятором*, который анализирует данный код, а затем преобразует его в машинный язык. После этого компьютер может выполнить скомпилированную программу.

В качестве примера компилируемых языков программирования можно привести C, C++, Fortran, Java, Objective-C и COBOL.

Интерпретируемые языки программирования

Интерпретируемые языки программирования — это языки, которые также компилируются компьютером в машинный язык, но процесс компиляции осуществляется в браузере пользователя непосредственно во время выполнения программы. Программистам, которые пишут программы с использованием интерпретируемых языков, нет нужды компилировать свой код, прежде чем передавать его компьютеру для выполнения.

Программирование с использованием интерпретируемых языков обладает тем преимуществом, что изменения могут быть легко внесены в программу в любой момент.

Но у таких языков есть и недостаток, заключающийся в том, что компиляция кода во время его выполнения создает дополнительную стадию процесса, которая может замедлять выполнение программ.

Отчасти именно из-за такого снижения производительности интерпретируемые языки получили репутацию не очень серьезных языков программирования. Однако с появлением улучшенных компиляторов, осуществляющих компиляцию кода на стадии его выполнения (так называемые *JIT-компиляторы*¹ или *оперативные компиляторы*), и процессоров с повышенным быстродействием такая точка зрения очень быстро устарела. JavaScript значительно улучшился в этом отношении.

Примерами интерпретируемых языков программирования могут служить PHP, Perl, Haskell, Ruby и, конечно же, JavaScript.

Что можно делать с помощью JavaScript?

Если вы пользуетесь Интернетом, то регулярно применяете JavaScript. Список того, что позволяет делать JavaScript, довольно обширен и охватывает как простые напоминания, которые вы получаете всякий раз, когда забываете заполнить обязательное поле формы, так и сложные приложения, такие как Google Docs или Facebook. Вот лишь краткий перечень наиболее распространенных вариантов применения JavaScript:

- ✓ эффекты;
- ✓ проверка вводимых данных;
- ✓ трансформация изображения при наведении мыши;
- ✓ раскрывающиеся и выдвигающиеся меню;
- ✓ возможности перетаскивания и вставки;
- ✓ бесконечная прокрутка веб-страниц;
- ✓ автозаполнение;
- ✓ индикаторы выполнения;
- ✓ вкладки внутри веб-страниц;
- ✓ сортируемые списки;
- ✓ Magic Zoom (рис. 1.4).

Почему именно JavaScript?

Язык JavaScript стал стандартом для создания динамических пользовательских интерфейсов интернет-приложений. Всякий раз, когда вы посещаете веб-страницу с анимацией, автоматически обновляемыми данными, кнопкой, изменяющей свой вид при наведении на нее мыши, или раскрывающимся меню, в большинстве случаев вы имеете дело с результатами работы JavaScript. Благодаря тому, что программы, написанные на JavaScript, могут применяться для решения различных задач и способны

¹ От англ. “just-in-time” — непосредственно в момент возникновения необходимости. — *Примеч. ред.*

выполняться в любом браузере, программирование на JavaScript приобрело необычайную популярность, а владение этим языком стало обязательным требованием в современной веб-разработке.

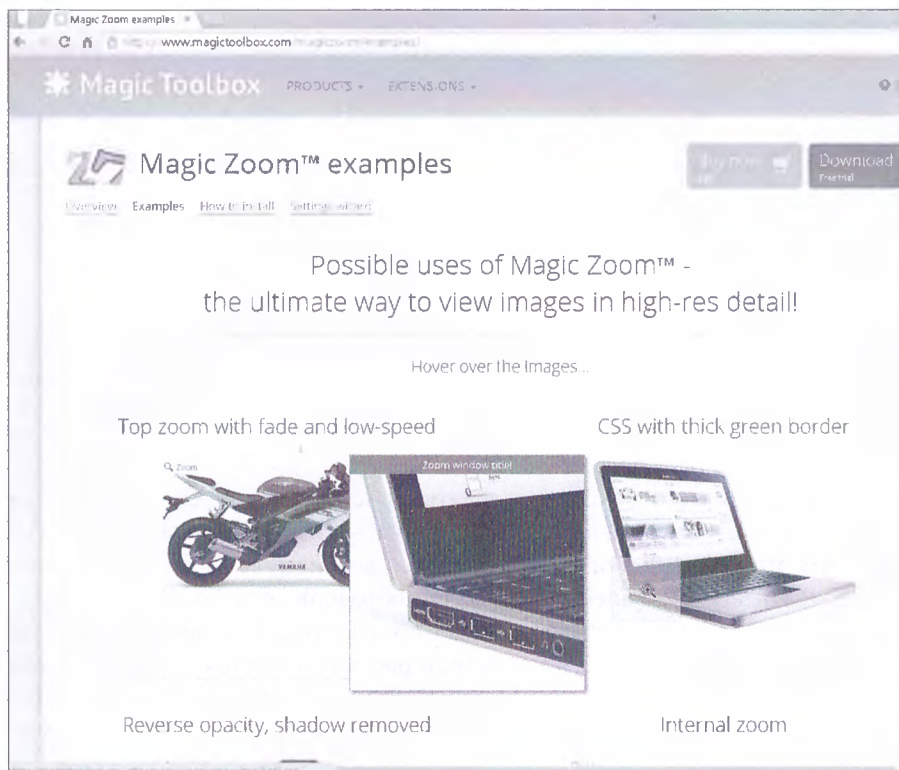


Рис. 1.4. Эффект Magic Zoom, активизированный с помощью JavaScript

Язык JavaScript легок в изучении

Языки программирования создавались для того, чтобы предоставить людям простой способ общения с компьютерами и возможность отдавания им приказов относительно того, что именно они должны делать. По сравнению с машинным языком, т.е. языком, на котором “говорит” центральный процессор (CPU) компьютера, любой язык программирования гораздо проще и понятнее. Чтобы вы могли получить представление о том, какого рода инструкциям в действительности подчиняется ваш компьютер, взгляните на приведенный ниже образец программы на машинном языке, которая выводит строку “Hello World”.

```
b8 21 0a 00 00
a3 0c 10 00 06
b8 6f 72 6c 64
a3 08 10 00 06
b8 6f 2c 20 57
a3 04 10 00 06
```

b8	48	65	6c	6c
a3	00	10	00	06
b9	00	10	00	06
ba	10	00	00	00
bb	01	00	00	00
b8	04	00	00	00
cd	80			
b8	01	00	00	00
cd	80			

А теперь посмотрите, как можно решить эту же простую задачу с помощью JavaScript:

```
alert("Hello World");
```

Намного проще, не правда ли?



Как только вы изучите основные правила дорожного движения (называемые *синтаксисом*), например, в каких ситуациях следует использовать круглые или фигурные скобки, JavaScript станет для вас столь же привычным, как обычный разговорный язык.



Первое, что необходимо сделать, приступая к изучению любого языка, в том числе и языка программирования, — это избавиться от страха неудачи. В этом отношении у вас не должно возникнуть сложностей с JavaScript. В Интернете существуют тысячи образцов кода JavaScript, любой из которых можно скачать и начать с ним работать. Все необходимые для этого инструменты у вас уже есть (они рассматриваются в главе 2), и вам остается лишь начать изучение JavaScript с небольших примеров, чтобы впоследствии постепенно дорабатывать их для решения полезных задач.

Где работает JavaScript? JavaScript работает везде!

Изначально язык JavaScript предназначался для использования в браузерах, однако он нашел применение во многих местах. В наши дни JavaScript используется в смартфонах и планшетах, в веб-серверах и в настольных приложениях, а также во всякого рода мобильных устройствах.

JavaScript в браузере

Чаще всего JavaScript можно обнаружить там, где его первоначально предполагалось использовать, — в веб-браузерах. В подобных случаях о JavaScript говорят как о *клиентском JavaScript* или *JavaScript на стороне клиента*.

Клиентский JavaScript позволяет сделать веб-страницы интерактивными. Это осуществляется следующими способами:

- ✓ путем управления самим браузером или использования его функциональности;
- ✓ манипулированием структурой и содержимым веб-страниц;