

NITE 国家信息技术紧缺人才培养工程指定教材



工业和信息化“十三五”人才培养规划教材

JavaScript前端开发 案例教程

黑马程序员 编著

有问题，就找黑马程序员问答精灵！



本书讲解了JavaScript、Ajax、jQuery的相关知识，并提供103个案例，以及一个综合项目。

提供免费教学资源，包括12个精美教学PPT、700道测试题、长达30小时的教学视频等。

添加QQ或微信号208695827，获取教学答案、源码，抢“助学金红包”。

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

NITE 国家信息技术紧缺人才培养工程指定教材

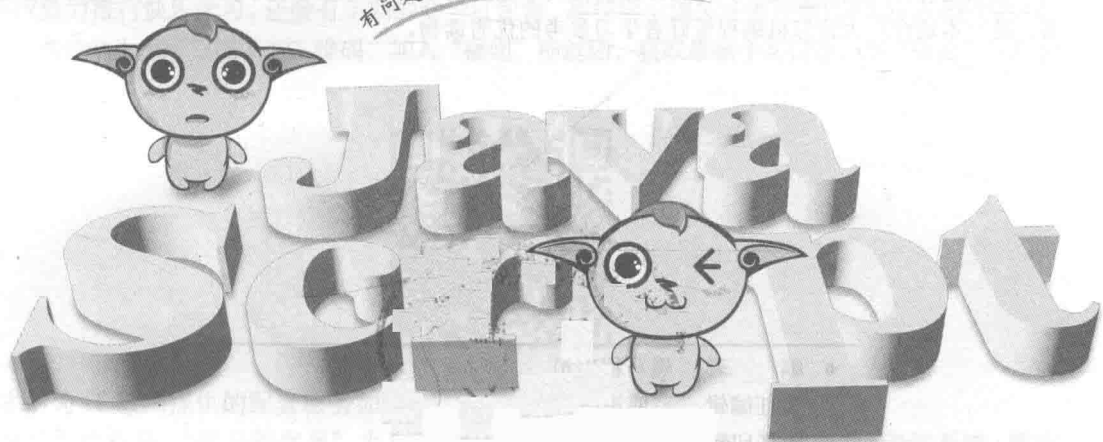


工业和信息化部“十三五”人才培养规划教材

JavaScript前端开发 案例教程

黑马程序员 编著

有问题，就找黑马程序员问答精灵！



人民邮电出版社

北京

图书在版编目(CIP)数据

JavaScript前端开发案例教程 / 黑马程序员编著

— 北京 : 人民邮电出版社, 2018.2

工业和信息化“十三五”人才培养规划教材

ISBN 978-7-115-44318-2

I. ①J… II. ①黑… III. ①JAVA语言—程序设计—
高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第321095号

内 容 提 要

JavaScript 是一种广泛应用于 Web 前端开发的脚本语言, 具有简单、易学、易用的特点, 用 JavaScript 开发网页可以增强网页的互动性, 为用户提供实时的、动态的交互体验。

本书共分为 12 章, 内容包括 JavaScript 快速入门、基本语法、数组、函数、对象、BOM、DOM、事件、正则表达式、Ajax、jQuery 和网页版 2048 小游戏。为了加深初学者对知识的领悟, 本书在确保知识讲解系统、全面的基础上, 还配备了精彩的案例, 将多个知识点综合运用。

本书适合作为高等院校本、专科计算机相关专业的教材使用, 也可作为 JavaScript 爱好者的参考书, 是一本适合广大计算机编程爱好者学习参考的优秀读物。

◆ 编 著 黑马程序员

责任编辑 范博涛

责任印制 马振武

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市祥达印刷包装有限公司印刷

◆ 开本: 787×1092 1/16

印张: 21.5

2018 年 2 月第 1 版

字数: 523 千字

2018 年 2 月河北第 1 次印刷

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

江苏传智播客教育科技股份有限公司(简称传智播客)是一家致力于培养高素质软件开发人才的科技公司,“黑马程序员”是传智播客旗下高端IT教育品牌。

“黑马程序员”的学员多为大学毕业后,想从事IT行业,但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格,包括了严格的技术测试、自学能力测试,还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员的质量,降低了企业的用人风险。

自“黑马程序员”成立以来,教学研发团队一直致力于打造精品课程资源,不断在产、学、研3个层面创新自己的执教理念与教学方针,并集中“黑马程序员”的优势力量,有针对性地出版了计算机系列教材50多册,制作教学视频数十套,发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发IT系列教材,还为高校师生提供相关的配套学习资源与服务。

本书为大学生提供的配套服务如下。

1. 专业的辅助学习平台“博学谷”(http://yuanxiao.boxuegu.com),专业老师在线为您解答。
2. 针对高校学生在学习过程中存在的压力等问题,我们还面向大学生量身打造了“播妞”。“播妞”不仅致力推行快乐学习,还会有定期的助学红包雨。高校学生请关注播妞微信/QQ:208695827。
3. 高校学生也可扫描下方二维码,加入“播妞”粉丝团,获取最新学习资源,与“播妞”一起快乐学习。



本书为IT教师提供的配套服务如下。

针对高校教学,“黑马程序员”为IT系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源,高校老师请关注码大牛老师微信/QQ:2011168841,获取配套资源,也可以扫描下方二维码,加入专为IT教师打造的师资服务平台——“教学好助手”,获取“黑马程序员”最新教师教学辅助资源的相关动态。



JavaScript 是一种脚本语言，从诞生至今广泛应用于 Web 开发，可以实现网页的交互，为用户提供流畅美观的浏览效果。近几年，互联网行业对用户体验的要求越来越高，前端开发技术越来越受到重视，JavaScript 作为 Web 前端开发领域中举足轻重的一门语言，如何能够快速、全面、系统地了解并掌握它的应用，成为 Web 开发人员的迫切需求。

为什么要学习本书

本书面向具有网页（HTML、CSS）基础的人群，讲解如何将 JavaScript 与 HTML、CSS 相结合，开发交互性强的网页。通过先易后难、从简入繁、从基础到高级的阶梯方式逐步深入讲解，将 JavaScript 基础知识和应用技术进行了综合讲解。

本书遵循基础知识先行、实用案例辅助、综合项目护航的原则，按照学习的难易度以及先后顺序，采用“知识讲解+案例实践”的混合方式来安排全书的内容，及时有效地引导初学者将学过的内容串联起来，培养分析问题和解决问题的综合运用能力。本书将抽象的概念具体化，学到的知识实践化，让读者不仅理解和掌握基本知识，还能根据实际需求进行扩展与提高，达到“学用结合”的效果。

如何使用本书

本书讲解的内容主要包括 JavaScript 基本语法、数组、函数、对象、BOM、DOM、事件等基础知识，以及正则表达式、Ajax、jQuery 等扩展知识，最后还提供了一个综合性的实战项目——“2048”网页游戏开发。全书共分为 12 章，分别对每章进行简要的介绍，具体如下。

第 1 章主要讲解 JavaScript 的基本概念，包括 JavaScript 的由来、特点、应用等。为了帮助初学者快速体验 JavaScript 编程，本章铺垫了一些基础知识，如编程实现数学运算、比较两个数字大小等。考虑到初学者在学习过程中经常会遇到函数、对象、事件等陌生名词，本章对这些基本的概念进行了通俗易懂的讲解，让读者理解这些技术出现的原因，建立学习的目标感和方向感。

第 2~4 章分别围绕 JavaScript 基本语法、数组和函数进行深入的讲解，这部分内容是 JavaScript 编程的基本功，只有准确理解和熟练掌握了这部分内容，才能够解决实际开发中的一些基本需求。为了避免学习过程枯燥乏味，在课程中安排了九九乘法表、省份城市的三级联动、网页计算器等趣味案例，对重点知识进行强化练习，使学习情况得到即时反馈。

第 5 章主要讲解对象相关的内容。近年来出现了许多流行的 JavaScript 框架和模板引擎，从侧面反映了动态网页显示逻辑的开发逐渐从后端转向前端，这就对前端代码的可维护性、可扩展性提出了更高的要求。为此，本章讲解了 JavaScript 自定义对象、内置对象、代码调试、原型与继承等内容，通过学习这些课程，可以帮助读者提高代码编写的质量，为模块化开发、框架开发、团队协作开发等场景奠定基础。

第 6~8 章分别讲解 BOM、DOM 和事件，这些技术与浏览器和网页密切相关。诸如限时秒杀、定时跳转、购物车等交互功能的开发都离不开这些技术。开发人员甚至可以将贪吃蛇、俄罗斯方块这些经典游戏搬到网页上，将网页转变成支持用户操作的界面。

第 9 章讲解正则表达式的基本语法,以及在 JavaScript 中的使用。正则表达式经常用于对一些有规律的字符串进行处理。例如,验证用户填写的表单、查找和替换文本、抓取内容等。在实际开发中,通过正则表达式可以用简短的几行代码,完成原本可能需要几十甚至上百行代码的工作,极大地方便了对字符串的处理。

第 10 章讲解 Ajax 技术。Ajax 是 Web 开发中非常重要的一项技术,它使得客户端网页可以与服务器端进行异步通信。利用 Ajax 技术,可以在网页上开发在线聊天、自动接收消息、无刷新分页等需要与服务器通信的程序,而不需要安装任何额外的浏览器插件,极大增强了用户体验。

第 11 章讲解 jQuery 的使用。jQuery 是 JavaScript 开发中的利器,它简化了 DOM 操作,用更少的代码完成更多的功能。即使对于 JavaScript 基础非常薄弱的新手,也能通过 jQuery 轻松完成许多功能的开发。读者在掌握了 JavaScript 面向对象等课程后再来学习本章节,可以更好地理解 jQuery 的实现原理,借鉴 jQuery 的设计思想来编写简洁、高质量的代码。

第 12 章讲解了实战项目 2048 网页游戏的开发。游戏编程不仅有趣味性,也具有一定的难度,非常考验开发人员的技术功底。本章不仅完成了游戏所有功能逻辑的开发,对于代码也进行了合理的封装,为后期维护、修改和扩展提供便利。

在上面所列举的 12 章中,第 1~8 章是基础课程,主要帮助初学者奠定扎实的基本功;第 9~11 章是对扩展技术的讲解,这些章节内容比较复杂,希望初学者多加思考,认真完成书中所讲的每个案例;第 12 章的实战项目综合运用了前面的知识与技术,在完全掌握后,读者还可以尝试开发贪吃蛇、俄罗斯方块等其他类型的游戏。

在学习过程中,读者一定要亲自实践本书中的案例代码。如果不能完全理解书中所讲知识,读者可以登录博学谷平台,通过平台中的教学视频进行深入学习。学习完一个知识点后,要及时在博学谷平台上进行测试,以巩固学习内容。

另外,如果读者在理解知识点的过程中遇到困难,建议不要纠结于某个地方,可以先往后学习。通常来讲,通过逐渐地学习,前面不懂和疑惑的知识也就能理解了。在学习编程的过程中,一定要多多动手实践,如果在实践的过程中遇到问题,建议多思考,理清思路,认真分析问题发生的原因,并在问题解决后总结出经验。

致谢

本书的编写和整理工作由传智播客教育科技股份有限公司完成,主要参与人员有吕春林、陈欢、韩冬、乔治铭、高美云、马丹、李东超、王金涛、刘晓强、韩振国等,全体人员在近一年的编写过程中付出了很多心血,在此表示衷心的感谢。

意见反馈

尽管我们付出了最大的努力,但书中难免会有不妥之处,欢迎各界专家和读者朋友们来函给予宝贵意见,我们将不胜感激。您在阅读本书时,如发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至: itcast_book@vip.sina.com

专属于老师及学生的在线教育平台
yuanxiao.boxuegu.com

让 IT 教学更简单

教师获取教材配套资源



添加微信/QQ

2011168841

让 IT 学习更有效

学生获取课后作业习题答案及配套源码

添加播妞微信/QQ

208695827

学习问答精灵: ask.boxuegu.com

更多学习视频: dvd.boxuegu.com



专属大学生的圈子

第 1 章 JavaScript 快速入门 1

1.1 初识 JavaScript.....	2
1.1.1 什么是 JavaScript.....	2
1.1.2 JavaScript 的由来.....	3
1.1.3 JavaScript 的特点.....	4
1.1.4 JavaScript 与 ECMAScript 的关系.....	4
1.2 开发工具.....	5
1.2.1 浏览器.....	5
1.2.2 代码编辑器.....	6
1.2.3 【案例】第一个 JavaScript 程序.....	7
1.3 JavaScript 入门.....	9
1.3.1 JavaScript 引入方式.....	9
1.3.2 常用输出语句.....	11
1.3.3 注释.....	12
1.3.4 数据与运算.....	12
1.3.5 函数.....	14
1.3.6 对象.....	15
1.3.7 事件.....	17
1.3.8 【案例】改变网页背景色.....	18
动手实践: 验证用户输入的密码.....	19
本章小结.....	20
课后练习.....	20

第 2 章 基本语法 21

2.1 变量.....	22
2.1.1 标识符.....	22
2.1.2 关键字与保留字.....	22
2.1.3 变量的使用.....	23
2.1.4 变量的赋值.....	23
2.2 数据类型.....	24
2.2.1 数据类型分类.....	24
2.2.2 基本数据类型.....	24

2.2.3 数据类型检测.....	26	3.6.1 栈方法.....	60
2.2.4 数据类型转换.....	27	3.6.2 检索方法.....	61
2.3 表达式.....	29	3.6.3 数组转字符串.....	63
2.4 运算符.....	29	3.6.4 其他方法.....	63
2.4.1 算术运算符.....	29	3.6.5 【案例】猴子选大王.....	65
2.4.2 字符串运算符.....	30	动手实践：省份城市的三级联动.....	66
2.4.3 赋值运算符.....	30	本章小结.....	69
2.4.4 比较运算符.....	31	课后练习.....	70
2.4.5 逻辑运算符.....	32		
2.4.6 三元运算符.....	32	第4章 函数..... 71	
2.4.7 位运算符.....	33	4.1 函数的定义与调用..... 72	
2.4.8 运算符优先级.....	35	4.1.1 初识函数..... 72	
2.4.9 【案例】计算圆的周长和面积.....	35	4.1.2 参数设置..... 73	
2.5 流程控制..... 36		4.1.3 函数的调用..... 74	
2.5.1 选择结构..... 36		4.1.4 【案例】字符串大小写转换..... 75	
2.5.2 循环结构..... 39		4.2 变量的作用域..... 76	
2.5.3 跳转语句..... 42		4.3 匿名函数..... 78	
2.5.4 【案例】打印金字塔..... 43		4.3.1 函数表达式..... 78	
动手实践：九九乘法表..... 45		4.3.2 匿名函数..... 78	
本章小结..... 46		4.3.3 回调函数..... 79	
课后练习..... 46		4.4 嵌套与递归..... 81	
		4.4.1 函数嵌套与作用域链..... 81	
第3章 数组..... 48		4.4.2 递归调用..... 82	
3.1 初识数组..... 49		4.4.3 【案例】求斐波那契数列第N项 的值..... 83	
3.2 创建数组..... 49		4.5 闭包函数..... 83	
3.3 数组的基本操作..... 50		4.5.1 什么是闭包函数..... 83	
3.3.1 获取数组长度..... 50		4.5.2 闭包函数的实现..... 84	
3.3.2 数组的访问与遍历..... 51		动手实践：网页计算器..... 85	
3.3.3 元素的添加与修改..... 52		本章小结..... 86	
3.3.4 元素的删除..... 53		课后练习..... 86	
3.3.5 【案例】查找最大值与最小值..... 54			
3.4 常见二维数组操作..... 55		第5章 对象..... 88	
3.4.1 创建与遍历..... 55		5.1 面向对象概述..... 89	
3.4.2 【案例】二维数组转置..... 56		5.1.1 面向过程与面向对象..... 89	
3.5 数组排序..... 57		5.1.2 面向对象的特征..... 89	
3.5.1 冒泡排序..... 57		5.2 自定义对象..... 91	
3.5.2 插入排序..... 58			
3.6 常见数组方法..... 60			

5.2.1	对象的定义	91
5.2.2	访问对象成员	92
5.2.3	对象成员遍历	92
5.2.4	深拷贝与浅拷贝	93
5.3	构造函数	94
5.3.1	为什么使用构造函数	94
5.3.2	JavaScript 内置的构造函数	95
5.3.3	自定义构造函数	95
5.3.4	私有成员	96
5.3.5	函数中的 this 指向	97
5.4	内置对象	98
5.4.1	String 对象	98
5.4.2	Number 对象	100
5.4.3	Math 对象	100
5.4.4	Date 对象	101
5.4.5	【案例】制作年历	102
5.5	错误处理与代码调试	105
5.5.1	错误处理	105
5.5.2	错误对象	106
5.5.3	错误类型	107
5.5.4	代码调试	107
5.6	原型与继承	109
5.6.1	原型	109
5.6.2	继承	110
5.6.3	静态成员	113
5.6.4	属性搜索原则	113
5.6.5	原型链	114
	动手实践：表单生成器	118
	本章小结	123
	课后练习	123

第 6 章 BOM 125

6.1	什么是 BOM 对象	126
6.2	window 对象	126
6.2.1	全局作用域	126
6.2.2	弹出对话框和窗口	127
6.2.3	窗口位置和大小	130
6.2.4	框架操作	132

6.2.5	定时器	132
6.2.6	【案例】限时秒杀	134
6.3	location 对象	136
6.3.1	更改 URL	136
6.3.2	获取 URL 参数	137
6.3.3	【案例】定时跳转	138
6.4	history 对象	139
6.4.1	历史记录跳转	139
6.4.2	无刷新更改 URL 地址	140
6.5	navigator 对象	141
6.6	screen 对象	142
	动手实践：红绿灯倒计时	143
	本章小结	145
	课后练习	145

第 7 章 DOM 147

7.1	DOM 对象简介	148
7.1.1	什么是 DOM	148
7.1.2	DOM HTML 节点树	148
7.1.3	DOM 对象的继承关系	149
7.2	HTML 元素操作	150
7.2.1	获取操作的元素	150
7.2.2	元素内容	154
7.2.3	【案例】改变盒子大小	155
7.2.4	元素属性	156
7.2.5	元素样式	158
7.2.6	【案例】标签栏切换效果	161
7.3	文档节点操作	162
7.3.1	获取节点	162
7.3.2	节点追加	164
7.3.3	节点删除	165
7.3.4	【案例】列表的增删和移动	166
	动手实践：购物车	171
	本章小结	176
	课后练习	176

第 8 章 事件 178

8.1	事件处理	179
-----	------------	-----

8.1.1 事件概述.....	179	第 10 章 Ajax.....	224
8.1.2 事件的绑定方式.....	180	10.1 Web 基础知识.....	225
8.2 事件对象.....	182	10.1.1 Web 服务器.....	225
8.2.1 获取事件对象.....	182	10.1.2 HTTP.....	225
8.2.2 常用属性和方法.....	183	10.2 Web 服务器搭建.....	228
8.2.3 【案例】缓动的小球.....	186	10.2.1 PHP 开发环境.....	229
8.3 事件分类.....	188	10.2.2 前后端交互.....	230
8.3.1 页面事件.....	188	10.3 Ajax 入门.....	232
8.3.2 焦点事件.....	189	10.3.1 什么是 Ajax.....	232
8.3.3 鼠标事件.....	191	10.3.2 创建 Ajax 对象.....	233
8.3.4 【案例】鼠标拖曳特效.....	193	10.3.3 Ajax 向服务器发送请求.....	233
8.3.5 键盘事件.....	195	10.3.4 处理服务器返回的信息.....	236
8.3.6 表单事件.....	197	10.4 数据交换格式.....	238
动手实践：图片放大特效.....	198	10.4.1 XML 数据格式.....	239
本章小结.....	202	10.4.2 JSON 数据格式.....	240
课后练习.....	202	10.4.3 【案例】无刷新分页.....	242
第 9 章 正则表达式.....	204	10.5 跨域请求.....	250
9.1 认识正则表达式.....	205	10.5.1 Ajax 跨域问题.....	250
9.1.1 什么是正则表达式.....	205	10.5.2 JSONP 实现跨域请求.....	251
9.1.2 如何使用正则.....	205	10.6 扩展技术.....	254
9.1.3 获取正则对象.....	206	10.6.1 Cookie.....	254
9.2 字符类别与集合.....	207	10.6.2 FormData.....	255
9.2.1 字符类别.....	207	10.6.3 Promise.....	256
9.2.2 字符集合.....	208	10.6.4 WebSocket.....	260
9.2.3 【案例】限定输入内容.....	208	动手实践：进度条文件上传.....	262
9.3 字符限定与分组.....	211	本章小结.....	265
9.3.1 字符限定.....	211	课后练习.....	266
9.3.2 括号字符.....	212	第 11 章 jQuery.....	267
9.3.3 正则运算符优先级.....	214	11.1 jQuery 快速入门.....	268
9.3.4 【案例】内容查找与替换.....	214	11.1.1 什么是 jQuery.....	268
9.4 与正则相关的方法.....	215	11.1.2 下载 jQuery.....	268
9.4.1 RegExp 类中的方法.....	216	11.1.3 使用 jQuery.....	270
9.4.2 String 类中的方法.....	217	11.2 元素操作.....	271
动手实践：表单验证.....	217	11.2.1 jQuery 对象.....	271
本章小结.....	222	11.2.2 jQuery 选择器.....	271
课后练习.....	223	11.2.3 元素遍历.....	278

11.2.4	元素内容	279
11.2.5	元素样式	280
11.2.6	元素筛选	281
11.2.7	元素属性	283
11.2.8	【案例】折叠菜单	283
11.3	文档节点操作	285
11.3.1	节点追加	285
11.3.2	节点替换	286
11.3.3	节点删除	286
11.3.4	节点复制	287
11.3.5	【案例】左移与右移	287
11.4	事件操作	289
11.4.1	常用事件	289
11.4.2	页面加载事件	292
11.4.3	事件绑定与切换	292
11.4.4	【案例】手风琴效果	294
11.5	动画特效	295
11.5.1	常用动画	295
11.5.2	自定义动画	297
11.5.3	【案例】无缝轮播图	299
11.6	jQuery 操作 Ajax	302

11.7	插件机制	307
11.7.1	自定义插件	307
11.7.2	jQuery 插件库	308
11.7.3	jQuery UI	309
11.7.4	【案例】自定义全选与反选 插件	311
	本章小结	314
	课后练习	314

第 12 章 网页版 2048 小游戏 ... 315

12.1	游戏功能展示	316
12.2	实现步骤分析	317
12.3	游戏功能实现	317
12.3.1	设计游戏界面	317
12.3.2	控制游戏数值	319
12.3.3	实现单元格移动	323
12.3.4	设置游戏分数	326
12.3.5	判断胜利和失败	327
12.3.6	重新开始游戏	330
	本章小结	330
	课后练习	331

Chapter 1

第 1 章

JavaScript 快速入门

JavaScript

学习目标

- 熟悉 JavaScript 的用途和发展状况
- 理解 JavaScript 与 ECMAScript 的关系
- 掌握 JavaScript 的基本使用方法



在 Web 前端开发中, HTML、CSS 和 JavaScript 是开发网页所必备的技术。在掌握了 HTML 和 CSS 技术之后, 已经能够编写出各式各样的网页了, 但若想让网页具有良好的交互性, JavaScript 是一个极佳的选择。本章将介绍 JavaScript 的基本概念, 并通过实践案例来体验 JavaScript 编程。

1.1 初识 JavaScript

1.1.1 什么是 JavaScript

JavaScript 是 Web 开发领域中的一种功能强大的编程语言, 主要用于开发交互式的 Web 页面。在计算机、手机等设备上浏览的网页, 其大多数的交互逻辑几乎都是由 JavaScript 实现的。

对于制作一个网页而言, HTML、CSS 和 JavaScript 分别代表了结构、样式和行为, 结构是网页的骨架, 样式是网页的外观, 行为是网页的交互逻辑, 如表 1-1 所示。

表 1-1 比较 HTML、CSS 和 JavaScript

语言	作用	说明
HTML	结构	从语义的角度, 描述页面结构
CSS	样式	从审美的角度, 美化页面
JavaScript	行为	从交互的角度, 提升用户体验

JavaScript 内嵌于 HTML 网页中, 通过浏览器内置的 JavaScript 引擎直接编译, 把一个原本只用来显示的页面, 转变成支持用户交互的页面程序。

下面通过一些示例来展示 JavaScript 能够制作的页面效果, 如图 1-1 所示。



图 1-1 JavaScript 在网页中的应用

从图 1-1 中可以看出, 网页中许多常见的交互效果, 都可以利用 JavaScript 来实现。JavaScript 可以使网页的互动性更强, 用户体验更好。

1.1.2 JavaScript 的由来

在 1995 年时, Netscape (网景) 公司 (现在的 Mozilla) 的 Brendan Eich (布兰登·艾奇) 在网景导航者浏览器上首次设计出了 JavaScript。Netscape 最初将这个脚本语言命名为 LiveScript, 后来 Netscape 公司与 Sun 公司 (2009 年被 Oracle 公司收购) 合作之后将其改名为 JavaScript, 这是由于当时 Sun 公司推出的 Java 语言备受关注, Netscape 公司为了营销借用了 Java 这个名称, 但实际上 JavaScript 与 Java 的关系就像“大熊猫”与“小熊猫”, 它们本质上是两种不同的编程语言。

在设计之初, JavaScript 是一种可以嵌入到网页中的编程语言, 用来控制浏览器的行为。例如, 直接在浏览器中进行表单验证, 用户只有填写格式正确的内容后才能够提交表单, 避免了因表单填写错误导致的反复提交, 节省了时间和网络资源。

JavaScript 语言非常灵活, 其语言特性也产生了一些不良的影响。例如, 一些网站利用 JavaScript 制作网页上的漂浮广告、弹窗, 让用户感到厌烦。甚至还有一些不怀好意的人, 利用 Web 开发中的安全漏洞, 在网页中编写恶意代码, 窃取用户网站身份信息、传播病毒等。

尽管如此, JavaScript 仍然是 Web 开发中的一个不可或缺的技术。能否合理使用 JavaScript 取决于网站端正的态度和开发人员扎实的技术功底。人们更希望看到 JavaScript 推动 Web 技术的发展, 造福每一位互联网用户。

今天的 JavaScript 承担了更多的责任, 尤其是当 Ajax 技术兴起之后, 浏览器和服务器可以进行异步交互了, 网站的用户体验又得到了更大的提升。例如, 当用户在百度的搜索框中输入几个字以后, 网页会智能感知用户接下来要搜索的内容, 如图 1-2 所示。这个效果的实现, 离不开 JavaScript 编程。



图1-2 百度搜索框

另外, JavaScript 的用途已经不仅局限于浏览器了, Node.js 的出现使得开发人员能够在服务器端编写 JavaScript 代码, 使得 JavaScript 的应用更加广泛, 而本书主要针对浏览器端的 JavaScript 语言基础进行讲解, 推荐读者在掌握语言基础后再学习更高级的技术。

1.1.3 JavaScript 的特点

1. JavaScript 是脚本语言

脚本 (Script) 简单地说就是一条条的文本命令, 按照程序流程执行。常见的脚本语言有 JavaScript、VBScript、Perl、PHP、Python 等, 而 C、C++、Java、C# 这些语言不属于脚本语言。它们的区别在于, 非脚本语言一般需要编译、链接, 生成独立的可执行文件后才能运行; 而脚本语言依赖于解释器, 只在被调用时自动进行解释或编译。脚本语言缩短了传统语言“编写 → 编译 → 链接 → 运行”的过程。

脚本语言通常都有简单、易学、易用的特点, 语法规则比较松散, 使开发人员能够快速完成程序的编写工作, 但其缺点是执行效率不如编译型的语言快。不过, 由于计算机的运行速度越来越快, Web 应用的需求变化也越来越快, 人们更加重视软件的开发速度, 脚本语言带来的执行效率下降已经可以忽视了。

2. JavaScript 可以跨平台

JavaScript 语言不依赖操作系统, 仅需要浏览器的支持。目前, 几乎所有的浏览器都支持 JavaScript。在移动互联网时代, 利用手机等各类移动设备上网的用户越来越多, JavaScript 的跨平台性使其在移动端也承担着重要的职责。例如, JavaScript 可以搭配 CSS3 编写响应式的网页, 或者将网页模仿成移动 APP 的交互方式, 使 APP 开发和更新的周期变短。JavaScript 还可以搭配 HTML5 中的 Canvas (画布) 技术在网页上进行动画和游戏制作。

3. JavaScript 支持面向对象

面向对象是软件开发中的一种重要的编程思想, 其优点非常多。例如, 基于面向对象思想诞生了许多优秀的库和框架, 可以使 JavaScript 开发变得快捷和高效, 降低了开发成本。近几年, Web 前端开发技术日益受到重视, 除了经典的 JavaScript 库 jQuery, 又诞生了 Bootstrap、AngularJS、Vue.js、Backbone.js、React、webpack 等框架和工具。

1.1.4 JavaScript 与 ECMAScript 的关系

1996 年, 网景公司在 Navigator 2.0 浏览器中正式内置了 JavaScript 脚本语言后, 微软公司开发了一种与 JavaScript 相近的语言 JScript, 内置于 Internet Explorer 3.0 浏览器发布。网景公司面临丧失浏览器脚本语言的主导权的局面, 决定将 JavaScript 提交 Ecma 国际, 希望 JavaScript 能够成为国际标准。

Ecma 国际 (前身为欧洲计算机制造商协会) 是一家国际性会员制度的信息和电信标准组织, 该组织发布了 262 号标准文件 (ECMA-262), 规定了浏览器脚本语言的标准, 并将这种语言称为 ECMAScript。JavaScript 和 JScript 可以理解为 ECMAScript 的实现和扩展。

2015 年, Ecma 国际发布了新版本 ECMAScript 2015 (人们习惯称为 ECMAScript 6、ES6), 相比前一个版本做出了大量的改进。考虑到仍然有很多用户还在使用旧版本的浏览器, 为了保证网页的兼容性, 不建议开发人员使用这些新特性。但为了顺应技术更新, 本书在讲解时也会为大家补充介绍一些关于 ES6 的新技术。

1.2 开发工具

JavaScript 的开发工具主要包括浏览器和代码编辑器两种软件。浏览器用于执行、调试 JavaScript 代码，代码编辑器用于编写代码。本节将针对这两种开发工具进行讲解。

1.2.1 浏览器

浏览器是访问互联网中各种网站所必备的工具。由于浏览器的种类、版本比较多，作为 JavaScript 开发人员需要解决各种浏览器的兼容性，确保用户使用的浏览器能够准确执行自己编写的程序。表 1-2 列举了几种常见的浏览器及其特点。

表 1-2 常见浏览器

开发商	浏览器	特点
Microsoft	Internet Explorer	Windows 操作系统的内置浏览器，用户数量较多
	Microsoft Edge	Windows 10 操作系统提供的浏览器，速度更快、功能更多
Google	Google Chrome	目前市场占有率较高的浏览器，具有简洁、快速的特点
Mozilla	Mozilla Firefox	一款优秀的浏览器，但市场占有率低于 Google Chrome
Apple	Safari	主要应用在苹果 iOS、macOS 操作系统中的浏览器

在表 1-2 列举的浏览器中，Internet Explorer 浏览器的常见版本有 6、7、8、9、10、11。其中 6、7、8 发布时间较早，用户数量多，但兼容性和执行效率稍微低一些。本书选择各方面比较优秀的 Google Chrome 浏览器进行讲解。

面对市面上众多的浏览器，开发人员如何掌控程序的兼容性呢？实际上，许多浏览器都使用了相同的内核，了解其内核就能对浏览器有一个清晰的归类。浏览器内核分成两部分：排版引擎和 JavaScript 引擎。排版引擎负责将取得的网页内容（如 HTML、CSS 等）进行解析和处理，然后显示到屏幕中。JavaScript 引擎用于解析 JavaScript 语言，通过执行代码来实现网页的交互效果。下面分别介绍一些常见的排版引擎和 JavaScript 引擎。

1. 排版引擎

(1) Trident

Trident 是 Internet Explorer (IE) 浏览器使用的引擎。Trident 在 Windows 操作系统中被设计为一个功能模块，使得其他软件的开发人员可以便捷地将网页浏览功能加入到其开发的应用程序里。

国内很多的双核浏览器提供了“兼容模式”，该模式便是使用了 Trident 引擎。其代表软件有遨游、世界之窗、QQ 浏览器、猎豹安全浏览器、360 安全浏览器、360 极速浏览器等。

(2) EdgeHTML

微软公司在 Windows 10 操作系统中提供了一个新的浏览器 Microsoft Edge，其最显著的特点是使用了新引擎 EdgeHTML。EdgeHTML 在速度方面有了极大的提升，在 Trident 基础上删除了过时的旧技术支持代码，增加了许多对现代浏览器的技术支持。

(3) Gecko

Gecko 是 Mozilla FireFox (火狐浏览器) 使用的引擎，其特点是源代码完全公开，可开发程

度很高，全世界的程序员都可以为其编写代码、增加功能。Gecko 原本是由网景公司开发的，现在由 Mozilla 基金会维护。Gecko 是跨平台的，支持在 Windows、Linux 和 macOS 等操作系统上运行。

(4) WebKit

WebKit 是一个开放源代码的浏览器引擎，其所包含的 WebCore 排版引擎和 JavaScriptCore 引擎来自于 KDE 项目组的 KHTML 和 KJS。苹果公司采用了 KHTML 作为开发 Safari 浏览器的引擎后，衍生出了 WebKit 引擎，并按照开源协议开放了 WebKit 的源代码。WebKit 具有高效稳定、兼容性好、源码结构清晰、易于维护的特点。Google Chrome 浏览器也曾经使用过 WebKit 引擎。

(5) Blink

Blink 是一个由 Google 公司和 Opera Software ASA 开发的浏览器排版引擎，Google 公司将这个引擎作为开源浏览器 Chromium 项目的一部分。Blink 是 WebKit 中 WebCore 组件的一个分支，并且在 Chrome (28 及后续版本)、Opera (15 及后续版本) 等浏览器中使用。

目前国内大部分浏览器都采用了 WebKit 或 Blink 内核，一些双核浏览器将其作为“急速模式”的内核。在移动设备中，iPhone 和 iPad 等苹果 iOS 平台使用 Webkit 内核；Android 4.4 之前的 Android 系统浏览器内核是 Webkit，在 Android 4.4 系统中更改为 Blink。

2. JavaScript 引擎

(1) Chakra

Chakra 是微软公司在 IE 9~11、Microsoft Edge 等浏览器中使用的 JavaScript 引擎。目前，该引擎的核心部分已经开源，其开源版本称为 ChakraCore。

(2) SpiderMonkey

SpiderMonkey 是 Mozilla 项目中的一个 JavaScript 引擎，主要用于 Firefox 浏览器。其后又发布了 TraceMonkey、JaegerMonkey、IonMonkey、OdinMonkey 等改进版引擎，提高了性能。

(3) Rhino

Rhino 是 Mozilla 项目中的一个使用 Java 语言编写的 JavaScript 引擎，它被作为 Java SE6 上的默认 JavaScript 引擎，主要用于为 Java 执行环境提供 JavaScript 的支持。

(4) JavaScriptCore

JavaScriptCore 是苹果在 Safari 浏览器使用的 JavaScript 引擎。

(5) V8

V8 是 Google 为 Chrome 浏览器开发的 JavaScript 引擎，具有较快的执行速度。V8 非常受欢迎，其他一些软件也整合了 V8 引擎，如 Node.js。

1.2.2 代码编辑器

工欲善其事，必先利其器，一款优秀的开发工具能够极大提高程序开发效率与体验。在 Web 前端开发中，常用的编辑工具有 Notepad++、Sublime Text、WebStorm 等，下面介绍这些编辑工具的特点。

1. Notepad++

Notepad++ 是一款在 Windows 环境下免费开源的代码编辑器，支持的语言包括 HTML、CSS、JavaScript、XML、PHP、Java、C/C++、C# 等。