

二维码扫一扫，随时随地看视频

每个章节都有二维码，微信扫一扫，可随时随地看视频。

从配套到拓展，资源库一应俱全

除配套视频和素材源文件外，各类学习资源库一应俱全。

案例丰富详尽，边做边学更快捷

跟着大量案例去学习，边学边做，从做中学，可以更深入、更高效。

遵循学习规律，入门实战相结合

编写模式采用基础知识+中小实例+实战案例，循序渐进，结合紧密实战。

提供在线服务，随时随地可交流

提供微信公众号、QQ群等多渠道互动服务。

JavaScript

从入门到精通 (标准版)

512 节配套微视频讲解 **1016** 个实例案例分析

☑习题库 ☑面试题库 ☑案例库 ☑工具库 ☑网页模板库 ☑网页配色库 ☑网页素材库 等

未来科技◎编著



中国水利水电出版社
www.waterpub.com.cn

Web 开发视频点播大系

JavaScript 从入门到精通

(标准版)

未来科技 编著



中国水利水电出版社
www.waterpub.com.cn

• 北京 •

内 容 提 要

《JavaScript 从入门到精通（标准版）》系统地讲解了 JavaScript 语言的使用，并结合 HTML5 介绍了如何开发更富可用性的 Web 程序。本书分为 5 大部分，共 23 章。第 1 部分介绍 JavaScript 的基本概念和基础知识，以及如何快速上手测试 JavaScript 代码；第 2 部分介绍 JavaScript 核心部分——编程，包括变量、数据类型、表达式、运算符、语句、函数、数组、对象、正则表达式、函数式编程和面向对象编程；第 3 部分介绍了 JavaScript 客户端开发的相关知识和技术；第 4 部分讲解 JavaScript+HTML5 的应用，如本地存储、图形绘制、定位、多线程、离线应用、文件操作等；第 5 部分通过多个综合实例演示了如何使用 JavaScript 进行实战开发的过程。

《JavaScript 从入门到精通（标准版）》配备了极为丰富的学习资源，其中配套资源有：512 节教学视频（可二维码扫描）、素材源程序；附赠的拓展学习资源有：习题及面试题库、案例库、工具库、网页模板库、网页配色库、网页素材库、网页案例欣赏库等。

《JavaScript 从入门到精通（标准版）》适合作为 JavaScript 入门、JavaScript 实战、JavaScript 高级程序设计、HTML5 移动开发方面的自学用书，也可作为高等院校网页设计、网页制作、网站建设、Web 前端开发等专业的教学参考书或相关机构的培训教材。

图书在版编目（CIP）数据

JavaScript 从入门到精通：标准版 / 未来科技编著

— 北京：中国水利水电出版社，2017.8（2017.8 重印）

（Web 开发视频点播大系）

ISBN 978-7-5170-5414-6

I. ①J… II. ①未… III. ①JAVA 语言—程序设计

IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2017）第 115048 号

书 名	JavaScript 从入门到精通（标准版）JavaScript CONG RUMEN DAO JINGTONG （BIAOZHUN BAN）
作 者	未来科技 编著
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座 100038） 网址：www.waterpub.com.cn E-mail：sales@waterpub.com.cn 电话：（010）68367658（营销中心）
经 售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京智博尚书文化传媒有限公司
印 刷	三河市龙大印装有限公司
规 格	203mm×260mm 16 开本 49.75 印张 1400 千字
版 次	2017 年 8 月第 1 版 2017 年 8 月第 2 次印刷
印 数	5001—8000 册
定 价	89.80 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

前言

Preface

随着网络技术的不断进步,以及 HTML5 应用的不断拓展,其核心技术 JavaScript 越来越受到人们的关注。各种针对 JavaScript 的框架层出不穷,jQuery 就是这些框架中优秀的代表,它掀起了互联网技术的新一轮革命。

JavaScript 语言比较灵活、轻巧,兼顾函数式编程和面向对象编程的特性,备受 Web 开发人员的欢迎。本书将系统讲解 JavaScript 语言的使用,并结合 HTML5 介绍如何开发更富可用性的 Web 程序。

本书内容

本书分为 5 大部分,共 23 章,具体结构划分及内容如下。

第 1 部分:JavaScript 概述,包括第 1~3 章,主要介绍了 JavaScript 基础知识,以及如何快速上手测试 JavaScript 代码。

第 2 部分:JavaScript 核心编程,包括第 4~9 章,主要介绍 JavaScript 的核心部分——编程,包括变量、数据类型、表达式、运算符、语句、函数、数组、对象、正则表达式、函数式编程和面向对象编程。

第 3 部分:JavaScript DOM,包括第 10~16 章,主要介绍 JavaScript 客户端开发的相关知识和技术,如浏览器脚本化、网页脚本化、样式表脚本化以及事件处理。

第 4 部分:JavaScript + HTML5 应用,包括第 17~22 章,主要介绍 HTML5 应用新技术,如本地存储、图形绘制、定位、多线程、离线应用、文件操作等。

第 5 部分:综合实例,包括第 23 章,通过多个综合实例演示了使用 JavaScript 进行实战开发的过程。

本书编写特点

内容全面

本书不仅全面介绍 JavaScript 语言的基础知识,还系统讲解 JavaScript 客户端的开发,以及与 HTML5 相结合进行页面开发,并关注 JavaScript 拓展技术及其应用。

语言简练

本书语言通俗、简练,读起来不累、不绕。对于重难点技术和知识点,力求简洁明了,避免专业式说明,或者钻牛角尖。这对于初学者学习技术、理解和铭记一些重难点概念和知识是非常必要的。

循序渐进

本书以初、中级程序员为对象,先从 JavaScript 基础讲起,然后讲解 JavaScript 的核心技术,最后讲解 JavaScript 的高级应用。讲解过程中步骤详尽、内容新颖。

讲解贴心

书中每一章节均提供声图并茂的语音视频教学录像,读者可以根据书中提供的视频位置,在资源包中找到。这些视频能够引导初学者快速入门,感受编程的快乐和成就感,增强进一步学习的信心,从而快速成为编程高手。

实例丰富

通过例子学习是最好的学习方式。本书通过一个知识点、一个例子、一个结果、一段评析、一个综合应用的模式,透彻、详尽地讲述了实际开发中所需的各类知识。

操作性强

书中几乎每章都提供了大量案例,帮助读者实践与练习。通过反复上机练习,读者可以重新回顾、熟悉所学的知识,举一反三,为进一步学习做好充分的准备。

本书显著特色

体验好

二维码扫一扫,随时随地看视频。书中几乎每个章节都提供了二维码,读者朋友可以通过手机微信扫一扫,随时随地看相关的教学视频(若个别手机不能播放,请参考前言中的“本书学习资源列表及获取方式”下载后在计算机上可以一样观看)。

资源多

从配套到拓展,资源库一应俱全。本书提供了几乎覆盖全书的配套视频和素材源文件。还提供了拓展的学习资源,如习题及面试题库、案例库、工具库、网页模板库、网页配色库、网页素材库、网页案例欣赏库等,拓展视野、贴近实战,学习资源一网打尽!

案例多

案例丰富详尽,边做边学更快捷。跟着大量的案例去学习,边学边做,从做中学,使学习更深入、更高效。

入门易

遵循学习规律,入门与实战相结合。本书编写模式采用“基础知识+中小实例+实战案例”的形式,内容由浅入深、循序渐进,从入门中学习实战应用,从实战应用中激发学习兴趣。

服务快

提供在线服务,随时随地可交流。本书提供QQ群、网站下载等多渠道贴心服务。

本书学习资源列表及获取方式

本书的学习资源十分丰富,全部资源分布如下:

配套资源

- (1) 本书的配套同步视频,共计 512 节(可用二维码扫描观看或从下述的网站下载)。
- (2) 本书的素材及源程序,共计 995 项。

拓展学习资源

- (1) 习题及面试题库(共计 1 000 题)。
- (2) 案例库(各类案例 4 396 个)。
- (3) 工具库(HTML 参考手册 11 部、CSS 参考手册 10 部、JavaScript 参考手册 26 部)。
- (4) 网页模板库(各类模板 1 636 个)。
- (5) 网页素材库(17 大类)。
- (6) 网页配色库(623 项)。
- (7) 网页欣赏案例库(共计 508 例)。

☞ 以上资源的获取及联系方式

(1) 读者朋友可以加入本书微信公众号咨询关于本书的所有问题。



(2) 登录网站 xue.bookln.cn, 输入书名, 搜索到本书后下载。

(3) 加入本书学习交流专业解答 QQ 群: 621135618, 获取网盘下载地址和密码。

(4) 读者朋友还可通过电子邮件 weilaitushu@126.com、945694286@qq.com 与我们联系。

(5) 登录中国水利水电出版社的官方网站: www.waterpub.com.cn/softdown/, 找到本书后, 根据相关提示下载。

本书约定

运行本书示例, 需要下列软件:

☐ Windows 2000、Windows Server 2003、Windows XP、Windows Vista、Window7、Windows10 或 Mac OS X。

☐ IE 5.5 或更高版本。

☐ Mozilla 1.0 或更高版本。

☐ Opera 7.5 或更高版本。

☐ Safari 1.2 或更高版本。

为了节省版面, 本书所显示的示例代码都是局部的, 读者需要在网页中输入 `<script>` 标签, 然后尝试把书中列举的 JavaScript 脚本代码写在 `<script>` 标签内, 在 Web 浏览器中试验, 以验证代码运行效果。针对部分示例可能需要服务器端的配合, 读者可参阅示例所在章节的说明进行操作。

本书适用对象

本书适用于 JavaScript 从入门到高级程序设计的读者, 适用于网页设计、网页制作、网站建设、Web 前端开发和后台设计人员, 也可以作为高等院校相关专业的教学参考书, 或作为相关机构的培训教材。

关于作者

未来科技是由一群热爱 Web 开发的青年骨干教师组成的一个松散组织, 主要从事 Web 开发、教学培训、教材开发等业务。该群体编写的同类图书在很多网店上的销量名列前茅, 让数十万的读者轻松跨进了 Web 开发的大门, 为 Web 开发的普及和应用做出了积极贡献。

参与本书编写的人员有: 李德光、刘坤、吴云、赵德志、马林、刘金、邹仲、谢党华、刘望、彭方强、雷海兰、郭靖、张卫其、杨艳、顾克明、班琦、蔡霞英、曾德剑、曾锦华、曾兰香、曾世宏、曾旺新、曾伟、常星、陈娣、陈凤娟、陈凤仪、陈福妹、陈国锋、陈海兰、陈华娟、陈金清、陈马路、陈石明、陈世超、陈世敏、陈文广等。

编者

目 录

Preface

第 1 章 JavaScript 基础..... 1	
 示例: 1 个	
1.1 JavaScript 概述..... 1	
1.1.1 JavaScript 发展历史..... 1	
1.1.2 ECMAScript 与 JavaScript 的 关系..... 1	
1.1.3 ECMAScript 版本变化..... 2	
1.1.4 ECMAScript5 和 ECMAScript6 ... 2	
1.2 JavaScript 相关概念..... 3	
1.2.1 JavaScript 核心..... 3	
1.2.2 文档对象模型..... 3	
1.2.3 浏览器对象模型..... 5	
第 2 章 初次使用 JavaScript..... 6	
 视频讲解: 9 个 示例: 16 个	
2.1 在网页中嵌入 JavaScript 脚本..... 6	
2.1.1 编写脚本..... 6	
2.1.2 脚本位置..... 9	
2.1.3 设置延迟执行..... 10	
2.1.4 设置异步响应..... 10	
2.2 执行 JavaScript 程序..... 11	
2.2.1 执行过程..... 11	
2.2.2 预编译..... 12	
2.2.3 代码块..... 13	
2.2.4 响应事件..... 14	
2.2.5 设计动态脚本..... 15	
第 3 章 代码测试和错误处理..... 16	
 视频讲解: 9 个 示例: 15 个	
3.1 浏览器与 JavaScript..... 16	
3.1.1 浏览器内核..... 16	
3.1.2 浏览器错误报告..... 16	
3.2 JavaScript 开发工具..... 20	
3.2.1 JavaScript 编辑器..... 20	
3.2.2 JavaScript 测试和调试..... 20	
3.2.3 使用控制台..... 23	
3.3 错误处理..... 24	
3.3.1 认识错误类型..... 25	
3.3.2 使用 try-catch..... 26	
3.3.3 使用 finally..... 27	
3.3.4 使用 throw..... 28	
3.3.5 抛出时机..... 28	
3.3.6 错误事件..... 29	
第 4 章 JavaScript 基本语法..... 31	
 视频讲解: 30 个 示例: 67 个	
4.1 基本词法..... 31	
4.1.1 字符编码..... 31	
4.1.2 区分大小写..... 32	
4.1.3 标识符..... 33	
4.1.4 直接量..... 33	
4.1.5 关键字和保留字..... 33	
4.1.6 分隔符..... 34	
4.1.7 注释..... 36	
4.1.8 转义序列..... 36	
4.2 使用变量..... 36	
4.2.1 声明变量..... 36	
4.2.2 赋值变量..... 37	
4.2.3 变量的作用域..... 38	
4.2.4 避免变量污染..... 39	
4.3 数据类型..... 40	
4.3.1 基本数据类型..... 40	
4.3.2 数值..... 40	
4.3.3 字符串..... 43	
4.3.4 布尔值..... 45	
4.3.5 Null..... 45	
4.3.6 Undefined..... 45	
4.4 严格模式..... 46	

4.4.1 启用严格模式	46	5.6.3 中括号和点号运算符	82
4.4.2 严格模式的执行限制	48	5.6.4 小括号运算符	84
4.5 实战案例	52	5.7 其他运算符	85
4.5.1 使用 <code>typeof</code> 检测类型	52	5.7.1 条件运算符	85
4.5.2 使用 <code>constructor</code> 检测类型	53	5.7.2 逗号运算符	85
4.5.3 封装类型检测方法: <code>toString()</code>	54	5.7.3 <code>void</code> 运算符	86
4.5.4 转换为字符串	55	5.8 实战案例	87
4.5.5 转换数字模式	57	5.8.1 使用表达式	87
4.5.6 设置数字显示的小数位数	57	5.8.2 连续运算	89
4.5.7 转换为数字	58	5.8.3 把命令转换为表达式	91
4.5.8 转换为布尔值	59	5.8.4 表达式中的函数	93
4.5.9 转换为对象	60		
4.5.10 把对象转换为值	61	第 6 章 设计程序结构	95
4.5.11 强制转换	62	 视频讲解: 23 个 示例: 67 个	
第 5 章 使用运算符	64	6.1 语句概述	95
 视频讲解: 26 个 示例: 77 个		6.1.1 表达式语句	96
5.1 运算符概述	64	6.1.2 复合语句	97
5.2 算术运算符	67	6.1.3 声明语句	98
5.2.1 加法运算	68	6.1.4 空语句	98
5.2.2 减法运算	68	6.2 分支结构	99
5.2.3 乘法运算	69	6.2.1 <code>if</code> 语句	99
5.2.4 除法运算	69	6.2.2 条件嵌套	100
5.2.5 余数运算	69	6.2.3 设计分支结构	101
5.2.6 取反运算	70	6.2.4 <code>switch</code> 语句	103
5.2.7 递增和递减	70	6.2.5 <code>default</code> 从句	104
5.3 逻辑运算符	71	6.2.6 比较 <code>if</code> 和 <code>switch</code> 结构	106
5.3.1 逻辑与运算符	71	6.2.7 优化分支结构	107
5.3.2 逻辑或运算符	72	6.3 循环结构	109
5.3.3 逻辑非运算符	73	6.3.1 <code>while</code> 语句	109
5.3.4 案例: 逻辑运算训练	74	6.3.2 <code>do/while</code> 语句	110
5.4 关系运算符	75	6.3.3 <code>for</code> 语句	111
5.4.1 大小比较	75	6.3.4 <code>for/in</code> 语句	112
5.4.2 案例: 包含检测	76	6.3.5 比较 <code>while</code> 和 <code>for</code> 结构	114
5.4.3 案例: 等值检测	77	6.3.6 优化循环结构	116
5.5 赋值运算符	78	6.4 结构跳转	118
5.6 对象操作运算符	80	6.4.1 标签语句	118
5.6.1 <code>new</code> 运算符	80	6.4.2 <code>break</code> 语句	118
5.6.2 <code>delete</code> 运算符	81	6.4.3 <code>continue</code> 语句	121
		6.5 实战案例	122
		6.5.1 提升分支运算性能	122

6.5.2 提升循环运算性能	123	8.2.2 调用函数	164
6.5.3 设计杨辉三角	125	8.2.3 函数作用域	165
第 7 章 使用数组	128	8.3 使用参数	166
 视频讲解: 21 个 示例: 71 个		8.3.1 定义参数	166
7.1 定义数组	128	8.3.2 使用 arguments 对象	167
7.1.1 构造数组	128	8.3.3 使用 callee 回调函数	168
7.1.2 数组直接量	129	8.3.4 应用 arguments 对象	169
7.2 使用数组	129	8.4 使用函数对象	170
7.2.1 存取数组元素	129	8.4.1 获取函数形参个数	171
7.2.2 数组长度	130	8.4.2 自定义属性	171
7.2.3 对象与数组	131	8.4.3 使用 call() 和 apply()	172
7.2.4 定义多维数组	133	8.4.4 使用 bind()	175
7.3 使用数组对象	134	8.5 使用 this	177
7.3.1 检索数组	134	8.5.1 this 用法	177
7.3.2 操作元素	134	8.5.2 this 安全策略	180
7.3.3 操作子数组	136	8.5.3 应用 this	183
7.3.4 数组排序	138	8.5.4 函数调用模式	185
7.3.5 使用排序函数	139	8.5.5 函数的标识符	187
7.3.6 数组与字符串的转换	141	8.6 使用闭包函数	188
7.3.7 定位	142	8.6.1 认识闭包函数	189
7.3.8 迭代	143	8.6.2 使用闭包	190
7.3.9 汇总	150	8.6.3 定义闭包存储器	192
7.4 实战案例	151	8.6.4 在事件处理中应用闭包	194
7.4.1 快速交换	152	8.7 实战案例	197
7.4.2 数组下标	152	8.7.1 绑定函数	197
7.4.3 扩展数组方法	153	8.7.2 链式语法	198
7.4.4 设计迭代器	154	8.7.3 函数节流	199
7.4.5 使用迭代器	156	8.7.4 分支函数	200
7.4.6 使用数组维度	157	8.7.5 惰性载入函数	201
第 8 章 使用函数	159	8.7.6 惰性求值	202
 视频讲解: 37 个 示例: 109 个		8.7.7 记忆	204
8.1 定义函数	159	8.7.8 构建模块	205
8.1.1 声明函数	159	8.7.9 柯里化	207
8.1.2 构造函数	160	8.7.10 高阶函数	208
8.1.3 函数直接量	160	8.7.11 递归运算	210
8.1.4 定义嵌套函数	161	8.7.12 尾递归算法	211
8.1.5 比较定义函数的方法	162	第 9 章 使用对象	213
8.2 使用函数	163	 视频讲解: 33 个 示例: 77 个	
8.2.1 函数返回值	164	9.1 创建对象	213

9.1.1	使用构造函数创建对象	213
9.1.2	使用对象直接量创建对象	214
9.1.3	使用 create() 方法创建对象	216
9.2	操作对象	217
9.2.1	引用对象	217
9.2.2	复制对象	217
9.2.3	克隆对象	218
9.2.4	销毁对象	219
9.3	操作属性	219
9.3.1	定义属性	219
9.3.2	访问属性	221
9.3.3	赋值属性	224
9.3.4	删除属性	224
9.3.5	使用方法	224
9.3.6	配置特性	226
9.3.7	检测特性	227
9.4	使用方法	227
9.4.1	使用 toString()	227
9.4.2	使用 valueOf()	228
9.4.3	检测私有属性	229
9.4.4	检测枚举属性	230
9.4.5	检测原型对象	231
9.4.6	静态方法	232
9.5	使用原型	234
9.5.1	定义原型	234
9.5.2	比较原型属性和本地属性	235
9.5.3	应用原型	237
9.5.4	原型域和原型域链	240
9.5.5	原型继承	242
9.5.6	扩展原型方法	243
9.6	实战案例	244
9.6.1	设计工厂模式	244
9.6.2	设计类继承	245
9.6.3	设计构造原型模式	248
9.6.4	设计动态原型模式	248
9.6.5	设计实例继承	249
9.6.6	惰性实例化	250
9.6.7	安全构造对象	252

第 10 章 BOM 操作 255

 视频讲解: 20 个 示例: 35 个

10.1	使用 window 对象	255
10.1.1	访问浏览器窗口	255
10.1.2	全局作用域	256
10.1.3	使用系统测试方法	256
10.1.4	打开和关闭窗口	258
10.1.5	使用框架集	260
10.1.6	控制窗口位置	262
10.1.7	控制窗口大小	263
10.1.8	使用定时器	264
10.2	使用 navigator 对象	267
10.2.1	浏览器检测方法	267
10.2.2	检测浏览器类型和版本号	268
10.2.3	检测客户操作系统	269
10.2.4	检测插件	270
10.3	使用 location 对象	271
10.4	使用 history 对象	273
10.5	使用 screen 对象	273
10.6	使用 document 对象	274
10.6.1	访问文档对象	275
10.6.2	动态生成文档内容	276
10.7	实战案例	277
10.7.1	使用远程脚本	277
10.7.2	设计远程交互	279
10.7.3	使用浮动框架	281
10.7.4	封装用户代理检测	283

第 11 章 DOM 操作 289

 视频讲解: 48 个 示例: 99 个

11.1	DOM 基础	289
11.2	使用节点	291
11.2.1	节点类型	291
11.2.2	节点名称和值	293
11.2.3	节点关系	294
11.2.4	访问节点	295
11.2.5	操作节点	298
11.3	使用文档节点	300
11.3.1	访问文档子节点	300
11.3.2	访问文档信息	301

11.3.3 访问文档元素	301	11.10 实战案例.....	344
11.3.4 访问文档集合	302	11.10.1 设计动态脚本.....	344
11.3.5 使用 HTML5 Document.....	302	11.10.2 使用 script 加载远程数据 ...	346
11.4 使用元素节点	303	11.10.3 使用 script 实现异步交互 ...	348
11.4.1 访问元素	303	11.10.4 使用 JSONP	351
11.4.2 遍历元素	304	11.10.5 设计动态表格.....	355
11.4.3 创建元素	306	11.10.6 访问 DOM 集合	357
11.4.4 复制节点	307	11.10.7 在微博分享选中文本	359
11.4.5 插入节点	308		
11.4.6 删除节点	310	第 12 章 事件处理	362
11.4.7 替换节点	312	 视频讲解: 34 个 示例: 57 个	
11.4.8 获取焦点元素	312	12.1 事件基础	362
11.4.9 检测包含节点	313	12.1.1 事件模型	362
11.5 使用文本节点	315	12.1.2 事件流	363
11.5.1 访问文本节点	315	12.1.3 事件类型	365
11.5.2 创建文本节点	316	12.1.4 绑定事件	367
11.5.3 操作文本节点	316	12.1.5 事件处理函数	368
11.5.4 读取 HTML 字符串	317	12.1.6 注册事件	370
11.5.5 插入 HTML 字符串	317	12.1.7 销毁事件	371
11.5.6 替换 HTML 字符串	319	12.1.8 使用 event 对象	373
11.5.7 插入文本	321	12.1.9 事件委托	375
11.6 使用文档片段节点	322	12.2 使用鼠标事件	377
11.7 使用属性节点	323	12.2.1 鼠标点击	377
11.7.1 访问属性节点	323	12.2.2 鼠标移动	378
11.7.2 读取属性值	325	12.2.3 鼠标经过	380
11.7.3 设置属性值	325	12.2.4 鼠标来源	380
11.7.4 删除属性	327	12.2.5 鼠标定位	381
11.7.5 使用类选择器	328	12.2.6 鼠标按键	384
11.7.6 自定义属性	328	12.3 使用键盘事件	384
11.8 使用范围	329	12.3.1 键盘事件属性	385
11.8.1 创建范围	329	12.3.2 键盘响应顺序	387
11.8.2 选择范围	330	12.4 使用页面事件	388
11.8.3 设置范围	334	12.4.1 页面初始化	388
11.8.4 操作范围内容	336	12.4.2 结构初始化	390
11.8.5 插入范围内容	338	12.4.3 页面卸载	391
11.8.6 折叠范围	339	12.4.4 窗口重置	392
11.8.7 比较范围	341	12.4.5 页面滚动	393
11.8.8 复制和清除范围	342	12.4.6 错误处理	393
11.9 使用 CSS 选择器	342	12.5 使用 UI 事件	394
		12.5.1 焦点处理	394

12.5.2	选择文本	395	14.1.5	字符串截取	444
12.5.3	字段值变化监测	395	14.1.6	字符串替换	446
12.5.4	提交表单	397	14.1.7	字符串大小转换	447
12.5.5	重置表单	399	14.1.8	字符串比较	447
12.5.6	剪贴板数据	400	14.1.9	字符串与数组转换	448
12.6	实战案例	401	14.1.10	字符串格式化	449
12.6.1	封装事件	401	14.1.11	字符编码和解码	449
12.6.2	模拟事件	404	14.1.12	Unicode 编码和解码	451
12.6.3	设计弹出对话框	406	14.2	实战案例	453
12.6.4	设计遮罩层	407	14.2.1	访问表单对象	453
12.6.5	自定义事件	408	14.2.2	访问表单元素	454
12.6.6	设计事件触发模型	409	14.2.3	访问字段属性	455
12.6.7	应用事件模型	411	14.2.4	访问文本框的值	457
第 13 章	使用正则表达式与表单验证	415	14.2.5	文本框过滤	459
	 视频讲解: 15 个 示例: 48 个		14.2.6	切换焦点	461
13.1	正则表达式操作基础	415	14.2.7	访问选择框的值	462
13.1.1	定义正则表达式	415	14.2.8	编辑选项	465
13.1.2	访问正则表达式对象	417	14.2.9	字符串替换的高级应用	467
13.1.3	执行匹配操作	418	14.2.10	字符串修剪	469
13.1.4	访问匹配信息	419	14.2.11	检测特殊字符	471
13.1.5	条件检测	421	14.2.12	自定义加密和解密	472
13.2	正则表达式语法基础	422	14.2.13	表单序列化	474
13.2.1	字符描述	422	14.2.14	设计文本编辑器	478
13.2.2	字符范围	423	第 15 章	CSS 脚本化与网页特效	483
13.2.3	选择操作	424		 视频讲解: 26 个 示例: 43 个	
13.2.4	重复类量词	425	15.1	CSS 脚本化基础	483
13.2.5	惰性模式	426	15.1.1	访问 CSS 行内样式	483
13.2.6	边界量词	427	15.1.2	使用 style 对象	484
13.2.7	声明量词	428	15.1.3	使用 styleSheets 对象	489
13.2.8	表达式分组	428	15.1.4	使用 selectorText 对象	492
13.2.9	子表达式引用	429	15.1.5	编辑样式	493
13.3	实战案例	431	15.1.6	添加样式	493
第 14 章	字符串处理与表单开发	437	15.1.7	访问计算样式	495
	 视频讲解: 26 个 示例: 48 个		15.2	元素大小	498
14.1	字符串操作基础	437	15.2.1	访问 CSS 宽度和高度	498
14.1.1	定义字符串	437	15.2.2	把值转换为整数	500
14.1.2	字符串的值和字符长度	439	15.2.3	使用 offsetWidth 和 offset Height	501
14.1.3	字符串连接	441	15.2.4	元素尺寸	503
14.1.4	字符串查找	441			

15.2.5	视图尺寸	506
15.2.6	窗口尺寸	507
15.3	位置偏移	508
15.3.1	窗口位置	508
15.3.2	相对位置	510
15.3.3	定位位置	511
15.3.4	设置偏移位置	511
15.3.5	设置相对位置	512
15.3.6	鼠标指针绝对位置	512
15.3.7	鼠标指针相对位置	513
15.3.8	滚动条位置	515
15.3.9	设置滚动条位置	515
15.4	显示隐藏	516
15.4.1	可见性	516
15.4.2	透明度	517
15.5	实战案例	518
15.5.1	滑动	518
15.5.2	渐隐渐显	519

第 16 章 使用 Ajax 实现异步通信 520

 视频讲解: 33 个 示例: 51 个

16.1	使用 XML 数据	520
16.1.1	新建 XML 文档	520
16.1.2	访问 XML 数据	521
16.1.3	创建 XML DOM 对象	523
16.1.4	加载 XML 数据	524
16.1.5	显示 XML 数据	526
16.1.6	案例: 在网页中显示 XML 数据	527
16.1.7	案例: 异步加载 XML 数据	529
16.2	使用 JSON 数据	531
16.2.1	JSON 结构	531
16.2.2	案例: JSON 与 XML 格式 比较	533
16.2.3	案例: JSON 数据优化	535
16.2.4	案例: 解析 JSON	537
16.2.5	案例: 序列化 JSON	538
16.3	使用 Ajax	541
16.3.1	HTTP 头部信息	541
16.3.2	定义 XMLHttpRequest 对象	543

16.3.3	建立 XMLHttpRequest 连接 ..	545
16.3.4	发送 GET 请求	546
16.3.5	发送 POST 请求	547
16.3.6	转换串行化字符串	548
16.3.7	跟踪状态	549
16.3.8	中止请求	550
16.3.9	获取 XML 数据	550
16.3.10	获取 HTML 文本	551
16.3.11	获取 JavaScript 脚本	552
16.3.12	获取 JSON 数据	552
16.3.13	获取纯文本	553
16.3.14	获取头部信息	553

16.4 实战案例 554

16.4.1	封装异步请求操作	554
16.4.2	动态显示提示信息	555
16.4.3	动态查询记录集	557
16.4.4	记录集分页显示	560
16.4.5	设计 Tab 面板	562
16.4.6	关键字匹配	565
16.4.7	使用灯标	568

第 17 章 本地数据存储 571

 视频讲解: 14 个 示例: 16 个

17.1	使用 cookie	571
17.1.1	写入 cookie 信息	571
17.1.2	读取 cookie 信息	573
17.1.3	修改和删除 cookie 信息	574
17.1.4	附加 cookie 信息	574
17.1.5	封装 cookie 操作	576
17.1.6	案例: 打字游戏	577
17.2	使用 Web Storage	579
17.2.1	基本操作	580
17.2.2	案例: 设计网页皮肤	581
17.2.3	案例: 跟踪 localStorage 数据	583
17.2.4	案例: 设计计数器	585
17.3	使用 Web SQL	586
17.3.1	基本操作	586
17.3.2	案例: 创建本地数据库	588
17.3.3	案例: 批量存储本地数据	590

17.4 实战案例	592	18.6.3 裁切图像	639
第 18 章 JavaScript 图形设计	599	18.6.4 图像平铺	640
 视频讲解: 35 个 示例: 45 个		18.6.5 像素处理	642
18.1 HTML5 canvas 基础	599	18.7 实战案例	643
18.1.1 在页面中插入 canvas 元素	599	18.7.1 设计 canvas 动画	643
18.1.2 绘制图形的基本方法	600	18.7.2 保存绘图	646
18.1.3 使用 canvas	601	第 19 章 离线应用	649
18.2 绘制图形	603	 视频讲解: 6 个 示例: 8 个	
18.2.1 绘制直线	603	19.1 HTML5 离线应用基础	649
18.2.2 绘制矩形	605	19.1.1 认识 HTML5 离线应用	649
18.2.3 绘制圆形	606	19.1.2 浏览器支持	650
18.2.4 绘制多边形	608	19.1.3 使用 manifest 文件	651
18.2.5 绘制曲线	609	19.1.4 使用离线缓存	654
18.2.6 绘制二次方曲线	610	19.1.5 监听离线存储	657
18.2.7 绘制三次方曲线	611	19.2 实战案例	659
18.3 设置图形样式	613	19.2.1 缓存首页	659
18.3.1 设置线型	613	19.2.2 离线编辑内容	661
18.3.2 绘制线性渐变	616	19.2.3 离线跟踪	666
18.3.3 绘制径向渐变	617	第 20 章 多线程处理	671
18.3.4 绘制图案	618	 视频讲解: 8 个 示例: 6 个	
18.3.5 设置不透明度	619	20.1 Web Workers 基础	671
18.3.6 设置阴影	620	20.1.1 认识 Web Workers	671
18.4 操作图形	621	20.1.2 浏览器支持	672
18.4.1 保存和恢复 canvas 状态	621	20.1.3 创建 Web Workers	673
18.4.2 清除绘图	622	20.1.4 Web Workers 通信	674
18.4.3 移动坐标	623	20.1.5 案例: 使用 Web Workers	675
18.4.4 旋转坐标	625	20.2 实战案例	678
18.4.5 缩放图形	626	20.2.1 后台运算	678
18.4.6 变换矩阵	627	20.2.2 数值过滤	680
18.4.7 组合图形	629	20.2.3 并发处理	682
18.4.8 裁切路径	632	20.2.4 线程通信	684
18.5 绘制文字	633	20.2.5 Fibonacci 数列运算	686
18.5.1 绘制填充文字	633	第 21 章 文件操作	688
18.5.2 设置文字属性	634	 视频讲解: 23 个 示例: 30 个	
18.5.3 绘制轮廓文字	634	21.1 访问文件域	688
18.5.4 测量宽度	635	21.2 使用 Blob 对象	689
18.6 绘制图像	636	21.2.1 在文件域中访问 Blob 对象	689
18.6.1 导入图像	636	21.2.2 创建 Blob 对象	691
18.6.2 变换图像	638		

21.2.3 截取 Blob 对象	693	21.5.14 使用 filesystem:URL	726
21.2.4 保存 Blob 对象	694	21.6 实战案例	728
21.3 使用 FileReader 对象	696	第 22 章 使用 History	732
21.3.1 读取并显示文件	696	 视频讲解: 4 个 示例: 1 个	
21.3.2 监测读取操作	698	22.1 History API 基础	732
21.4 使用缓存对象	700	22.1.1 History API 处理方式	732
21.4.1 使用 ArrayBuffer 对象	700	22.1.2 浏览器兼容和扩展	733
21.4.2 使用 ArrayBufferView 对象	700	22.1.3 操作历史记录	733
21.4.3 使用 DataView 对象	701	22.2 实战案例	735
21.5 使用 FileSystem	704	22.2.1 设计无刷新页面导航	735
21.5.1 访问文件系统	704	22.2.2 设计主题宣传网站	738
21.5.2 申请配额	706	22.2.3 设计图片画廊	741
21.5.3 创建文件	710	22.2.4 设计历史恢复	744
21.5.4 写入文件	712	第 23 章 实战案例	748
21.5.5 添加数据	713	 视频讲解: 6 个 示例: 8 个	
21.5.6 读取文件	715	23.1 设计折叠面板	748
21.5.7 复制文件	716	23.2 设计计算器	752
21.5.8 删除文件	717	23.3 设计万年历	757
21.5.9 创建目录	718	23.4 设计俄罗斯方块	765
21.5.10 读取目录	720	23.4.1 设计游戏界面	766
21.5.11 删除目录	722	23.4.2 设计游戏模型	767
21.5.12 复制目录	723	23.4.3 实现游戏功能	769
21.5.13 移动和重命名目录	724		

第 1 章 JavaScript 基础

JavaScript 是面向 Web 的编程语言，并获得了所有网页浏览器的支持，是目前应用最广泛的编程语言之一，也是网页设计和 Web 应用必须掌握的基本工具。本章将简单介绍 JavaScript 的发展历史、概况以及相关基本概念，为后面的学习奠定知识基础。

【学习重点】

- 了解 JavaScript 发展历史。
- 了解 ECMAScript。
- 了解 JavaScript 实现构成。

1.1 JavaScript 概述

1.1.1 JavaScript 发展历史

1995 年 2 月 Netscape 公司发布 Netscape Navigator 2 浏览器，并开发了一种名为 LiveScript 的脚本语言。为了搭上媒体热炒 Java 的顺风车，临时把 LiveScript 改名为 JavaScript，这也是最初的 JavaScript 1.0 版本。

由于 JavaScript 1.0 获得了巨大成功，Netscape 公司随即在 Netscape Navigator 3 中又发布了 JavaScript 1.1 版本。

在 Netscape Navigator 3 发布后不久，微软在 Internet Explorer 3 中加入 JavaScript 脚本语言。为了避免与 Netscape 的 JavaScript 产生纠纷，微软特意将其命名为 JScript。

1997 年，欧洲计算机制造商协会（ECMA）以 JavaScript 1.1 为蓝本制订了 ECMA-262 新脚本语言的标准，并命名为 ECMAScript。

1998 年，国际标准化组织和国际电工委员会（ISO/IEC）也采用了 ECMAScript 作为标准（即 ISO/IEC-16262）。自此以后，浏览器开发商就开始致力于将 ECMAScript 作为各自 JavaScript 实现的参考标准。

1.1.2 ECMAScript 与 JavaScript 的关系

1997 年，ECMA 发布 262 号标准文件（ECMA-262）的第 1 版，规定了浏览器脚本语言的标准，并将这种语言命名为 ECMAScript，这个版本就是 ECMAScript 1.0 版。之所以不叫 JavaScript，有以下两个原因。

一是商标，Java 是 Sun 公司的商标，根据授权协议，只有 Netscape 公司可以合法地使用 JavaScript 这个名字，且 JavaScript 本身也已经被 Netscape 公司注册为商标。

二是体现这门语言的制定者是 ECMA，而不是 Netscape，这样有利于保证这门语言的开放性和中立性。

因此，ECMAScript 和 JavaScript 的关系是：ECMAScript 是 JavaScript 语言的国际标准，JavaScript 是 ECMAScript 的一种实现。

但是在一般场合中，这两个词是可以互换的。

1.1.3 ECMAScript 版本变化

1998 年 6 月, ECMAScript 2.0 版发布。

1999 年 12 月, ECMAScript 3.0 版发布, 成为 JavaScript 的通行标准, 得到了广泛支持。

2007 年 10 月, ECMAScript 4.0 版草案发布, 对 3.0 版做了大幅升级。由于 4.0 版的目标过于激进, 各方对于是否通过这个标准, 发生了严重分歧。

2008 年 7 月, ECMA 中止 ECMAScript 4.0 的开发, 将其中涉及现有功能改善的一小部分发布为 ECMAScript 3.1。不久, ECMAScript 3.1 更名为 ECMAScript 5。

2009 年 12 月, ECMAScript 5.0 版正式发布。

2011 年 6 月, ECMAScript 5.1 版发布, 并且成为 ISO 国际标准 (ISO/IEC 16262:2011)。

2013 年 12 月, ECMAScript 6 草案发布。

2015 年 6 月, ECMAScript 6 发布正式版本, 并更名为 ECMAScript 2015。Mozilla 在这个标准的基础上, 推出 JavaScript 2.0。从此以后, JavaScript 以年份命名, 新版本将按照 ECMAScript+年份的形式发布, 以便更频繁地发布包含小规模增量更新的新版本。

1.1.4 ECMAScript5 和 ECMAScript6

ECMAScript 5.1 (或 ECMAScript 5) 是 ECMAScript 标准的最新修正。与 HTML5 规范进程本质类似, ECMAScript 5 通过对现有 JavaScript 方法添加语句和对原生 ECMAScript 对象进行合并实现了标准化。此外, ECMAScript 5 还引入了一个语法的严格变种, 被称为严格模式 (Strict Mode)。

读者可以查看 ECMAScript 5 完整列表官方 ECMAScript 语言规范附录 D (<http://125.39.35.137/files/1119000004 CFAC44>) 和附录 E (www.ecma-international.org/publications/files/ECMAST/Ecma-262.pdf), 或者查看 <http://www.ecmascript.org/>, 还可以以 HTML 形式查看 Michael[tm] Smith 非官方的 HTML 版本说明 (<http://es5.github.io/>)。

随着 Opera 11.60 的发布, 五大主流浏览器都已支持 ECMAScript 5。具体信息如下:

- ✎ Opera 11.60。
- ✎ Internet Explorer 9。
- ✎ Firefox 4。
- ✎ Safari 5.1。
- ✎ Chrome 13。

提示:

IE9 不支持严格模式, 直到 IE10 才开始支持; Safari 5.1 仍不支持 `Function.prototype.bind`, 尽管 `Function.prototype.bind` 已经被 Webkit 所支持。

对于旧版浏览器的支持信息, 可以查看 Juriy Zaytsev 的 ECMAScript 5 兼容性列表 (<http://kangax.github.io/compat-table/es5/>)。

ECMAScript 6 是继 ECMAScript 5 之后的一次重大改进, 语言规范由 ECMAScript 5.1 时代的 245 页扩充至 600 页。ECMAScript 6 新增了许多必要的特性, 如模块和类, 以及一些实用特性, 如 Maps、Sets、Promises、生成器 (Generators) 等。尽管 ECMAScript 6 做了大量的更新, 但是它依旧完全向后兼容以前的版本 (这主要是缘于标准化委员会决定避免由不兼容版本语言导致的 Web 体验破碎), 因此所有老代码都可以正常运行, 整个过渡也显得更为平滑, 但随之而来的问题是, 开发者们抱怨了多年的老问题依然存在。

由于 ECMAScript 6 还没有定案, 有些语法规则还会变动, 目前支持 ECMAScript 6 的软件和开发环境还不多。各大浏览器的最新版本对 ECMAScript 6 的支持情况, 可以查看 <http://kangax.github.io/compat-table/es6/>。