

万维网权威大师亲自撰写，全面引领下一代万维网技术  
不容错过最新版JavaScript——ECMAScript5



张亚飞 编著

# JavaScript 权威指南

—ECMAScript5—HTML5BOM

## 专业级教学视频、实例源程序下载

- 全面破除当前普遍存在的错误认识，让您不走弯路
- 包含ECMAScript5最新语法，涵盖AJAX、DOM和ActiveXObject
- 与HTML5、CSS3紧密结合，紧跟时代进展
- 完整、系统、全面的知识体系，新功能、新特性一览无余
- 以近500余范例解释理论，清楚易懂，让你轻松上手
- 浏览器兼容性提示，让您学以致用，工作中游刃有余



清华大学出版社

# JavaScript 权威指南—— ECMAScript5+HTML5DOM+HTML5BOM

张亚飞 编 著

清华大学出版社

北 京

## 内 容 简 介

本书主要针对网页设计和开发初学者编写，现在学习 JavaScript 应该从最新标准 ECMAScript5 开始，这是毋庸置疑的。对于有经验的工程师来说，如果想完整掌握最新的 JavaScript 进展，那么使用本书是明智的选择。

本书包含了 JavaScript 开发的所有内容，包括 ECMAScript5、OOP、原型链、HTML5BOM、HTML5DOM、CoreDOM、StyleDOM、EventDOM、DHTML、XMLHttpRequest、AJAX、Web Workers、Web Storage、Web SQL Database、Cross Document Messaging、Server-Sent Event、Web Sockets、ActiveXObject、HTA 等。这些功能介绍都严格按照 ECMAScript5 和 HTML5 标准的要求编写，并附有详细的范例，侧重于实战。书中提供了各种浏览器兼容性指南，以帮助用户深入理解。

另外，本书修正了当前普遍存在的错误认识，有利于初学者的学习。

新的机遇已经到来，对于想尽快掌握 JavaScript 新技术的用户来说，本书是最完整的教程，希望本书能成为您学习道路上的领航者。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目（CIP）数据

JavaScript 权威指南——ECMAScript5+HTML5DOM+HTML5BOM/张亚飞编著. —北京：清华大学出版社，2011.9

ISBN 978-7-302-26428-6

I. ①J… II. ①张… III. ①JAVA 语言-程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 162912 号

责任编辑：刘利民

版式设计：文森时代

责任校对：姜彦 张彩凤

责任印制：王秀菊

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京密云胶印厂

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：185×260 印 张：40.5 字 数：1086 千字

版 次：2011 年 9 月第 1 版 印 次：2011 年 9 月第 1 次印刷

印 数：1~5000

定 价：69.80 元

# 前言

Preface



作为万维网前端最重要的程序语言，JavaScript 的出现使得网页和用户之间实现了一种实时、动态的交互关系，并且使网页可以包含更多活动的元素和更加精彩的呈现，此功能现在被称为 RIA（富 Internet 应用）。

所有的浏览器都嵌入有 JavaScript 解释引擎，这也使 JavaScript 成为创建 RIA 的首要程序语言。在过去的几年，AJAX 技术的出现使 JavaScript 成为富 Internet 应用的核心，随着社会各个组织机构向富 Internet 应用迁移，JavaScript 创造了无限的财富机遇，AJAX 也成为开发者推崇的解决方案。

## 新标准带给开发者更大的机遇

2009 年 12 月，在万维网主要参与者的推动下，ECMAScript5 正式被批准成为正式的标准，并提交给 ISO/IEC 制定为国际标准。这距离前一个版本——ECMAScript3 的产生已经过了将近 10 年，并且也成为近几年来最重大的语言更新，作为万维网前端最重要的程序语言，JavaScript 正迎来最大的发展机遇，每个万维网参与者都不能错过，每个开发者也不能错过。

## Web 标准就是 Web 的未来

ECMAScript 往往被称为 JavaScript，后者为更多开发者所熟知。但是 JavaScript 仅仅是浏览器厂商的一个产品商标的名称，而真正的标准是 ECMAScript。对于 ECMAScript，不同的浏览器厂商有不同的名称，如 Mozilla 称为 JavaScript，Microsoft 称为 JScript，Google 称为 V8，Apple 称为 JavaScriptCore，Opera 则直接称为 ECMAScript。

作为万维网的核心语言，HTML 在经历了近 10 年的停滞之后终于迎来了一个新的标准——HTML5。HTML5 不仅将成为 HTML4 的新标准，而且将成为 XHTML 和 HTMLDOM 的新标准。同时，HTML5 还为 BOM 第一次明确定义了标准，使得与浏览器交互有了可靠的依据。

所有这一切为浏览器成为应用程序的运行平台提供了坚实的基础，而这一切都需要编写 JavaScript 程序来实现。

## 市场前景巨大

ECMAScript5 正在快速成长，富 Internet 应用也正在向更广、更深的领域前进，所有最新版本的主流浏览器都宣布并已经开始支持 ECMAScript5，其中包括为 Android 和 iPhone 开发的移动浏览器。

这值得所有人密切关注，最近的一两年，会有很多公司进入这个领域。对于年轻一代开发者，ECMAScript5 应当是他们的首选技能，因为 ECMAScript5 会形成很大的市场，会有很多公司需要这方面的人才。



## 关于本书

本书涵盖了 JavaScript 开发的所有内容，包括 ECMAScript5、OOP、原型链、HTML5BOM、HTML5DOM、CoreDOM、StyleDOM、EventDOM、DHTML、XMLHttpRequest、AJAX、Web Workers、Web Storage、Web SQL Database、Cross Document Messaging、Server-Sent Event、Web Sockets、ActiveXObject、HTA 等，可以帮助读者完整、系统地掌握 JavaScript 的开发知识。

本书的作者是万维网开发的专家，也是《万维网应用程序开发纲要》的起草者，他可以帮助读者完整、系统、快速、有效地掌握 JavaScript 开发的知识。

## 保持与作者沟通

有时与作者沟通是十分必要的，用户可以从作者处获取知识的更新或者勘误（如果书中有的话），也可以让作者了解到用户的想法。作者目前有以下联系方式：

- E-mail  
zhang-yafei@hotmail.com
- BLOG  
http://zhangyafei.wordpress.com

## 发送邮件的注意事项

如果向作者发送邮件，请在“标题”处以以下格式书写，以便作者分类处理。

- 假定你对本书第 2 章中内容有疑问，只需在标题处按如下书写（请不要在标题处添加其他内容）：  
JavaScript5\_1\_Chapter\_02
- 假定你对本书第 12 章中内容有疑问，则如下书写：  
JavaScript5\_1\_Chapter\_12
- 假定你对本书附录 A 中内容有疑问，则如下书写：  
JavaScript5\_1\_Chapter\_A

## 获取本书教学视频和源程序

读者可以通过以下步骤获取本书的教学视频和源程序：

- (1) 登录清华大学出版社网站，网址为 <http://www.tup.com.cn>。
- (2) 在网站首页的右上角处的“高级搜索”文本框中输入本书书名，再单击“搜索”按钮。
- (3) 在显示本书相关信息的页面中，单击“网络资源”右侧的超链接，即可下载本书教学视频和源程序等网络资源。

# 目 录



## Contents

## 第 I 篇 ECMAScript 和 JavaScript 语言基础入门篇

第 1 章 正确认识 ECMAScript5 和 JavaScript.....	2
1.1 ECMAScript 和 JavaScript 的关系 ...	3
1.1.1 关于标准和实现.....	3
1.1.2 JavaScript 与 ECMAScript 版本的关系.....	4
1.2 有关 JavaScript 的认识.....	5
1.2.1 区分 JavaScript 和 JScript.....	5
1.2.2 JavaScript 不是解释型 Java 语言 .....	5
1.2.3 什么是脚本语言.....	6
1.2.4 区分客户端和服务端 JavaScript.....	6
第 2 章 准备开发环境和运行环境 .....	8
2.1 正确认识 JavaScript 的开发工具 .....	9
2.1.1 Dreamweaver 的安装和配置向导 .....	9
2.1.2 Aptana Studio 的安装和配置向导.....	10
2.2 准备执行环境 .....	11
2.2.1 浏览器的发展史.....	11
2.2.2 安装浏览器.....	12
2.2.3 在浏览器中启用 JavaScript 执行环境.....	14
第 3 章 在网页中编写 JavaScript 脚本代码 .....	15
3.1 创建第一个 JavaScript 应用程序 ...	16
3.1.1 使用 Dreamweaver 编写 JavaScript.....	16
3.1.2 使用 Aptana Studio 编写 JavaScript.....	18
3.2 在 HTML 网页中书写 JavaScript 脚本代码.....	22
3.2.1 使用 script 元素定义脚本代码...22	
3.2.2 在事件属性值中定义脚本.....	24
3.2.3 在超链接中定义脚本.....	27
3.3 浏览器不支持脚本时应注意的问题.....	28
3.3.1 noscript 元素.....	28
3.3.2 关于隐藏脚本数据.....	29
3.4 声明脚本语言 .....	29
3.4.1 默认的脚本语言 .....	29
3.4.2 脚本语言的本地声明.....	30
第 4 章 程序开发的基石——声明和使用变量.....	32
4.1 变量的声明 .....	33
4.1.1 声明和命名变量 .....	33
4.1.2 变量和标识符的命名规范.....	34
4.1.3 变量的数据类型 .....	34
4.1.4 自定义常量和内建常量.....	38
4.2 在程序中使用变量 .....	39
4.2.1 为变量赋值和更改变量值.....	39
4.2.2 使用变量和获取变量值.....	41
4.3 变量的传值和传址 .....	42
4.3.1 传值变量——按值传递.....	42
4.3.2 传址变量——按引用传递.....	43
4.4 内存使用规则——堆和栈 .....	43
4.4.1 栈的使用规则 .....	44
4.4.2 堆的使用规则 .....	44
4.4.3 易犯的错误 .....	45
4.5 数据类型转换 .....	46



Note

## 第5章 JavaScript 语法规范和良好的

编程习惯 .....	47
5.1 空白和多行书写 .....	48
5.2 点语法 .....	48
5.3 花括号 .....	49
5.4 分号 .....	49
5.5 圆括号 .....	50
5.6 字母的大小写 .....	50
5.7 程序注释 .....	51
5.7.1 单行注释和尾随注释 .....	51
5.7.2 多行注释 .....	51
5.7.3 文档注释 .....	52
5.7.4 HTML 注释 .....	52
5.8 关键字 .....	53
5.9 培养良好的编程习惯 .....	53
5.9.1 通用的变量名、方法名和 类名命名约定 .....	53
5.9.2 重构 .....	55
5.9.3 使用注释 .....	55
5.10 ECMAScript5 严格模式 .....	56
5.10.1 使用严格模式 .....	56
5.10.2 严格模式的特定要求 .....	57

  

第6章 数据运算和运算规则 .....	59
6.1 JavaScript 表达式 .....	60
6.1.1 简单表达式和复杂表达式 .....	60
6.1.2 赋值表达式和单值表达式 .....	60
6.2 算术运算符 .....	60
6.2.1 算术运算符的使用 .....	60
6.2.2 进行算术运算时的注意事项 .....	63
6.3 赋值运算符 .....	63
6.4 字符串运算符 .....	64
6.5 比较运算符和逻辑运算符 .....	65
6.5.1 数据类型的影响 .....	66
6.5.2 全等运算符 .....	66
6.5.3 特殊值的比较 .....	67
6.5.4 易犯错的比较运算 .....	68
6.6 位运算符 .....	68
6.7 其他运算符 .....	69
6.7.1 三元条件运算符 .....	69
6.7.2 typeof 运算符 .....	70
6.7.3 instanceof 运算符 .....	71
6.7.4 圆括号和逗号运算符的	

结合使用 .....	71
6.7.5 delete 运算符 .....	71
6.7.6 void 运算符 .....	72
6.8 运算符的使用规则 .....	72
6.8.1 运算符的优先级规则 .....	72
6.8.2 运算符的结合规则 .....	72
6.9 运行时进行类型检查 .....	73

  

第7章 流程控制 .....	75
7.1 语句和语句块 .....	76
7.2 if...else 条件语句 .....	76
7.3 switch 条件语句 .....	80
7.4 do...while 循环 .....	82
7.5 for 循环 .....	83
7.6 for...in 循环 .....	84
7.7 for each...in 循环 .....	85
7.8 其他流程控制 .....	86
7.8.1 break 语句 .....	86
7.8.2 continue 语句 .....	86
7.8.3 标签语句 .....	87
7.8.4 let 语句 .....	88
7.8.5 yield 语句 .....	88
7.9 with 语句 .....	89
7.10 流程控制总结 .....	90

  

第8章 内建的全局函数 .....	91
8.1 eval()函数 .....	92
8.2 转义操作函数 .....	92
8.2.1 escape()和 unescape()函数 .....	92
8.2.2 encodeURI()和 decodeURI() 函数 .....	93
8.2.3 encodeURIComponent()和 decodeURIComponent()函数 .....	93
8.3 转换函数 .....	94
8.3.1 parseFloat()函数 .....	94
8.3.2 parseInt()函数 .....	94
8.3.3 Number()和 String()函数 .....	94
8.3.4 Boolean()函数 .....	95
8.4 判断函数 .....	95
8.4.1 isFinite()函数 .....	95
8.4.2 isNaN()函数 .....	96

  

第9章 自定义函数 .....	97
9.1 函数的定义 .....	98



9.2 函数的调用 .....	98	9.5.1 了解变量的作用范围和 生命周期 .....	103
9.2.1 在函数中定义参数的初始值 .....	99	9.5.2 局部变量的声明 .....	104
9.2.2 嵌套函数的定义和调用 .....	99	9.5.3 全局变量的声明 .....	104
9.2.3 条件函数的定义和调用 .....	100	9.5.4 全局变量和局部变量的使用 .....	104
9.2.4 特殊的调用方式 .....	100	9.5.5 易出错的变量使用 .....	105
9.3 使用函数表达式定义函数 (匿名函数) .....	100	9.6 形参和实参 .....	106
9.3.1 函数表达式和函数语句的 区别 .....	100	9.7 闭包函数 (绑定函数) .....	107
9.3.2 匿名自执行函数 .....	102	9.7.1 什么是闭包函数 .....	107
9.4 使用 Function 类定义函数 .....	103	9.7.2 闭包函数常用的环境 .....	108
9.5 变量的作用范围和生命周期 .....	103	9.7.3 易犯的错误 .....	109
		9.8 产生器 .....	110



Note

## 第 II 篇 JavaScript 面向对象编程篇

### 第 10 章 面向对象编程 (OOP)

——创建类和使用类 .....	112
10.1 面向对象编程思想 .....	113
10.1.1 对象的抽象 .....	113
10.1.2 对象的使用 .....	113
10.1.3 易犯的错误 .....	114
10.1.4 面向对象程序设计的优点 和基本特征 .....	114
10.1.5 面向对象和结构化程序 设计的区别 .....	115
10.2 创建类 .....	116
10.2.1 创建一个 JavaScript 类 .....	116
10.2.2 类的定义 .....	116
10.2.3 关于构造器方法 .....	117
10.2.4 获取构造器方法 .....	118
10.3 类的使用 .....	119
10.3.1 创建类的实例 .....	119
10.3.2 访问对象属性 .....	120
10.3.3 调用对象方法 .....	121
10.3.4 调用静态方法和访问 静态属性 .....	122
10.4 使用 this 关键字定义方法和 属性 .....	122
10.5 使用 prototype 定义方法和 属性 .....	123
10.5.1 使用 prototype 定义方法 .....	123
10.5.2 使用 prototype 定义属性 .....	123

10.5.3 关于 prototype 和 __proto__ 属性 .....	124
10.6 使用 Object 类为实例定义方法 和属性 .....	124
10.6.1 使用 return 语句定义 方法和属性 .....	125
10.6.2 使用 prototype 和 Object 配合为类定义方法和属性 .....	125
10.7 使用 get 和 set 存取器方法 定义属性 .....	126
10.8 在类体内使用 this 关键字 .....	127
10.8.1 使用 this 时易犯的错误 .....	128
10.8.2 闭包方法 (绑定方法) 内的 this 关键字 .....	129
10.8.3 事件处理中的闭包 .....	130
10.8.4 另一个使用 this 时易犯的 错误 .....	131

### 第 11 章 建立更可靠的 OOP 程序—— 类和成员的访问控制 .....

### 第 12 章 重复使用类——继承和组合 ... 135

12.1 类的继承 .....	136
12.1.1 创建子类 .....	136
12.1.2 使用基类的构造器方法 .....	137
12.1.3 检测继承关系 .....	137
12.1.4 使用 \$super 实现访问基类 .....	137
12.1.5 创建子类时的注意事项 .....	139





Note

12.2	扩展内建 JavaScript 类.....	139			
12.3	类的组合 .....	139			
12.3.1	实现类的组合.....	139			
12.3.2	组合的优点.....	140			
12.3.3	了解 is-a 和 has-a.....	141			
12.3.4	关于聚合和组合.....	141			
<b>第 13 章 静态成员、静态类和枚举</b>					
	(Enumeration) .....	142			
13.1	定义静态成员和静态类的意义.....	143			
13.2	创建静态成员 .....	143			
13.2.1	使用静态成员.....	144			
13.2.2	静态成员和继承.....	144			
13.2.3	使用静态成员的注意事项....	145			
13.3	静态类及其声明方法 .....	145			
13.4	枚举 (Enumeration) .....	146			
13.4.1	创建枚举.....	146			
13.4.2	使用枚举的优点.....	146			
<b>第 14 章 多态 (Polymorphism)</b>					
	——重载和覆载.....	148			
14.1	方法的重载 (overLoad) .....	149			
14.1.1	重载参数列表的数量.....	149			
14.1.2	重载参数列表的数据类型....	150			
14.2	方法的覆载 (overRide) .....	151			
14.3	属性的覆载 .....	152			
<b>第 15 章 原型链本质论</b>					
15.1	原型链的根源 .....	154			
15.1.1	内部属性.....	154			
15.1.2	访问内部属性[Prototype]....	155			
15.2	prototype 的作用——原型链的实现.....	155			
15.2.1	实现继承.....	156			
15.2.2	分享属性 (和方法) .....	156			
15.2.3	__proto__ 和 prototype 属性的区别 .....	157			
15.2.4	使用 ECMAScript5 新增的原型处理功能.....	157			
15.3	检查原型链 .....	158			
15.3.1	查看对象是否在指定对象的原型链中.....	158			
15.3.2	查看指定对象是否定义了				
	特定属性和方法.....	159			
15.3.3	使用 hasOwnProperty()				
	方法过滤原型链.....	159			
15.3.4	查看指定对象是否定义了				
	特定属性且可以枚举.....	160			
<b>第 16 章 ECMAScript5 增强的对象模型</b>					
16.1	属性和描述符 .....	162			
16.1.1	易犯的错误 .....	162			
16.1.2	使用 ECMAScript 5 增强的对象模型定义属性.....	162			
16.1.3	定义 get 和 set 存取器方法....	163			
16.1.4	一次定义多个属性.....	164			
16.1.5	获取特定属性的属性描述符 .....	164			
16.2	枚举属性 .....	165			
16.2.1	Object.keys(obj)方法 .....	165			
16.2.2	Object.getOwnPropertyNames(obj)方法.....	165			
16.3	密封对象和动态对象 .....	166			
16.3.1	密封对象不允许添加成员....	166			
16.3.2	密封对象不允许添加和删除成员 .....	167			
16.3.3	完全密封对象 .....	168			
16.3.4	易犯的错误 .....	170			
16.4	根据原型创建对象 .....	171			
<b>第 17 章 处理字符串——String 类和正则表达式</b>					
17.1	正则表达式语法 .....	174			
17.1.1	正则表达式的基本语法.....	174			
17.1.2	优先权顺序 .....	179			
17.2	使用 RegExp 处理字符串正则匹配.....	179			
17.2.1	创建一个 RegExp 对象.....	179			
17.2.2	使用 exec()和 test()方法进行正则匹配 .....	180			
17.3	使用 String 类处理字符串.....	181			
17.3.1	String 类的方法和属性.....	182			
17.3.2	创建打字机效果.....	183			
17.3.3	使用 match()方法 .....	184			
17.3.4	使用 replace()方法 .....	185			



17.3.5 使用\$替换变量.....	185	18.7 Array 推知.....	208
17.3.6 使用 search()方法.....	186	<b>第 19 章 处理日期和时间.....</b>	<b>210</b>
17.3.7 使用 split()方法.....	187	19.1 使用 Date 类处理日期和时间....	211
17.3.8 创建 HTML 格式的字符串 ...	187	19.1.1 返回当前时间和日期.....	213
<b>第 18 章 数组、多维数组和复合数组</b>		19.1.2 返回特定的时间和日期.....	213
(哈希映射) .....	189	19.1.3 获取特定时间和日期字段....	213
18.1 创建数组 .....	190	19.1.4 设置特定时间和日期字段....	214
18.1.1 创建数组的基本方法.....	190	19.1.5 使用 parse()方法解析日期	
18.1.2 使用索引数组.....	190	字符串 .....	214
18.1.3 使用全局函数 Array()时的		19.1.6 使用 Date.UTC()方法返回	
注意事项.....	191	时间戳 .....	215
18.2 实现和使用多维数组 .....	192	19.2 实现日期输入控件 .....	215
18.3 使用 Array API 处理数组 .....	193	<b>第 20 章 其他 JavaScript 内建类</b>	
18.3.1 常用的范例.....	194	(核心对象) .....	220
18.3.2 ECMAScript 5 新增方法的		20.1 使用 Boolean 类处理逻辑值... ..	221
使用 .....	195	20.2 使用 Number 类进行数字类	
18.3.3 数组的排序.....	199	型转换.....	221
18.4 克隆数组 .....	201	20.3 使用 Math 类进行复杂的数学	
18.4.1 浅表克隆.....	201	运算.....	223
18.4.2 深度克隆.....	201	20.4 使用 Function 和 arguments 类... ..	225
18.5 创建和使用复合数组 .....	203	20.4.1 Function 类 .....	225
18.5.1 使用 Object 对象创建复合		20.4.2 Arguments 类和 arguments	
数组 .....	203	属性 .....	228
18.5.2 使用 Array 对象创建复合		20.5 JSON 操作.....	229
数组 .....	204	20.5.1 序列化 JavaScript 数据为	
18.5.3 遍历复合数组.....	204	JSON 数据 .....	230
18.6 哈希表 .....	205	20.5.2 逆序列化 JSON 数据为	
18.6.1 实现哈希表.....	205	JavaScript 数据 .....	232
18.6.2 使用哈希表.....	207		



## 第 III 篇 HTML5BOM 篇

<b>第 21 章 BOM 之源——window 对象....</b>	<b>234</b>	延迟调用 .....	236
21.1 定时器——间隔调用和		21.1.4 易犯的错误 .....	236
延迟调用 .....	235	21.1.5 强制清除 .....	237
21.1.1 使用 setInterval()和		<b>21.2 基本的交互操作 .....</b>	<b>237</b>
clearInterval()方法实现		21.2.1 警告对话框 .....	237
间隔调用.....	235	21.2.2 确认对话框 .....	238
21.1.2 范例——创建打字机效果....	235	21.2.3 提示输入消息对话框.....	238
21.1.3 使用 setTimeout()和		21.2.4 使用增强的交互式对话框....	239
clearTimeout()方法实现		21.2.5 范例——使用交互式	



Note

对话框相互传值.....	240	23.3.3 清除指定的键 .....	266
21.2.6 打印 .....	242	23.3.4 添加新键和更新键值.....	266
21.3 窗口交互操作 .....	243	23.4 storage 事件.....	267
21.3.1 打开新窗口 .....	243	23.5 Web SQL Database.....	268
21.3.2 关闭窗口 .....	245	23.5.1 第 1 个 Web SQL Database	
21.3.3 取消文档加载.....	245	应用 .....	268
21.3.4 获取窗口引用.....	245	23.5.2 使用 Web SQL Database 的	
21.3.5 调整窗口的焦点.....	246	基本步骤 .....	270
21.4 网页和浏览器交互 .....	246	23.6 异步数据库 .....	271
<b>第 22 章 BOM 基本应用——管理浏览</b>		23.6.1 执行数据库事务.....	272
<b>历史、解析地址和获取浏览</b>		23.6.2 数据库版本管理.....	274
<b>器信息 .....</b>	<b>247</b>	23.6.3 执行 SQL 语句 .....	274
22.1 浏览历史管理 .....	248	23.7 同步数据库 .....	276
22.2 解析地址 .....	249	23.7.1 执行数据库事务.....	276
22.3 获取浏览器信息 .....	251	23.7.2 数据库版本管理.....	277
22.3.1 遍历 navigator 对象成员 .....	251	23.7.3 执行 SQL 语句 .....	278
22.3.2 检测浏览器类型 .....	252	23.8 处理数据库操作结果 .....	278
22.3.3 自定义架构处理.....	253	23.9 错误和异常 .....	280
22.3.4 自定义内容类型处理.....	254	23.10 在开发人员工具中使用 SQL	
22.3.5 手动处理 storage 独占		语句.....	281
访问权 .....	255	23.11 SQLite SQL 语法和数据库	
22.3.6 检测脱机模式.....	255	的数据类型.....	281
22.4 非标准应用 .....	256	23.12 SQL 注入缺陷和参数化查询... 283	
22.4.1 screen 对象 .....	257	<b>第 24 章 BOM 新成就（2）——新的</b>	
22.4.2 clipboardData 对象.....	257	<b>服务器通信技术 .....</b>	<b>284</b>
<b>第 23 章 BOM 新成就（1）——客户端</b>		24.1 服务器推送技术 .....	285
<b>数据存储技术.....</b>	<b>259</b>	24.1.1 使用 Server-Sent Event 入门.. 285	
23.1 Web 存储 .....	260	24.1.2 服务器要求 .....	286
23.2 本地存储和会话存储 .....	260	24.1.3 使用 EventSource 对象 .....	287
23.2.1 在客户端持久存储数据 .....	260	24.1.4 在 Web Workers 中使用	
23.2.2 关于浏览器兼容.....	261	EventSource 对象 .....	288
23.2.3 在会话期存储数据 .....	262	24.2 套接字连接 .....	290
23.3 Storage 接口 .....	263	24.2.1 使用 Web Socket 入门.....	290
23.3.1 获取和更新存储.....	265	24.2.2 关于 Web Socket 通信协议... 292	
23.3.2 清除存储.....	266	24.2.3 使用 WebSocket 对象.....	293

## 第 IV 篇 DOM 篇

<b>第 25 章 HTML5DOM 新标准——处理</b>		25.1.1 DOM 基于节点树的表现	
<b>文档元信息和管理交互功能... 296</b>		形式 .....	297
25.1 了解 DOM.....	297	25.1.2 什么是 HTML DOM .....	299



- 25.1.3 HTML DOM 和核心 DOM 的区别与联系 ..... 299
- 25.1.4 什么是接口 ..... 299
- 25.2 document 对象 ..... 300
  - 25.2.1 处理文档元信息 ..... 301
  - 25.2.2 实用功能——处理 Cookie ..... 302
  - 25.2.3 检查文档状态 ..... 304
  - 25.2.4 开启和关闭文档 ..... 305
  - 25.2.5 易犯的错误——区分两个 open 方法 ..... 306
  - 25.2.6 向网页中写入内容 ..... 307
  - 25.2.7 清除网页中的内容 ..... 307
  - 25.2.8 实用功能——弹出通知窗口 ..... 307
  - 25.2.9 使整个文档都可编辑 ..... 308
  - 25.2.10 焦点管理 ..... 308
- 25.3 编辑 API ..... 309
  - 25.3.1 命令 ..... 309
  - 25.3.2 创建网页富文本编辑器 ..... 312
- 第 26 章 使用 CoreDOM 方法处理 HTML DOM 树 ..... 315
  - 26.1 一切皆是节点——学习 DOM 的首要认识 ..... 316
    - 26.1.1 JavaScript DOM API 的节点类型 ..... 316
    - 26.1.2 节点的层次关系 ..... 316
  - 26.2 处理 DOM 文档 ..... 318
  - 26.3 使用 DOM API 处理 HTML 文档 ..... 318
    - 26.3.1 基本的节点处理 ..... 321
    - 26.3.2 节点名、节点值和属性 ..... 323
    - 26.3.3 删除 HTML 文档中的元素节点、属性和内容 ..... 324
    - 26.3.4 使用 replaceChild() 方法替换节点 ..... 326
  - 26.4 主要接口的使用 ..... 327
    - 26.4.1 Document 接口 ..... 328
    - 26.4.2 Attr 接口 ..... 329
    - 26.4.3 CDATASection 接口 ..... 330
    - 26.4.4 CharacterData 接口 ..... 330
    - 26.4.5 Comment 接口 ..... 331
    - 26.4.6 DocumentFragment 接口 ..... 331
    - 26.4.7 DocumentType 接口 ..... 331
    - 26.4.8 DOMException 接口 ..... 332
    - 26.4.9 DOMImplementation 接口 ..... 333
    - 26.4.10 Element 接口 ..... 334
    - 26.4.11 Entity 接口 ..... 338
    - 26.4.12 EntityReference 接口 ..... 339
    - 26.4.13 Notation 接口 ..... 339
    - 26.4.14 ProcessingInstruction 接口 ..... 339
    - 26.4.15 Text 接口 ..... 340
    - 26.4.16 NodeList 接口 ..... 342
    - 26.4.17 NamedNodeMap 接口 ..... 342
- 第 27 章 使用 HTML5DOM 扩展 CoreDOM 方法 ..... 343
  - 27.1 使用 HTML5DOM 访问元素 ..... 344
    - 27.1.1 遍历节点树 ..... 344
    - 27.1.2 getElementsByTagName() 方法 ..... 346
    - 27.1.3 getElementByName() 方法和 getter 属性 ..... 347
    - 27.1.4 getElementById() 方法 ..... 348
    - 27.1.5 getElementByClassName() 方法 ..... 349
    - 27.1.6 关于 HTMLElement 接口 ..... 350
    - 27.1.7 innerHTML 属性、outerHTML 属性和 insertAdjacentHTML() 方法 ..... 351
  - 27.2 修改 HTML 文档 ..... 351
  - 27.3 根据 CSS 选择符访问元素 ..... 351
  - 27.4 表单处理 ..... 353
    - 27.4.1 处理输入控件 ..... 353
    - 27.4.2 获取上传文件 ..... 354
    - 27.4.3 使用 HTML5DOM API 验证控件 ..... 355
  - 27.5 HTML5DOM 通用接口的使用 ..... 357
    - 27.5.1 HTMLCollection 接口 ..... 358
    - 27.5.2 HTMLAllCollection 接口 ..... 358
    - 27.5.3 HTMLFormControlsCollection 接口 ..... 359
    - 27.5.4 HTMLOptionsCollection 接口 ..... 360
    - 27.5.5 HTMLPropertiesCollection 接口 ..... 360
    - 27.5.6 DOMTokenList 接口 ..... 361
    - 27.5.7 DOMSettableTokenList 接口 ..... 361



Note

## 第 28 章 使用 DHTML 和样式 DOM

27.5.8	DOMStringMap 接口	362
27.5.9	DOMException 接口	362
27.5.10	RadioNodeList 接口	363
<b>第 28 章 使用 DHTML 和样式 DOM</b>		
处理 CSS 样式表..... 364		
28.1	在 HTML 文档中应用 CSS	365
28.1.1	嵌入式	365
28.1.2	外部引用式	365
28.1.3	内联式	366
28.2	了解规则块、规则和样式	366
28.2.1	规则块	366
28.2.2	规则和样式	366
28.2.3	@规则	367
28.3	处理 CSS	367
28.3.1	使用 style 属性处理内联 CSS	367
28.3.2	处理 style 和 link 元素定义的样式表	369
28.3.3	处理导入的样式表	370
28.3.4	添加指令和规则	371
28.3.5	样式表相关对象	371
28.4	DOM 样式表	373
28.4.1	StyleSheet 接口	373
28.4.2	StyleSheetList 接口	374
28.4.3	MediaList 接口	374
28.4.4	LinkStyle 接口	375
28.4.5	DocumentStyle 接口	375
28.5	DOM CSS 规范	375
28.5.1	CSSStyleSheet 接口	375
28.5.2	CSSRuleList 接口	376
28.5.3	CSSRule 接口	376
28.5.4	CSSImportRule 接口	377
28.5.5	CSSStyleRule 接口	377
28.5.6	CSSStyleDeclaration 接口	377
28.6	兼容性指南	378
28.6.1	获取样式规则	378
28.6.2	获取导入的样式规则	378

28.6.3	新增@import 规则	380
28.6.4	新增@page 规则	380
28.6.5	新增样式规则	380
28.6.6	删除样式规则	381

## 第 29 章 事件 DOM..... 382

29.1	基本事件处理	383
29.1.1	事件处理的三要素	383
29.1.2	事件处理的实现	383
29.1.3	事件处理的总结	387
29.1.4	处理事件	387
29.2	事件流	388
29.2.1	捕获阶段 (capturing phase)	390
29.2.2	目标阶段 (targeting phase)	390
29.2.3	冒泡阶段 (bubbling phase)	390
29.2.4	可用的基本交互事件	392
29.3	事件对象	394
29.4	响应键盘事件	396
29.4.1	获取按键	396
29.4.2	响应组合键事件	397

## 第 30 章 DHTML 范例 ..... 398

30.1	滚动播出效果 (marquee)	399
30.1.1	竖直滚动的文本行	399
30.1.2	竖直滚动的图片	402
30.1.3	水平滚动的图片	403
30.2	常用的广告效果	405
30.2.1	广告的随机滚动出现	405
30.2.2	对联广告	407
30.3	导航菜单和选项卡面板	408
30.3.1	二级分层式导航菜单	409
30.3.2	弹出菜单式导航	414
30.3.3	选项卡面板	416
30.4	处理表单	418
30.4.1	表单复选框全选	418
30.4.2	表单的验证	420

# 第 V 篇 AJAX 篇

第 31 章	XMLHttpRequest 和 AJAX 解决方案	426
--------	----------------------------	-----

31.1	AJAX 的核心	427
31.2	AJAX 的使用	427
31.2.1	一个标准的 AJAX 应用	



所应具备的步骤.....	428	32.2.3 Spry 框架的组成部分.....	451
31.2.2 XMLHttpRequest 的方法和属性.....	429	32.3 jQuery 使用入门.....	452
31.2.3 一个标准的同步调用所应具备的步骤.....	432	32.3.1 第一个 jQuery 范例.....	452
31.2.4 编码混乱问题的解决.....	432	32.3.2 基本语法.....	453
31.2.5 跨浏览器兼容.....	433	<b>第 33 章 网页、网站和网络应用程序... 455</b>	
31.2.6 AJAX 的特点.....	435	33.1 安装和设置网络应用程序的运行环境.....	456
31.2.7 解决请求被浏览器缓存问题.....	435	33.1.1 安装 IIS.....	456
31.3 处理返回值中的 XML.....	435	33.1.2 通过服务器请求网页.....	456
31.3.1 处理解析错误.....	436	33.2 Web 服务器根目录和虚拟目录.....	457
31.3.2 DOMParser.....	437	33.2.1 Web 服务器根目录.....	457
31.3.3 XMLSerializer.....	438	33.2.2 虚拟目录.....	458
31.3.4 DOMImplementation.....	438	33.2.3 启用默认文件.....	461
31.4 跨域请求.....	439	33.3 安装 ASP.NET 应用程序服务器.....	462
31.4.1 使用代理实现跨域请求.....	440	<b>第 34 章 网页的请求和响应..... 464</b>	
31.4.2 使用 XDomainRequest.....	443	34.1 了解请求和响应的基本过程..	465
31.4.3 XDomainRequest 跨域访问控制.....	444	34.1.1 请求和响应的基本过程.....	465
<b>第 32 章 AJAX 框架——Spry 和 JQuery..... 446</b>		34.1.2 了解 URI、URL 和 URN ...	466
32.1 当前比较流行的 AJAX 框架.....	447	34.2 HTTP 协议基础.....	467
32.2 Spry 使用入门.....	447	34.2.1 了解 HTTP 消息的内容.....	467
32.2.1 第一个 Spry 范例.....	448	34.2.2 了解 HTTP 请求的内容.....	468
32.2.2 数据集的动态绑定.....	450	34.2.3 了解 HTTP 响应的内容.....	469
		34.2.4 了解 HTTP 消息报头的内容.....	470
		<b>第 VI 篇 安全性、执行效率、部署与实战篇</b>	
<b>第 35 章 JavaScript 安全和同源策略... 474</b>		36.2 控制内存.....	480
35.1 跨站脚本攻击 (XSS).....	475	36.3 优化代码的一般常识.....	481
35.1.1 如何防范跨站脚本攻击.....	475	36.4 页面重构——DOM Reflow 和 Repaint.....	483
35.1.2 使用 ASP.NET 的请求验证防范 XSS.....	476	36.4.1 关于 DOM Reflow 和 Repaint.....	483
35.2 相同 Origin 策略的限制——同源策略.....	477	36.4.2 优化 DOM Reflow.....	483
35.2.1 了解 Origin.....	477	36.4.3 优化 Repaint.....	484
35.2.2 放松相同起点限制.....	477	36.5 使用 Web Workers 优化 JavaScript 执行——多线程.....	484
<b>第 36 章 JavaScript 部署和高性能 JavaScript..... 479</b>		36.5.1 检查浏览器支持.....	484
36.1 文档瘦身.....	480	36.5.2 求素数的复杂运算.....	485
		36.5.3 使用 Web Workers 入门.....	486



Note

36.6	Web Workers 规范 API .....	487
36.6.1	Worker 及其类型 .....	487
36.6.2	Worker 中可以使用的 API .....	488
36.6.3	创建 Worker .....	490
36.6.4	处理运行时错误 .....	491
36.7	创建和使用专职 Worker .....	491
36.7.1	终止 Worker .....	491
36.7.2	消息的互传 .....	491
36.8	创建和使用共享 Worker .....	492
36.8.1	消息的互传 .....	493
36.8.2	连接同一个共享 Worker .....	494

### 第 37 章 处理程序异常——建立

	可靠的应用程序 .....	496
37.1	错误和异常情况 .....	497
37.2	异常情况处理 .....	498
37.2.1	使用 try...catch...finally 语句 .....	498
37.2.2	使用 throw 语句 .....	499
37.2.3	使用异常捕捉 .....	500
37.3	使用 Error 处理异常错误信息 .....	501
37.4	条件 catch 语句 .....	502

### 第 38 章 JavaScript 调试技术 .....

38.1	了解开发人员工具 .....	505
38.2	开发人员工具的组成和使用 ..	507
38.2.1	调试工具栏 .....	507
38.2.2	激活 JavaScript 调试 .....	507
38.2.3	主要内容窗口 .....	508
38.2.4	使用断点 .....	508
38.2.5	使用条件断点 .....	509
38.2.6	控制执行 .....	509
38.2.7	“局部变量”窗口 .....	510
38.2.8	“监视”窗口 .....	511
38.2.9	“调用堆栈”窗口 .....	511
38.3	使用脚本调试工具范例实战 ..	512
38.3.1	定义一个 JavaScript 网页 ..	512
38.3.2	调试实战 .....	513
38.4	“控制台”窗口 .....	517
38.4.1	使用“控制台”窗口 .....	517
38.4.2	对警报和错误消息进行 日志记录 .....	518

## 第 VII 篇 JavaScript 和 ActiveXObject 篇

### 第 39 章 使用 FSO 访问文件系统 .....

39.1	使用 FileSystemObject 类 .....	524
39.1.1	类和集合 .....	524
39.1.2	检查磁盘、文件和文件夹 是否存在 .....	524
39.1.3	访问磁盘、文件和文件夹 .....	525
39.1.4	关于 IE 专用的 Enumerator ..	530
39.2	处理文件夹 .....	531
39.3	处理文件 .....	532
39.3.1	创建文件 .....	533
39.3.2	向文件中添加数据 .....	534
39.3.3	读取文件内容 .....	535
39.3.4	TextStream 类成员概要 .....	536
39.3.5	移动、复制和删除文件 .....	536

### 第 40 章 使用 ADO 访问数据库 .....

40.1	安装和使用 Access 数据库 .....	539
40.1.1	安装 Access 数据库软件 .....	539
40.1.2	使用 Access 创建数据库 .....	540
40.2	理解数据库连接和配置	

### 数据源 .....

40.2.1	理解数据库连接 .....	542
40.2.2	理解数据库连接驱动 .....	543
40.2.3	通过 ODBC 连接数据库 .....	543
40.3	基本数据库查询操作 .....	546
40.3.1	查询和显示数据库记录 .....	546
40.3.2	遍历列出数据库内容 .....	547
40.4	修改数据 .....	548
40.4.1	插入数据记录 .....	548
40.4.2	更新数据记录 .....	549
40.4.3	删除数据记录 .....	550
40.5	操作数据库结构 .....	550
40.6	在 JavaScript 程序中应用 ADO 的基本步骤 .....	551
40.6.1	创建连接 .....	551
40.6.2	创建命令 .....	553
40.6.3	执行命令 .....	553
40.6.4	处理结果 .....	554
40.6.5	关闭连接 .....	554
40.7	SQL 语法入门 .....	554



40.7.1	SQL 语法的基础知识 .....	555	42.2.5	增加数据系列 .....	593
40.7.2	定义记录集的列 .....	556	42.2.6	使用 OWC 配合数据库查询 ..	594
40.7.3	限制记录集中的记录 .....	557	<b>第 43 章 使用 CDO 发送和接收</b>		
40.7.4	对记录集中的记录进行 排序 .....	558	电子邮件 .....		
40.7.5	联接表 .....	558	43.1 安装和使用 SMTP 服务 .....		
<b>第 41 章 使用 MSXML 处理 XML .....</b>			43.1.1 安装 SMTP 服务 .....		
41.1 在 JavaScript 应用程序中处理 XML 文档 .....			43.1.2 配置 SMTP 服务 .....		
41.1.1 了解 XML 文档对象 .....			43.1.3 配置 Outlook 使用本地 SMTP 服务发送邮件 .....		
41.1.2 DOM 基于节点的浏览 .....			43.1.4 区分 POP3 和 SMTP 服务 .....		
41.1.3 了解 MSXML DOM 组件 ..			43.2 创建发送邮件的 JavaScript 应用程序 .....		
41.1.4 使用 DOMDocument .....			43.2.1 了解 CDO .....		
41.2 创建和保存 XML 文档对象 ..			43.2.2 创建简单的邮件发送程序 ..		
41.2.1 解析字符串为 DOM .....			43.2.3 使用远程 SMTP 服务 .....		
41.2.2 解析 XML 文件为 XMLDOM .....			43.2.4 了解 CDO 组件成员 .....		
41.2.3 使用版本号创建 XMLDOM .....			43.2.5 SMTP 服务和 CDO 组件的 工作原理 .....		
41.2.4 保存和写出一个 XML 文档 对象 .....			43.3 创建增强的邮件发送程序 .....		
41.3 使用 XML 对象处理数据内容 ..			43.3.1 发送带附件的电子邮件 .....		
41.3.1 遍历节点树 .....			43.3.2 发送多媒体邮件 .....		
41.3.2 使用 getElementByTag Name()方法 .....			43.3.3 群发邮件——抄送和密送 .....		
41.3.3 使用 XPath 从 XML 文档中 提取数据 .....			43.4 使用 HTTP 方式收信 .....		
41.3.4 修改 XML 文档 .....			43.4.1 分拣邮件 .....		
41.4 使用 SAX 处理 XML 文档 .....			43.4.2 管理本地 SMTP 服务接收 邮件的基本原理 .....		
41.4.1 什么是 SAX .....			43.5 如何解决错误消息 .....		
41.4.2 SAX 的工作原理 .....			附录 A 保留的关键字 .....		
41.4.3 MSXML SAX 的使用 .....			附录 B 键盘键和键控代码值对照表 .....		
<b>第 42 章 使用 OWC 创建统计图表 .....</b>			附录 C W3C 的 XMLHttpRequest 接口规范 .....		
42.1 创建简单统计图表 .....			附录 D input 元素 type 属性值与 HTMLInputElement API 匹配索引 .....		
42.1.1 创建简单柱状图 .....			附录 E HTML5 元素和 HTML5DOM 接口的对应关系 .....		
42.1.2 创建立体柱状图 .....			附录 F HTML4 和 HTML5 事件属性 对照表 .....		
42.1.3 使用样式表创建柱状图 .....			附录 G 参考资料来源 .....		
42.2 使用 Office Web 组件 (OWC) 创建统计图表 .....					
42.2.1 安装 OWC .....					
42.2.2 使用 OWC .....					
42.2.3 OWC 组件的工作原理 .....					
42.2.4 创建饼图 .....					



Note



# ECMAScript 和 JavaScript 语言基础入门篇

JavaScript 是一种功能强大的程序开发语言，在几乎所有应用程序开发中，用户都要接触到 3 个相互关联的方面，JavaScript 也不例外，分别是：

(1) 开发环境——Dreamweaver、Aptana Studio、Eclipse 等。

(2) 运行环境——Internet Explorer、FireFox、Chrome 等浏览器。

(3) 开发语言——JavaScript。

本篇首先帮助用户判明这 3 个方面的基本关系，然后在此基础上使用 Dreamweaver 和 Aptana Studio 创作软件作为开发环境详细介绍 JavaScript 的基础知识，包括 JavaScript 语言的基本语法、变量、数据运算、流程控制、函数、异常处理和数据类型等。