

<<JavaScript权威指南>>

图书基本信息

书名：<<JavaScript权威指南>>

13位ISBN编号：9787302264286

10位ISBN编号：7302264287

出版时间：2011-9

出版时间：清华大学出版社

作者：张亚飞

页数：629

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

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。

## <<JavaScript权威指南>>

### 内容概要

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。

## <<JavaScript权威指南>>

### 作者简介

张亚飞——务实且极富远见的万维网权威大师——《万维网应用程序开发纲要》的起草者——多本权威书籍的撰写者，浸淫万维网15年，功力非凡

## 书籍目录

- 第1篇 ecmascript和javascript语言基础入门篇
  - 第1章 正确认识ecmascript5和javascript
    - 1.1 ecmascript和javascript的关系
      - 1.1.1 关于标准和实现
      - 1.1.2 javascript与ecmascript版本的关系
    - 1.2 有关javascript的认识
      - 1.2.1 区分javascript和jscript
      - 1.2.2 javascript不是解释型java语言
      - 1.2.3 什么是脚本语言
      - 1.2.4 区分客户端和服务端javascript
  - 第2章 准备开发环境和运行环境
    - 2.1 正确认识javascript的开发工具
      - 2.1.1 dreamweaver的安装和配置向导
      - 2.1.2 aptana studio的安装和配置向导
    - 2.2 准备执行环境
      - 2.2.1 浏览器的发展史
      - 2.2.2 安装浏览器
      - 2.2.3 在浏览器中启用javascript执行环境
  - 第3章 在网页中编写javascript脚本代码
    - 3.1 创建第一个javascript应用程序
      - 3.1.1 使用dreamweaver编写javascript
      - 3.1.2 使用aptana studio编写javascript
    - 3.2 在html网页中书写javascript脚本代码
      - 3.2.1 使用script元素定义脚本代码
      - 3.2.2 在事件属性值中定义脚本
      - 3.2.3 在超链接中定义脚本
    - 3.3 浏览器不支持脚本时应注意的问题
      - 3.3.1 noscript元素
      - 3.3.2 关于隐藏脚本数据
    - 3.4 声明脚本语言
      - 3.4.1 默认脚本语言
      - 3.4.2 脚本语言的本地声明
  - 第4章 程序开发的基石——声明和使用变量
    - 4.1 变量的声明
      - 4.1.1 声明和命名变量
      - 4.1.2 变量和标识符的命名规范
      - 4.1.3 变量的数据类型
      - 4.1.4 自定义常量和内建常量
    - 4.2 在程序中使用变量
      - 4.2.1 为变量赋值和更改变量值
      - 4.2.2 使用变量和获取变量值
    - 4.3 变量的传值和传址
      - 4.3.1 传值变量——按值传递
      - 4.3.2 传址变量——按引用传递
    - 4.4 内存使用规则——堆和栈

## &lt;&lt;JavaScript权威指南&gt;&gt;

- 4.4.1 栈的使用规则
- 4.4.2 堆的使用规则
- 4.4.3 易犯的错误
- 4.5 数据类型转换
- 第5章 javascript语法规则和良好的编程习惯
  - 5.1 空白和多行书写
  - 5.2 点语法
  - 5.3 花括号
  - 5.4 分号
  - 5.5 圆括号
  - 5.6 字母的大小写
  - 5.7 程序注释
    - 5.7.1 单行注释和尾随注释
    - 5.7.2 多行注释
    - 5.7.3 文档注释
    - 5.7.4 html注释
  - 5.8 关键字
  - 5.9 培养良好的编程习惯
    - 5.9.1 通用的变量名、方法名和类名命名约定
    - 5.9.2 重构
    - 5.9.3 使用注释
  - 5.10 ecmascript5严格模式
    - 5.10.1 使用严格模式
    - 5.10.2 严格模式的特定要求
- 第6章 数据运算和运算规则
  - 6.1 javascript表达式
    - 6.1.1 简单表达式和复杂表达式
    - 6.1.2 赋值表达式和单值表达式
  - 6.2 算术运算符
    - 6.2.1 算术运算符的使用
    - 6.2.2 进行算术运算时的注意事项
  - 6.3 赋值运算符
  - 6.4 字符串运算符
  - 6.5 比较运算符和逻辑运算符
    - 6.5.1 数据类型的影响
    - 6.5.2 全等运算符
    - 6.5.3 特殊值的比较
    - 6.5.4 易犯错的比较运算
  - 6.6 位运算符
  - 6.7 其他运算符
    - 6.7.1 三元条件运算符
    - 6.7.2 typeof运算符
    - 6.7.3 instanceof运算符
    - 6.7.4 圆括号和逗号运算符的结合使用
    - 6.7.5 delete运算符
    - 6.7.6 void运算符
  - 6.8 运算符的使用规则

## &lt;&lt;JavaScript权威指南&gt;&gt;

- 6.8.1 运算符的优先级规则
- 6.8.2 运算符的结合规则
- 6.9 运行时进行类型检查
- 第7章 流程控制
  - 7.1 语句和语句块
  - 7.2 if...else条件语句
  - 7.3 switch条件语句
  - 7.4 do...while循环
  - 7.5 for循环
  - 7.6 for...in循环
  - 7.7 for each...in循环
  - 7.8 其他流程控制
    - 7.8.1 break语句
    - 7.8.2 continue语句
    - 7.8.3 标签语句
    - 7.8.4 let语句
    - 7.8.5 yield语句
  - 7.9 with语句
  - 7.10 流程控制总结
- 第8章 内建的全局函数
  - 8.1 eval ( ) 函数
  - 8.2 转义操作函数
    - 8.2.1 escape ( ) 和unescape ( ) 函数
    - 8.2.2 encodeuri ( ) 和decodeuri ( ) 函数
    - 8.2.3 encodeuricomponent ( ) 和decodeuricomponent ( ) 函数
  - 8.3 转换函数
    - 8.3.1 parseFloat ( ) 函数
    - 8.3.2 parseInt ( ) 函数
    - 8.3.3 number ( ) 和string ( ) 函数
    - 8.3.4 boolean ( ) 函数
  - 8.4 判断函数
    - 8.4.1 isfinite ( ) 函数
    - 8.4.2 isnan ( ) 函数
- 第9章 自定义函数
  - 9.1 函数的定义
  - 9.2 函数的调用
    - 9.2.1 在函数中定义参数的初始值
    - 9.2.2 嵌套函数的定义和调用
    - 9.2.3 条件函数的定义和调用
    - 9.2.4 特殊的调用方式
  - 9.3 使用函数表达式定义函数 ( 匿名函数 )
    - 9.3.1 函数表达式和函数语句的区别
    - 9.3.2 匿名自执行函数
  - 9.4 使用function类定义函数
  - 9.5 变量的作用范围和生命周期
    - 9.5.1 了解变量的作用范围和生命周期
    - 9.5.2 局部变量的声明

## &lt;&lt;JavaScript权威指南&gt;&gt;

- 9.5.3 全局变量的声明
- 9.5.4 全局变量和局部变量的使用
- 9.5.5 易出错的变量使用

## 9.6 形参和实参

## 9.7 闭包函数（绑定函数）

- 9.7.1 什么是闭包函数
- 9.7.2 闭包函数常用的环境
- 9.7.3 易犯的错误

## 9.8 产生器

## 第2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

## 10.1 面向对象编程思想

- 10.1.1 对象的抽象
- 10.1.2 对象的使用
- 10.1.3 易犯的错误
- 10.1.4 面向对象程序设计的优点和基本特征
- 10.1.5 面向对象和结构化程序设计的区别

## 10.2 创建类

- 10.2.1 创建一个javascript类
- 10.2.2 类的定义
- 10.2.3 关于构造器方法
- 10.2.4 获取构造器方法

## 10.3 类的使用

- 10.3.1 创建类的实例
- 10.3.2 访问对象属性
- 10.3.3 调用对象方法
- 10.3.4 调用静态方法和访问静态属性

## 10.4 使用this关键字定义方法和属性

## 10.5 使用prototype定义方法和属性

- 10.5.1 使用prototype定义方法
- 10.5.2 使用prototype定义属性
- 10.5.3 关于prototype和\_\_proto\_\_属性

## 10.6 使用object类为实例定义方法和属性

- 10.6.1 使用return语句定义方法和属性
- 10.6.2 使用prototype和object配合为类定义方法和属性

## 10.7 使用get和set存取器方法定义属性

## 10.8 在类体内使用this关键字

- 10.8.1 使用this时易犯的错误
- 10.8.2 闭包方法（绑定方法）内的this关键字
- 10.8.3 事件处理中的闭包
- 10.8.4 另一个使用this时易犯的错误

## 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

## 12.1 类的继承

- 12.1.1 创建子类
- 12.1.2 使用基类的构造器方法
- 12.1.3 检测继承关系



## &lt;&lt;JavaScript权威指南&gt;&gt;

- 12.1.4 使用\$super实现访问基类
- 12.1.5 创建子类时的注意事项
- 12.2 扩展内建javascript类
- 12.3 类的组合
  - 12.3.1 实现类的组合
  - 12.3.2 组合的优点
  - 12.3.3 了解is-a和has-a
  - 12.3.4 关于聚合和组合
- 第13章 静态成员、静态类和枚举 (enumeration)
  - 13.1 定义静态成员和静态类的意义
  - 13.2 创建静态成员
    - 13.2.1 使用静态成员
    - 13.2.2 静态成员和继承
    - 13.2.3 使用静态成员的注意事项
  - 13.3 静态类及其声明方法
  - 13.4 枚举 (enumeration)
    - 13.4.1 创建枚举
    - 13.4.2 使用枚举的优点
- 第14章 多态 (polymorphism) ——重载和覆载
  - 14.1 方法的重载 (overload)
    - 14.1.1 重载参数列表的数量
    - 14.1.2 重载参数列表的数据类型
  - 14.2 方法的覆载 (override)
  - 14.3 属性的覆载
- 第15章 原型链本质论
  - 15.1 原型链的根源
    - 15.1.1 内部属性
    - 15.1.2 访问内部属性[prototype]
  - 15.2 prototype的作用——原型链的实现
    - 15.2.1 实现继承
    - 15.2.2 分享属性 (和方法)
    - 15.2.3 \_\_proto\_\_和prototype属性的区别
    - 15.2.4 使用ecmascript5新增的原型处理功能
  - 15.3 检查原型链
    - 15.3.1 查看对象是否在指定对象的原型链中
    - 15.3.2 查看指定对象是否定义了特定属性和方法
    - 15.3.3 使用hasownproperty ( )方法过滤原型链
    - 15.3.4 查看指定对象是否定义了特定属性且可以枚举
- 第16章 ecmascript5增强的对象模型
  - 16.1 属性和描述符
    - 16.1.1 易犯的错误
    - 16.1.2 使用ecmascript 5增强的对象模型定义属性
    - 16.1.3 定义get和set存取器方法
    - 16.1.4 一次定义多个属性
    - 16.1.5 获取特定属性的属性描述符
  - 16.2 枚举属性
    - 16.2.1 object.keys ( obj ) 方法

<<JavaScript权威指南>>

16.2.2 object.getownpropertynames ( obj ) 方法

16.3 密封对象和动态对象

16.3.1 密封对象不允许添加成员

16.3.2 密封对象不允许添加和删除成员

16.3.3 完全密封对象

16.3.4 易犯的错误

16.4 根据原型创建对象

第17章 处理字符串——string类和正则表达式

第3篇 html5bom篇

第4篇 dom篇

第5篇 javascript和activexobject篇

附录a 保留的关键字

附录b 键盘键和键控代码值对照表

附录c w3c的xmlhttprequest接口规范

附录d input元素type属性值与htmlinputelement api匹配索引

附录e html5元素和html5dom接口的对应关系

附录f html4和html5事件属性对照表

附录g 参考资料来源

## 章节摘录

版权页：插图：要了解JavaScript，首先要了解什么是脚本语言。

目前，动态的应用程序一般使用两种方式实现：二进制方式和脚本方式。

二进制（Binary）方式就是先将编写的程序代码编译为机器可识别的指令代码，然后再执行。

这种编译好的程序用户只能执行和使用，看不到原始程序的内容。

脚本（Script）方式是使用一种特定的描述性语言，依据一定的格式编写的文本文件。

简单地说，就是一条一条的文字命令，这些文字命令用户可以使用“记事本”程序看到。

脚本程序也是可执行文档，在执行时，由一个解释引擎（该解释引擎也是一个二进制的应用程序）将其逐条翻译成机器可识别的指令，并按程序顺序执行。

因为脚本在执行时多了一道翻译的程序，所以它比二进制程序的执行效率要稍低一些。

我们经常能看到的各种动态语言，如VBScript、JavaScript、Script、PHP、CGI、JSP、CFML等，都是脚本语言。

1. 客户端脚本在脚本语言中，有些是作为客户端脚本语言来运行的，它们由客户端的解释引擎来解释

。例如VB Script、JavaScript、Script等都可以作为客户端脚本语言，当它们被嵌入到HTML文件中时，可以按照顺序执行或者响应某个事件而对事件做出应答。

客户端脚本语言一般用来创建动画效果、执行简单的验证等，从而丰富了网页的呈现。

客户端脚本代码在客户端执行，可以在网页中增加动态效果和交互功能，丰富网页。

例如，下面是一些较常用的，实现动态效果和交互的简单功能：（1）当文档被加载到客户端后，文档中的脚本可以动态计算，从而动态地修改文档的内容。

（2）脚本可以用来获取表单控件中的输入数据。

一般，开发人员会使用这一功能来验证用户输入数据的有效性，如检测用户输入的密码是否至少6位长度等。

## <<JavaScript权威指南>>

### 编辑推荐

《JavaScript权威指南:ECMAScript5+HTML5DOM+HTML5BOM》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、WebWorkers、WebStorage、WebSQLDatabase、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。

《JavaScript权威指南:ECMAScript5+HTML5DOM+HTML5BOM》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。

全面破除当前普遍存在的错误认识，让您不走弯路，包含ECMAScripts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。

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。  
专业级教学视频，实例源程序下载。

<<JavaScript权威指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>