

图书基本信息

书名：《精通JavaScript》

13位ISBN编号：9787115175403

10位ISBN编号：7115175403

出版时间：2008-4-1

出版社：人民邮电出版社

作者：John Resig

页数：289

译者：江疆,陈贤安

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《精通JavaScript》

内容概要

本书是目前最深入的 JavaScript 图书，讲述了现代 JavaScript 的所有知识，展现了这门技术将能给网站建设带来如何丰富的体验。本书言简意赅，扩展了读者视野，并关注于基础且重要的主题——现代 JavaScript 是什么和不是什么，浏览器支持的当前状态，以及需要注意的陷阱等。书中所有概念都来自于现实案例的分析。

本书适合各层次 Web 开发人员阅读。

勘误 <http://realazy.org/jspro/erratra>

作者简介

John Resig，世界级Javascript大师，著名框架jQuery的创造者，拥有多年的Web应用程序开发经验。他现在是Mozilla公司的JavaScript传道者，此前曾经在非赢利组织“每个儿童一台笔记本电脑（OLPC）”担任程序员，并在罗切斯特理工学院从事过数据挖掘方面的研究。除了jQuery之外，他还开发7Sparklines程序库、AniWiki、建模工具dModeler等许多有影响的应用程序。

书籍目录

第一部分 认识现代javascript

第1章 现代javascript程序设计 2

1.1 面向对象的javascript 2

1.2 测试代码 3

1.3 打包分发 4

1.4 分离式dom脚本编程 5

1.4.1 dom 6

1.4.2 事件 7

1.4.3 javascript与css 8

1.5 ajax 8

1.6 浏览器支持 11

1.7 小结 12

第二部分 专业javascript开发

第2章 面向对象的javascript 14

2.1 语言特性 14

2.1.1 引用 14

2.1.2 函数重载和类型检查 16

2.1.3 作用域 19

2.1.4 闭包 20

2.1.5 上下文对象 23

2.2 面向对象基础 24

2.2.1 对象 25

2.2.2 对象的创建 25

2.3 小结 30

第3章 创建可重用代码 31

3.1 标准化面向对象的代码 31

3.1.1 原型式继承 31

3.1.2 类式继承 32

3.1.3 base库 35

3.1.4 prototype库 36

3.2 打包 39

3.2.1 命名空间 40

3.2.2 清理代码 42

3.2.3 压缩 43

3.3 分发 45

3.4 小结 47

第4章 调试与测试的工具 48

4.1 调试 48

4.1.1 错误控制台 48

4.1.2 dom查看器 52

4.1.3 firebug 54

4.1.4 venkman 55

4.2 测试 56

4.2.1 jsunit 56

4.2.2 j3unit 57

4.2.3 test.simple 58

4.3 小结 59

第三部分 分离式javascript

第5章 dom 62

5.1 dom简介 62

5.2 遍历dom 62

5.2.1 处理dom中的空格 64

5.2.2 简单的dom遍历 66

5.2.3 绑定到每一个html元素 67

5.2.4 标准的dom方法 68

5.3 等待html dom的加载 69

5.3.1 等待整个页面的加载 70

5.3.2 等待大部分dom的加载 70

5.3.3 判断dom何时加载完毕 71

5.4 在html文档中查找元素 73

5.4.1 通过类的值查找元素 73

5.4.2 使用css选择器查找元素 74

5.4.3 xpath 76

5.5 获取元素的内容 77

5.5.1 获取元素内的文本 77

5.5.2 获取元素内的html 78

5.6 操作元素特性 79

5.7 修改dom 82

5.7.1 使用dom创建节点 83

5.7.2 插入到dom中 83

5.7.3 注入html到dom 85

5.7.4 删除dom节点 87

5.8 小结 88

第6章 事件 89

6.1 javascript事件简介 89

6.1.1 异步事件与线程 89

6.1.2 事件阶段 91

6.2 常见事件特性 93

6.2.1 事件对象 93

6.2.2 this关键字 93

6.2.3 取消事件冒泡 94

6.2.4 重载浏览器的默认行为 95

6.3 绑定事件监听函数 97

6.3.1 传统绑定 98

6.3.2 dom绑定:w3c 99

6.3.3 dom绑定:ie 100

6.3.4 addevent和removeevent 100

6.4 事件类型 103

6.5 分离式脚本编程 103

6.5.1 javascript禁用的未雨绸缪 104

6.5.2 确保链接不依赖于javascript 104

6.5.3 监听css何时禁用 105

6.5.4 事件的亲和力 105

6.6 小结 106

第7章 javascript与css 107

7.1 访问样式信息 107

- 7.2 动态元素 109
 - 7.2.1 元素的位置 109
 - 7.2.2 元素的尺寸 115
 - 7.2.3 元素的可见性 117
- 7.3 动画 119
 - 7.3.1 滑动 119
 - 7.3.2 渐显 119
- 7.4 浏览器 120
 - 7.4.1 鼠标位置 120
 - 7.4.2 视口 121
- 7.5 拖放 123
- 7.6 库 128
 - 7.6.1 moo.fx和jquery 128
 - 7.6.2 scriptaculous 129
- 7.7 小结 131
- 第8章 改进表单 132
 - 8.1 表单验证 132
 - 8.1.1 必填字段 134
 - 8.1.2 模式匹配 136
 - 8.1.3 规则集合 138
 - 8.2 显示错误信息 139
 - 8.2.1 验证 140
 - 8.2.2 何时验证 142
 - 8.3 可用性的提升 144
 - 8.3.1 悬停的说明 144
 - 8.3.2 标记必填字段 146
 - 8.4 小结 147
- 第9章 制作图库 148
 - 9.1 图库示例 148
 - 9.1.1 lightbox 148
 - 9.1.2 thickbox 150
 - 9.2 制作图库 151
 - 9.2.1 分离加载 154
 - 9.2.2 半透明的覆盖层 155
 - 9.2.3 定位盒子 157
 - 9.2.4 导航 160
 - 9.2.5 幻灯片 162
 - 9.3 小结 165
- 第四部分 ajax
- 第10章 ajax导引 168
 - 10.1 使用ajax 168
 - 10.1.1 http请求 169
 - 10.1.2 http响应 173
 - 10.2 处理响应数据 176
 - 10.3 完整的ajax程序包 177
 - 10.4 数据的不同用途 179
 - 10.4.1 基于xml的rss feed 179
 - 10.4.2 html注入器 181
 - 10.4.3 json与javascript：远程执行 182

10.5	小结	182
第11章	用ajax改进blog	183
11.1	永不终止的blog	183
11.1.1	blog的模板	183
11.1.2	数据源	186
11.1.3	事件检测	187
11.1.4	请求	188
11.1.5	结果	188
11.2	实时网志	191
11.3	小结	193
第12章	自动补全的搜索	194
12.1	自动补全搜索的例子	194
12.2	制作页面	195
12.3	监听键盘输入	197
12.4	抓取结果	200
12.5	导航结果列表	202
12.5.1	键盘导航	202
12.5.2	鼠标导航	203
12.6	最终成果	203
12.7	小结	208
第13章	ajax wiki	209
13.1	wiki是什么	209
13.2	对话数据库	209
13.3	ajax请求	211
13.4	服务器端代码	212
13.4.1	处理请求	212
13.4.2	执行和格式化sql	213
13.5	处理json响应	215
13.6	附加的案例研究：javascript blog	216
13.7	应用程序的代码	217
13.7.1	核心javascript代码	218
13.7.2	javascript sql库	221
13.7.3	ruby服务器端代码	221
13.8	小结	224
第五部分	javascript的未来	
第14章	javascript路在何方	226
14.1	javascript 1.6与1.7	226
14.1.1	javascript 1.6	226
14.1.2	javascript 1.7	229
14.2	web applications 1.0	231
14.2.1	制作时钟	232
14.2.2	简单行星模拟	235
14.3	comet	238
14.4	小结	240
第六部分	附录	
附录a	dom参考手册	242
附录b	事件参考手册	257
附录c	浏览器	273

《精通JavaScript》

编辑推荐

原版问世不久就被业界视为经典，让读者大开眼界的JavaScript力作； 目前最深入的JavaScript图书，读者可跟随jQuery之父到达前所未有的深度； Amazon五星盛誉图书。

精彩短评

- 1、当年电话面试完，百姓网之后，被推荐读的
- 2、强烈建议 JSer 读读这本书
- 3、供自己多学习学习
- 4、相当于jquery源码分析
- 5、1-9章内容都不错，指导构建一个自己的js库，当然对jQuery的深入了解也有好处。10-13章ajax部分，作者从实际应用分析，所以综合性强，难度也增加了不少，有些代码不容易理解。对于初学者学习ajax，感觉不是最好的讲解，突然抛给你一大堆东西，没有循序渐进的过程。还有学习ajax，必须会一门后端语言。
- 6、不喜欢javascript面向对象的设计方式。
- 7、假装看完了吧。。。
- 8、不需要Hello World，够深入
- 9、书是好书，但是翻译的有几处没看懂
- 10、带你真正认识JavaScript, 第二章改变了我对JavaScript的印象，看完后感觉这门语言简直就是使Scheme语言看起来像C语言风格的版本。14章还有对JavaScript 1.7的介绍，引入了let关键字，更让我感到Scheme的亲切。
- 11、这本书真是读了好久才慢慢有点明白
- 12、jQuery作者的作品，详细介绍了javascript面向对象和闭包等特性，对于研究jquery源码有非常大的帮助。
- 13、大牛的代码即是艺术
- 14、jquery之父作品，还需深入理解。整体不错！
- 15、电子版，看完了，还不错
- 16、JS系Q2 John Resig jQuery作者。第2、3章不错。
- 17、作为入门经典和犀牛书不分伯仲
- 18、很实用
- 19、错别字有点多啊，有的代码也有错，总的来说，读读入门还不错
- 20、jquery之父写的js书,加深对jquery的理解,并且覆盖了很多其他的web开发方面的重要知识,图片处理,ajax,常用库等等
- 21、08年的书。哎。压力很大
- 22、jQ大神的书，颇有jQ源码分析的味道
- 23、1
- 24、深入JS从这里开始
- 25、后面章节的很多代码需要修改修改才能跑，还得研究研究。
- 26、学长推荐的书 可惜当当缺货了.. 读过之后 回来翻看目录 发现好多前面章节的内容都忘记了. 这书的代码实例 运行起来确实比较费劲。有时间重读下第二章往后的介绍基础知识的几章。
- 27、这可是jQuery作者写的js书....膜拜大牛。不过码神的书啥都一带而过不深究啊~~很蛋疼，不过还是偷了几个函数。
- 28、可能看这本书的时间有些晚吧，觉得很乏味。如果没有看过高程，这本书是不错的入门经典。
- 29、坑爹啊 这种书不应该是小学生读的么？完全就是一字典嘛，还定价49块，查字典啊。
- 30、基本就是一部关于jquery的一些创意集合。
- 31、可以一读，虽然有些过时了
- 32、全面的js教程 进阶的必备之选
- 33、淘宝上淘的中高级js绝版书，目前最深入的教程。其他的教材基本上都只是入门或者参考手册，不看这一本是没用的。
- 34、虽然说是jQuery之父写的，但却没有形成个系统，javascript的内部机制也没有说明白。远不如Crockford那本简短的javascript精华好。
- 35、js入门书。例子多好评。
- 36、循环次数 函数缓存 定时器 预加载 json-p ajax...

- 37、部分内容有点陈旧，但不失为一本好书。
- 38、比较旧的书了，但是有些代码还是看不懂。。代码也有不少错误，翻译的功底不错
- 39、还可以的一本书，可以作为一本js的入门后的小进阶书籍吧~
- 40、jQ之父的作品，介绍了不少js的编程技巧，而且其中相当的一部分都用到了jQ里面。看完这书后，再去看jQ的源码，感觉收获挺大的。毕竟，随着html5，css3，以及web标准的普及，前端如果只会重构或用一下jQ的话，基本是找不到工作的咯，所以学习js是大势所趋。
- 41、Javascript提高之书，泛读过后，了解了常用框架是如何设计的。
- 42、jquery之父的作品，前半部分挺有用但不够详细，后面就基本没啥意思
- 43、前几年读过，每次看书都有不同的体会。牛X的作者，但是翻译的一般把
- 44、新手进阶经典
- 45、不是入门书，适合有一定基础，正在深入学习js的人
- 46、非常棒！
- 47、和高级编程相比感觉有点少和浅
- 48、解决JS浏览器冲突的最佳实践。
- 49、这本书写于2006年,我当时还不会用电脑.John Resig就已经达到如此高的境界,书看似老.但里面的知识一点都没有过时,甚至就目前看来还有超前的部分.比如E4X,比如let,国内99%的jser到现在估计都没听说过这些.这是悲哀
- 50、很不错

精彩书评

1、学习JavaScript不久，很多JavaScript的核心概念都不怎么了解，对JavaScript的认知基本都是从W3C上面学到的，多是一些基本的语法。然后从图书馆借来JavaScript基础来看了下，很多内容都是懂得，于是自以为JavaScript的内容大致都已经了解了，只待应用了。看了这本书，才知道以前懂的只是皮毛，JavaScript的语言特性如闭包、引用等一概不知！现在就简单的谈谈自己学习这本书的收获吧（其实大部分是笔记而已）。1.JavaScript的语言特性1.1)引用（reference）：是一个指向实际对象的指针，对象包含的一系列属性只是其他对象的引用。需要注意的几点：(a)引用指向的只能是具体的对象，而不是另外一个引用，即单层引用。(b)若修改引用指向的对象，则指向该对象的所有其他引用也跟着改变；若该对象被置为新的对象，原来指向该对象的引用仍然保持指向旧的对象。如：`var a = "test"; //将a置为一个新的字符串对象var aRef = a; //aRef作为a的引用a+="ing"; //创建一个新的字符串对象alert(a != aRef); //a和aRef的值不相等`1.2)函数重载和类型检测函数重载的判断根据：参数个数和参数类型。参数个数可以根据arguments得到，参数类型可以根据typeof或者constructor判断（当变量类型为Object或者Array时，不能用typeof判断）。1.3)作用域在JavaScript中，作用域是由函数划分的，函数内部为局部变量，函数外边为全局变量，而所有属于全局作用域的变量都是window对象的property。但是如果变量没有显示定义，就算在函数内定义的，它也是全局变量。如：`function test(){f = "test"; }test(); //调用test来设置f的值alert(window.f == "test"); //f现在在全局作用域下`1.4)闭包闭包即内层的函数可以引用存在于包含它的函数内的变量。闭包能简化代码，让代码清晰。应用之一是函数的curry化。此不赘述。1.5)上下文对象--this2.对象Object对象有如下属性:(i)constructor：对创建对象的函数的引用（指针）。对于Object对象，该指针指向原始的Object()函数。(ii)prototype（原型）:对该对象的原型的引用。对于所有的对象，它默认返回Object对象的一个实例。所有对象原型的属性都能在该变量的每个实例中找到。给对象添加方法：(i)public method:使用prototype属性，这个属性可以作为所有新副本的基引用(base reference)。故public method都在prototype里添加。(ii)private method:定义一些只让对象内部访问，而外部访问不到的代码。可参考Private Members in JavaScript一文。(iii)privileged method:通过public method访问private method里面的变量和方法。(iv)static method:不能用该对象的实例来访问，而只能用该对象本身访问。类似Java中的静态方法。——代码示范：`function Test(string){ //创建一个Test构造函数this.s = string; }Test.staticMethodShow = function(){alert("this is static method show!"); }Test.prototype.show = function(){alert("public method show:"+this.s); }Test.staticMethodShow(); //this is static method show!var t = new Test("bbb");t.show(); //public method show:bbb}`——可以看到，static method与public method的区别——是否使用prototype属性。btw：通常我们使用function的prototype属性来创建对象的方法，而用普通方式来创建对象的属性。3.创建可重用代码3.1)JavaScript对象的创建与继承使用prototypal inheritance。这种方法的原理是对象的构造函数可以从其他对象中继承方法，原型对象创建后，所有其他的新对象都可以基于这个原型对象来构建。这种继承适用于单继承。——`function Person(name){this.name = name; }Person.prototype.getName = function(){return this.name; }function User(name, password){this.name = name;this.password = password; }User.prototype = new Person(); //User继承Person对象的方法User.prototype.getPassword = function(){return this.password; }var user = new User("aaa", "bbb");alert("name:"+user.getName()+" password:"+user.getPassword());`分析：User.prototype = new Person(); User是对User构造函数的引用。new Person()用Person的构造函数创建了新的Person对象，然后把User构造函数的原型置为这个操作的结果。——3.2)命名空间：`var YAHOO = {}; //创建一个默认的、全局的命名空间YAHOO.util = {}; //使用对象设置一些子命名空间YAHOO.util.Event = { //创建最终命名空间addEventListener:function(){...}};YAHOO.util.Event.addEventListener{.....}; //调用某个具体命名空间中的函数使用好命名空间，我们可以创建一个插件架构。可参考Dojo、YUI等。4.DOM这里简单说下DOM。有2种遍历DOM的方法：使用节点的parentNode，firstChild及nextSibling属性；使用getElementById()方法和getElementsByTagName()方法。获取元素内的文本：innerText或者nodeValue获取元素内的HTML`

: innerHTML。其中DOM的增删改此不赘述。5.事件5.1)JavaScript事件模型是异步事件模型，不是线程模型。对事件的处理分为两个阶段：捕获和冒泡。

```
<body><ul><li>Home</li><li>About</li></ul></body>onload =  
function(){var li = document.getElementsByTagName("li");for(var i = 0; i<li.length;  
i++){li[i].onmouseover = function(){this.style.backgroundColor = 'green'; };li[i].onmouseout =  
function(){this.style.backgroundColor = 'red'; };}}分析：事件的执行顺序：当用户点击<a>元素  
, document的点击处理函数先发生，然后是<body>的处理函数，然后是<ul>、<li>的处  
理函数，一直到<a>元素，这是事件的捕获阶段。一旦完成，再以<li><ul><body>  
和document的事件处理函数顺序触发，这是事件的冒泡阶段。JavaScript代码中，我们要注意每次鼠标  
悬停在<li>上时都会选中2个元素：<li>和它包含的<a>。而这种传统的事件绑定方法只支持  
冒泡，不支持捕获。即使代码中<a>元素添加了事件监听，事件仍会冒泡到其父元素<li>中  
的处理函数去，然后继续上升DOM树，直至DOM中所有相应事件处理完为止。那么，怎样才能阻止  
事件冒泡呢？
```

5.2)取消事件冒泡
function stopBubble(e){if(e && e.stopPropagation){ //W3C方式e.stopPropagation();}else{ //IE方
式window.event.cancelBubble = true;}}5.3)事件绑定监听函数：根据捕获和冒泡时处理事件的不同可分为
捕获型事件和冒泡型事件。1.传统绑定：即传统的事件处理函数。但传统绑定只会在事件冒泡中运行
, 而非捕获和冒泡都能运行；其次一个元素一次只能绑定一个事件处理函数，在使用window.onload时
, 可能出现意想不到的结果。2.W3C绑定：document.body.addEventListener('keypress',
myKeyPressHandler, false); //为文档的<body>绑定键盘点击事件处理函
数window.addEventListener('load', function(){.....}, false); //为页面绑定一个载入事件处理函数优点
：同时支持事件处理的捕获和冒泡阶段：false(冒泡), true(捕获); this关键字引用当前元素；可以绑定
多个事件。缺点：IE不支持。3.IE绑定：document.body.attachEvent('onkeypress',
myKeyPressHandler);window.attachEvent('load', function(){.....});优点：可以绑定多个事件。缺点：不
支持事件捕获；仅IE支持。此书剩余部分基本讲解了JavaScript的应用，这里就不说了。顺便吐槽一句
：看专业书还是看原版的好，翻译什么的还是少看吧！

2、这本书英文名是Pro JavaScript Techniques。是jQuery之父John Resig所写的。书中的内容确实比较高级些，并非如犀牛书般的基础参考而已。这本书看下来的感觉就是：果然是JQ之父写的！为什么这样说呢，因为这本书可以说是JQ的源码解读。这里的很多篇幅都可以看做JQ的一些方法的原型实现。包括选择器，attr,insert,append,remove等，还有event注册等方法。如果你想要体会JQ是如何炼成的，看这本书是最好不过的了，对于你对JS的理解会上一个层次，了解JQ编写的原理，体会到大师的想法。当然看了这个之后自然就看到了比大师更大师的，Dean Edwards (<http://dean.edwards.name/>)，这个牛人就是著名的JS压缩利器parker的作者。他写的cssQuery可以说是jQuery的最初的想法。但我们的John Resig很快发现光是选择器，比不过cssQuery，但是封装更丰富的操作那是很好很美妙的事情。同时，这本书也介绍了一些未来的javascript 1.6/1.7.这些已经部分的在Firefox中实现了。有兴趣的可以参考这里 (http://developer.mozilla.org/en/docs/JavaScript_Language_Resources)。文中另外的一些连接已经失效了取而代之找到了如下页面，这里还有1.8的特新(大部分中文)：

http://developer.mozilla.org/cn/docs/New_in_JavaScript_1.6

http://developer.mozilla.org/cn/docs/New_in_JavaScript_1.7

http://developer.mozilla.org/cn/docs/New_in_JavaScript_1.8

3、所谓文笔，并不是前阵子韩寒评论巴金冰心等人的“文笔”，只是说作者在讲解知识，分析问题的时候，不够流畅清晰。当然，这是相对于《ppk谈javascript》这本书而言的。我是读完ppk的书再买了这一本，因为前者说是初级的，后者是高级的。但是感觉，读完ppk的书，收获更多些。而Resig这一本，前几章的内容我大多都有所了解的了。可能我与众不同吧。后面几章，据说代码过多而分析太少，但我还没看到。就前面几章而言，代码占的比重也比较大。另外，如果对jquery比较熟悉的话，对书中的示例代码的功能会有一种心领神会的感觉。代码的构思很精巧，但是，有一些代码是有错误的。ps: John Resig的博客(<http://ejohn.org/>)我是必读的，有很多非常好的文章。

4、这本书的知识范围广泛，从这本书知道了很多之前不知道的事情，如JS中的编码规范，单元测试，打包发布等。值得一看。

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:www.tushu000.com