

主 编 汤 东 张富银
副主编 张元甜 江顺茂 龚 军



JavaScript

JavaScript SHIZHAN

实战



西南财经大学出版社
Southwest University of Finance & Economics Press

主 编○汤 东 张富银
副主编○张元甜 江顺茂 龚 军

JavaScript

实战

JavaScript SHIZHAN



西南财经大学出版社
Southwestern University of Finance & Economics Press

中国·成都

图书在版编目(CIP)数据

JavaScript 实战/汤东,张富银主编. —成都:西南财经大学出版社,
2016. 1

ISBN 978 - 7 - 5504 - 2172 - 1

I. ①J… II. ①汤…②张… III. ①JAVA 语言—程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 226010 号

JavaScript 实战

主 编:汤 东 张富银

副主编:张元甜 江顺茂 龚 军

责任编辑:植 苗

助理编辑:涂洪波

封面设计:何东琳设计工作室

责任印制:封俊川

出版发行	西南财经大学出版社(四川省成都市光华村街 55 号)
网 址	http://www. bookcj. com
电子邮件	bookcj@foxmail. com
邮政编码	610074
电 话	028 - 87353785 87352368
印 刷	四川森林印务有限责任公司
成品尺寸	185mm × 260mm
印 张	24. 5
字 数	575 千字
版 次	2016 年 1 月第 1 版
印 次	2016 年 1 月第 1 次印刷
印 数	1— 2000 册
书 号	ISBN 978 - 7 - 5504 - 2172 - 1
定 价	43. 00 元

1. 版权所有,翻印必究。
2. 如有印刷、装订等差错,可向本社营销部调换。
3. 本书封底无本社数码防伪标志,不得销售。

前 言

在互联网发展的早期, JavaScript 就已经成为支撑网页内容交互体验的基础技术。经过了大约 20 年的发展, JavaScript 的技术和能力都发生了天翻地覆的变化, 现在的 JavaScript 毫无疑问已经成了世界上使用范围最广的软件平台——互联网——的核心技术。

JavaScript 是 Web 开发中的一种脚本编程语言, 也是一种通用的、跨平台的、基于对象和事件驱动并具有安全性的脚本语言。它不需要进行编译, 而是直接嵌入 HTML 页面中, 把静态页面转变成支持用户交互并响应相应事件的动态页面。

本书的特点:

(1) 由浅入深, 循序渐进。本书以初、中级程序员为对象, 先从 JavaScript 基础学起, 再学习 JavaScript 的核心技术, 然后学习 JavaScript 的高级应用, 最后学习开发一个完整项目。讲解过程中步骤详尽, 版式新颖。

(2) 实例典型, 轻松易学。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。

(3) 应用实践, 随时练习。书中提供了实践与练习, 读者能够通过对问题的解答来回顾、熟悉所学的知识, 举一反三, 为进一步学习做好充分的准备。

本书由重庆电信职业学院汤东和成都源代码教育咨询公司(重庆分公司)张富银担任主编; 由重庆市丰都县职业教育中心张元甜、江顺茂和重庆市忠县职业教育中心龚军担任副主编。

感谢您购买本书, 希望本书能成为您编程路上的领航者。

目 录

第 1 章	JavaScript 概述	(1)
第 2 章	使用 JavaScript	(7)
第 3 章	语法、关键保留字及变量	(9)
第 4 章	数据类型	(13)
第 5 章	运算符	(22)
第 6 章	流程控制语句	(33)
第 7 章	函数	(39)
第 8 章	对象和数组	(42)
第 9 章	时间与日期	(49)
第 10 章	正则表达式	(52)
第 11 章	Function 类型	(63)
第 12 章	变量、作用域及内存	(68)
第 13 章	基本包装类型	(76)
第 14 章	内置对象	(82)
第 15 章	面向对象与原型	(87)
第 16 章	匿名函数和闭包	(104)
第 17 章	BOM	(114)
第 18 章	浏览器检测	(124)
第 19 章	DOM 基础	(135)
第 20 章	DOM 进阶	(146)
第 21 章	DOM 操作表格及样式	(154)
第 22 章	DOM 元素尺寸和位置	(165)
第 23 章	动态加载脚本和样式	(169)
第 24 章	事件入门	(172)
第 25 章	事件对象	(177)
第 26 章	事件绑定及深入	(185)

目 录

第 27 章 表单处理	(195)
第 28 章 错误处理与调试	(207)
第 29 章 Cookie 与存储	(216)
第 30 章 XML	(222)
第 31 章 XPath	(229)
第 32 章 JSON	(235)
第 33 章 Ajax	(239)
第 34 章 综合项目	(246)
项目 1 博客前端: 理解 JavaScript 库	(246)
项目 2 博客前端: 封装库——连缀	(248)
项目 3 博客前端: 封装库——CSS [上]	(250)
项目 4 博客前端: 封装库——CSS [下]	(252)
项目 5 博客前端: 封装库——下拉菜单	(254)
项目 6 博客前端: 封装库——弹出登录框	(256)
项目 7 博客前端: 封装库——遮罩锁屏	(257)
项目 8 博客前端: 封装库——拖拽 [上]	(259)
项目 9 博客前端: 封装库——拖拽 [下]	(260)
项目 10 博客前端: 封装库——事件绑定 [上]	(262)
项目 11 博客前端: 封装库——事件绑定 [中]	(264)
项目 12 博客前端: 封装库——事件绑定 [下]	(266)
项目 13 博客前端: 封装库——修缮拖拽	(268)
项目 14 博客前端: 封装库——插件	(269)
项目 15 博客前端: 封装库——CSS 选择器 [上]	(271)
项目 16 博客前端: 封装库——CSS 选择器 [下]	(273)
项目 17 博客前端: 封装库——浏览器检测	(274)
项目 18 博客前端: 封装库——DOM 加载 [上]	(276)

目 录

项目 19	博客前端：封装库——DOM 加载 [下]	(279)
项目 20	博客前端：封装库——调试封装	(280)
项目 21	博客前端：封装库——动画初探 [上]	(282)
项目 22	博客前端：封装库——动画初探 [中]	(283)
项目 23	博客前端：封装库——动画初探 [下]	(286)
项目 24	博客前端：封装库——透明度渐变	(287)
项目 25	博客前端：封装库——百度分享侧栏	(289)
项目 26	博客前端：封装库——增强弹窗菜单	(292)
项目 27	博客前端：封装库——同步动画	(294)
项目 28	博客前端：封装库——展示菜单	(295)
项目 29	博客前端：封装库——滑动导航	(296)
项目 30	博客前端：封装库——切换	(300)
项目 31	博客前端：封装库——菜单切换	(301)
项目 32	博客前端：封装库——注册验证 [1]	(305)
项目 33	博客前端：封装库——注册验证 [2]	(309)
项目 34	博客前端：封装库——注册验证 [3]	(312)
项目 35	博客前端：封装库——注册验证 [4]	(317)
项目 36	博客前端：封装库——注册验证 [5]	(321)
项目 37	博客前端：封装库——注册验证 [6]	(324)
项目 38	博客前端：封装库——注册验证 [7]	(329)
项目 39	博客前端：封装库——注册验证 [8]	(333)
项目 40	博客前端：封装库——注册验证 [9]	(335)
项目 41	博客前端：封装库——注册验证 [10]	(337)
项目 42	博客前端：封装库——轮播器	(340)
项目 43	博客前端：封装库——延迟加载	(345)
项目 44	博客前端：封装库——预加载	(348)

目 录

项目 45	博客前端：封装库——引入 Ajax	(358)
项目 46	博客前端：封装库——表单序列化	(361)
项目 47	博客前端：封装库——Ajax 注册	(363)
项目 48	博客前端：封装库——Ajax 登录	(369)
项目 49	博客前端：封装库——Ajax 发文	(372)
项目 50	博客前端：封装库——Ajax 换肤	(379)

第1章

JavaScript 概述

学习要点:

1. 什么是 JavaScript
2. JavaScript 的特点
3. JavaScript 的历史
4. JavaScript 的核心
5. 开发工具集

JavaScript 诞生于 1995 年。它诞生的目的是为了完成表单输入的验证。因为在 JavaScript 问世之前,表单的验证都是通过服务器端验证的,而当时还是电话拨号上网的年代,服务器验证数据是一件非常痛苦的事情。

经过多年的发展,JavaScript 从一个简单的输入验证成为一门强大的编程语言。所以,学会使用它是非常简单的,而真正掌握它则需要很漫长的时间。

一、什么是 JavaScript

JavaScript 是一种具有面向对象能力的、解释型的程序设计语言。更具体一点,它是基于对象和事件驱动并具有相对安全性的客户端脚本语言。因为它不需要在一个语言环境下运行,而只需要支持它的浏览器即可。它的主要目的是验证发往服务器端的数据、增加 Web 互动、增强用户体验度等。

二、JavaScript 的特点

1. 松散性

JavaScript 语言核心与 C、C++、Java 相似,比如条件判断、循环、运算符等。但它是一种松散类型的语言,也就是说,它的变量不必具有一个明确的类型。

2. 对象属性

JavaScript 中的对象把属性名映射为任意的属性值。它的这种方式很像哈希表或关

联数组,而不像 C 中的结构体或者 C++、Java 中的对象。

3. 继承机制

JavaScript 中的面向对象继承机制是基于原型的,这和另外一种不太为人所知的 Self 语言很像,而和 C++以及 Java 中的继承大不相同。

三、JavaScript 的历史

1. 引子

大概在 1992 年,有一家公司 Nombas 开发一种叫做 C--(C-minus-minus,简称 Cmm)的嵌入式脚本语言。后因开发者觉得名字比较晦气,最终改名为 ScriptEase。而这种可以嵌入网页中的脚本的理念将成为因特网的一块重要基石。

2. 诞生

1995 年,当时工作在 Netscape(网景)公司的布兰登(Brendan Eich)为解决类似于“向服务器提交数据之前验证”的问题,在 Netscape Navigator 2.0 与 Sun 公司联手开发一个称为 LiveScript 的脚本语言。为了营销便利,之后更名为 JavaScript(目的是在 Java 这棵大树下好乘凉)。

3. 邪恶的后来者

因为 JavaScript 1.0 如此成功,所以微软也决定进军浏览器,发布了 IE 3.0 并搭载了一个 JavaScript 的克隆版,叫做 JScript(这样命名是为了避免与 Netscape 潜在的许可纠纷),并且也提供了自己的 VBScript。

4. 标准的重要

在微软进入后,有三种不同的 JavaScript 版本同时存在:Netscape Navigator 3.0 中的 JavaScript、IE 中的 JScript 以及 CEnv 中的 ScriptEase。与 C 和其他编程语言不同的是,JavaScript 并没有一个标准来统一其语法或特性,而这三种不同的版本恰恰突出了这个问题。随着业界担心的增加,这个语言标准化显然已经势在必行。

5. ECMA

1997 年,JavaScript 1.1 作为一个草案提交给欧洲计算机制造商协会(ECMA)。第 39 技术委员会(TC39)被委派来“标准化一个通用、跨平台、中立于厂商的脚本语言的语法和语义”(http://www.ecma-international.org/memento/TC39.htm)。由来自 Netscape、Sun、微软、Borland 和其他一些对脚本编程感兴趣的公司的程序员组成的 TC39 锤炼出了 ECMA-262,该标准定义了叫做 ECMAScript 的全新脚本语言。

6. 灵敏的微软、迟钝的网景

虽然网景开发了 JavaScript 并首先提交给 ECMA 标准化,但因计划改写整个浏览器引擎的缘故,网景晚了整整一年才推出“完全遵循 ECMA 规范”的 JavaScript1.3。而微软

早在一年前就推出了“完全遵循 ECMA 规范”的 IE4.0。这导致一个直接恶果:JScript 成为 JavaScript 语言的事实标准。

7. 标准的发展

在接下来的几年里,国际标准化组织及国际电工委员会(ISO/IEC)也采纳 ECMAScript 作为标准(ISO/IEC-16262)。从此,Web 浏览器就开始努力(虽然有着不同程度的成功和失败)将 ECMAScript 作为 JavaScript 实现的基础。

8. 山寨打败原创

JScript 成为 JavaScript 语言的事实标准,加上 Windows 绑定着 IE 浏览器,几乎占据全部市场份额,因此,1999 年之后,所有的网页都是基于 JScript 来开发的。而 JavaScript1.x 变成可怜的兼容者。

9. 网景的没落与火狐的崛起

网景在微软强大的攻势下,1998 年全面溃败。但是,星星之火可以燎原,同年成立 Mozilla 项目中 Firefox(火狐浏览器)在支持 JavaScript 方面无可比拟,在后来的时间里一步步蚕食 IE 的市场,成为全球第二大浏览器。

10. 谷歌的野心

Google Chrome,又称 Google 浏览器,是一个由 Google(谷歌)公司开发的开放原始码网页浏览器。它以简洁的页面,极速的浏览,一举成为全球第三大浏览器。随着移动互联网的普及,嵌有 Android 系统的平板电脑和智能手机,在浏览器这块将大有作为。

11. 苹果的战略

Safari 浏览器是苹果公司各种产品的默认浏览器,在苹果的一体机(iMac)、笔记本(Mac)、MP4(ipod)、iPhone(智能手机)、iPad(平板电脑),以及在 Windows 和 Linux 平台都有相应版本。目前市场份额全球第四,但随着苹果的产品不断深入人心,具有称霸之势。

12. 幸存者

Opera 的市场份额在全球排名第五位,占 2%左右。它的背后没有财力雄厚的大公司,但它在“浏览器大战”存活下来的,有着非常大的潜力。

四、JavaScript 的核心

虽然 JavaScript 和 ECMAScript 通常被人们用来表达相同的含义,但 JavaScript 的含义却比 ECMA-262 中规定的要多得多。一个完整的 JavaScript 应该由三个不同的部分组成:① 核心(ECMAScript);② 文档对象模型(DOM);③ 浏览器对象模型(BOM)。

1. ECMAScript 介绍

由 ECMAScript-262 定义的 ECMAScript 与 Web 浏览器没有依赖关系。ECMAScript 定义的只是这门语言的基础,而在此基础之上可以构建更完善的脚本语言。我们常见的 Web 浏览器只是 ECMAScript 实现的可能宿主环境之一。

既然它不依赖于 Web 浏览器,那么它还在哪些环境中寄宿呢? 比如 ActionScript、ScriptEase 等。而它的组成部分有语法、类型、语句、关键字、保留字、操作符、对象等。

2. ECMAScript 版本

ECMAScript 目前有五个版本,这里不再进行详细探讨。有兴趣的同学,可以搜索查阅。

3. Web 浏览器对 ECMAScript 的支持

到了 2008 年,五大主流浏览器 (IE、Firefox、Safari、Chrome、Opera) 全部做到了与 ECMA-262 兼容。其中,只有 Firefox 力求做到与该标准的第 4 版兼容。以下是支持表。

浏览器	ECMAScript 兼容性
Netscape Navigator 2	—
Netscape Navigator 3	—
Netscape Navigator 4 - 4.05	—
Netscape Navigator 4.06 - 4.79	第 1 版
Netscape 6+ (Mozilla 0.6.0+)	第 3 版
Internet Explorer 3	—
Internet Explorer 4	—
Internet Explorer 5	第 1 版
Internet Explorer 5.5 - 7	第 3 版
Internet Explorer 8	第 3.1 版 (不完全兼容)
Internet Explorer 9	第 5 版
Opera 6 - 7.1	第 2 版
Opera 7.2+	第 3 版
Opera 11+	第 5 版
Safari 3+	第 3 版
Firefox 1--2	第 3 版
Firefox 3/4/5/6/7/8/9	第 3/5 版

4. 文档对象模型 (DOM)

文档对象模型 (Document Object Model, DOM) 是针对 XML 但经过扩展用于 HTML 的应用程序编程接口 (Application Programming Interface, API)。

DOM 有三个级别,每个级别都会新增很多内容模块和标准 (有兴趣可以搜索查询)。以下是主流浏览器对 DOM 支持的情况:

浏览器	DOM 兼容性
Netscape Navigator 1 - 4.x	—
Netscape Navigator 6+(Mozilla 0. 6. 0+)	1 级、2 级(几乎全部)、3 级(部分)
Internet Explorer 2 - 4.x	—
Internet Explorer 5	1 级(最小限度)
Internet Explorer 5. 5 - 7	1 级(几乎全部)
Opera 1 - 6	—
Opera 7 - 8.x	1 级(几乎全部)、2 级(部分)
Opera 9+	1 级、2 级(几乎全部)、3 级(部分)
Safari 1. 0x	1 级
Safari 2+	1 级、2 级(部分)
Chrome 0. 2+	1 级、2 级(部分)
Firefox 1+	1 级、2 级(几乎全部)、3 级(部分)

5. 浏览器对象模型(BOM)

访问和操作浏览器窗口的浏览器对象模型(Browser Object Model,BOM)。开发人员使用 BOM 可以控制浏览器显示页面以外的部分。而 BOM 真正与众不同的地方(也是经常会导致问题的地方),还是它作为 JavaScript 实现的一部分,至今仍没有相关的标准。

6. JavaScript 版本

身为 Netscape“继承人”的 Mozilla 公司,是目前唯一沿用最初的 JavaScript 版本编号的浏览器开发商。在网景把 JavaScript 转手给 Mozilla 项目的时候,JavaScript 在浏览器中最后的版本号是 1. 3。后来,随着 Mozilla 的继续开发,JavaScript 版本号逐步递增,如下表所示:

浏览器	JavaScript 版本
Netscape Navigator 2	1. 0
Netscape Navigator 3	1. 1
Netscape Navigator 4	1. 2
Netscape Navigator 4. 06	1. 3
Netscape 6+ (Mozilla 0. 6. 0+)	1. 5
Firefox 1	1. 5
Firefox 1. 5	1. 6
Firefox 2	1. 7
Firefox 3	1. 8
Firefox 3. 1+	1. 9

五、开发工具集

代码编辑器:Notepad++(在 360 软件管家里可以找到,直接下载安装即可)。

浏览器:谷歌浏览器、火狐浏览器、IE 浏览器、IETest 工具等。

PS:学习 JavaScript 需要一定的基础,必须有 xhtml+css 基础、至少一门服务器端编程语言的基础(比如 PHP)、一门面向对象技术(比如 Java)、至少有一个 Web 开发的项目基础(比如留言板程序等)。

第2章

使用 JavaScript

学习要点:

1. 创建一张 HTML 页面
2. <Script>标签解析
3. JS 代码嵌入的一些问题

一、创建一张 HTML 页面

虽然现在很多教材开始使用 html5 来讲解 JavaScript 课程。但我认为这样可能比较超前,对于 JavaScript 初学者,我们还是用比较普及和稳定的 xhtml1.x 来创建一张页面。

很多时候,你无法记住 xhtml1.x 过渡性的标准格式。这个时候,建议打开 Dreamweaver 来获取。页面创建好后,编写一个最简单的 JavaScript 脚本(简称 JS 脚本)。

注意网页的编码格式及文件存储的编码。

二、<Script>标签解析

<script>xxx</script>这组标签,是用于在 html 页面中插入 js 的主要方法。它主要有以下几个属性:

(1) charset: 可选。表示通过 src 属性指定的字符集。由于大多数浏览器忽略它,所以很少有人用它。

(2) defer: 可选。表示脚本可以延迟到文档完全被解析和显示之后再执行。由于大多数浏览器不支持,故很少用。

(3) language: 已废弃。原来用于代码使用的脚本语言。由于大多数浏览器忽略它,所以不要用了。

(4) src: 可选。表示包含要执行代码的外部文件。

(5) type: 必需。可以看作 language 的替代品。表示代码使用的脚本语言的内容类型。范例: type = "text/javascript"。

```
<script type="text/javascript">
    alert('欢迎来到 JavaScript 世界!');
</script>
```

三、JS 代码嵌入的一些问题

如果你想弹出一个</script>标签的字符串,那么浏览器会误解成 JS 代码已经结束了。解决的方法,就是把字符串分成两个部分,通过连接符‘+’来连接。

```
<script type="text/javascript">
    alert('</scr '+'ipt>');
</script>
```

一般来说,JS 代码越来越庞大的时候,我们最好把它另存为一个.js 文件,通过 src 引入即可。它还具有维护性高、可缓存(加载一次,无需加载)、方便未来扩展的特点。

```
<script type="text/javascript" src="demo1.js"></script>
```

这样标签内就没有任何 JS 代码了。但要注意的是,虽然没有任何代码,也不能用单标签:

```
<script type="text/javascript" src="demo1.js" />;
```

也不能在里面添加任何代码:

```
<script type="text/javascript" src="demo1.js">alert('我很可怜,执行不到!')</script>
```

按照常规,我们会把<script>标签存放到<head>...</head>之间。但有时也会放在 body 之间。

不再需要提供注释,以前为了让不支持 JavaScript 浏览器能够屏蔽掉<script>内部的代码,我们习惯在代码的前后用 html 注释掉,现在已经不需要了。

```
<script type="text/javascript">
    <!--
        alert('欢迎!');
    -->
</script>
```

平稳退化不支持 JavaScript 处理:<noscript>

```
<noscript>
    您没有启用 JavaScript
</noscript>
```

第3章

语法、关键保留字及变量

学习要点:

1. 语法构成
2. 关键字和保留字
3. 变量

任何语言的核心都必然会描述这门语言最基本的工作原理。而 JavaScript 的语言核心就是 ECMAScript,而目前使用最普遍的是第3版,我们就主要以这个版本来讲解。

一、语法构成

1. 区分大小写

ECMAScript 中的一切,包括变量、函数名和操作符都是要区分大小写的。例如: `text` 和 `Text` 表示两种不同的变量。

2. 标识符

所谓标识符,就是指变量、函数、属性的名字,或者函数的参数。标识符可以是下列格式规则组合起来的一个或多个字符:

- (1) 第一字符必须是一个字母、下划线(`_`)或一个美元符号(`$`)。
- (2) 其他字符可以是字母、下划线、美元符号或数字。
- (3) 不能把关键字、保留字、`true`、`false` 和 `null` 作为标识符。

例如: `myName`、`book123` 等。

3. 注释

ECMAScript 使用 C 风格的注释,包括单行注释和块级注释。

// 单行注释

/*

* 这是一个多行

* 注释