

新一代信息技术系列教材

基于新信息技术的 JavaScript

程序设计基础

主编/谢钟扬 屈 薇 刘 群
主审/符开耀 马 庆



西安电子科技大学出版社
<http://www.xduph.com>

CONTROL

新一代信息技术系列教材

基于新信息技术的 JavaScript 程序设计基础

主 编 谢钟扬 屈 薇 刘 群

副主编 张 珏 胡宇晴 彭 玲 李 建

主 审 符开耀 马 庆

西安电子科技大学出版社

内 容 简 介

本书主要介绍了在新信息技术条件下 JavaScript 程序设计的基础知识。全书共分为 11 章。第 1 至第 8 章为 JavaScript 程序语言的基础语法, 主要介绍 JavaScript 语言 and 核心语法以及基础应用; 第 9 至第 10 章为 JavaScript 程序的对象模型, 主要介绍 JavaScript 程序的浏览器对象模型 BOM 和文档对象模型 DOM 的基础知识; 第 11 章为 AJAX 技术, 主要介绍在异步状态下, JavaScript 程序与服务器通信的相关技术。

本书可作为大专院校软件技术、软件开发等专业的教学参考书, 也可供其他相关专业人员和技术人员参考。

图书在版编目(CIP)数据

基于新信息技术的 JavaScript 程序设计基础 / 谢钟扬, 屈薇, 刘群主编. —西安:
西安电子科技大学出版社, 2018.8(2018.9 重印)

ISBN 978-7-5606-5034-0

I. ① 基… II. ① 谢… ② 屈… ③ 刘… III. ① JAVA 语言—程序设计 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2018)第 174433 号

策划编辑 杨丕勇

责任编辑 祝婷婷 杨丕勇

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 北京虎彩文化传播有限公司

版 次 2018 年 8 月第 1 版 2018 年 9 月第 2 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 6.5

字 数 121 千字

定 价 20.00 元

ISBN 978-7-5606-5034-0/TP

XDUP 5336001-2

如有印装问题可调换

前 言

目前，互联网技术已融入到社会生活的方方面面，各种互联网应用所涉及的功能、需求日益复杂，面向的用户群体也日益广泛，对 Web 前端的复杂程度、用户友好性提出了更高的要求；移动互联网的发展，也使 Web 前端开发需要兼顾移动设备的浏览以及移动设备的使用习惯和功能要求，这些都对 Web 前端开发形成了巨大的挑战。在当前的 Web 前端技术大发展的浪潮中，JavaScript 语言扮演着核心角色。在整个前端技术中，JavaScript 负责在客户端串联起 HTML5、CSS3，动态地向用户展现复杂的、绚烂的用户界面，同时还可以完成一定的业务功能和运算。JavaScript 是整个 Web 前端领域的重要基础和核心内容。

本书综合了作者这些年来积累的 JavaScript 开发经验，详细介绍了 JavaScript 程序设计开发所需的核心知识和实用的解决方案，力图用简明扼要的语言、翔实具体的实例，让读者从原理上理解和掌握 JavaScript 程序开发所需的技术。

本书以 JavaScript 公开发布的技术标准和文档为基础，结合实际开发和教育教学过程中的实践积累编写而成，主要针对的是 JavaScript 零基础的读者或计算机、软件技术、软件开发类专业的大专、本科学生。

本书以基础知识为框架，结合具体代码应用示例，旨在让读者循序渐进地掌握 JavaScript 编程的基础知识，并能具备一定的应用、开发能力。本书第 1 章至第 8 章由屈薇编写，第 9 章至第 11 章由谢钟扬编写，刘群完成各章节习题的编写和整理工作，其他各位副主编分别贡献了相关代码、实例以及知识点的解释说明。

由于作者的水平有限，书中不妥之处在所难免，恳请读者批评指正。

编 者

2018 年 6 月

目 录

第 1 章 JavaScript 概述.....	1
1.1 JavaScript 性质	1
1.2 JavaScript 使用场景	1
1.3 JavaScript 的未来	3
课后习题	4
第 2 章 JavaScript 核心语法	5
2.1 词法结构	5
2.2 字符集	5
2.2.1 区分大小写	5
2.2.2 空格、换行符和格式控制符	6
2.2.3 Unicode 转义序列	6
2.2.4 标准化	6
2.3 注释	7
2.4 标识符和保留字	7
2.5 变量	8
2.6 HTML 中的引用	8
课后习题	9
第 3 章 JavaScript 的数据类型.....	10
3.1 字符串	10
3.1.1 字符串访问	10
3.1.2 字符串长度	10
3.1.3 特殊字符	11
3.1.4 字符串属性和方法	11
3.2 数字	12
3.2.1 整型直接量	13
3.2.2 浮点型直接量	13
3.3 布尔值	13

3.4 数组	14
3.5 对象	14
3.6 null	15
3.7 undefined	15
课后习题	15
第 4 章 表达式和运算符	16
4.1 原始表达式	16
4.2 属性访问表达式	17
4.3 调用表达式	17
4.4 算术运算符	18
4.4.1 “+”运算符	18
4.4.2 一元运算符	19
4.4.3 位运算符	20
4.5 关系运算符	20
4.5.1 相等和不等运算符	21
4.5.2 比较运算符	22
4.5.3 in 运算符	23
4.5.4 instanceof 运算符	23
4.6 逻辑表达式	23
4.6.1 逻辑与(&&)	24
4.6.2 逻辑或()	25
4.6.3 逻辑非(!)	26
4.7 赋值表达式	26
课后习题	27
第 5 章 流程控制	28
5.1 条件语句	28
5.1.1 if 语句	28
5.1.2 if...else 语句	29
5.1.3 if...else if...else 语句	30
5.2 分支语句	30
5.3 循环语句	32

5.3.1 while 语句.....	32
5.2.2 do...while 语句.....	33
5.2.3 for 语句.....	34
5.4 跳转语句.....	35
5.4.1 break 语句.....	35
5.4.2 continue 语句.....	36
5.4.3 return 语句.....	36
5.4.4 throw 语句.....	37
5.4.5 try...catch...finally 语句.....	37
课后习题.....	39
 第 6 章 对象.....	 41
6.1 创建对象.....	42
6.2 对象的属性.....	43
6.3 对象的方法.....	43
6.3.1 toString()方法.....	43
6.3.2 toLocaleString()方法.....	44
6.3.3 valueOf()方法.....	44
6.4 对象序列化.....	44
课后习题.....	44
 第 7 章 数组.....	 46
7.1 创建数组.....	46
7.2 数组的读和写.....	47
7.3 数组的长度.....	48
7.4 数组元素的添加和删除.....	49
7.5 数组遍历.....	50
7.6 多维数组.....	51
7.7 数组方法.....	52
7.7.1 join().....	52
7.7.2 reverse().....	52
7.7.3 sort().....	52
7.7.4 concat().....	53

7.7.5	slice()	54
7.7.6	splice()	54
7.7.7	push()和 pop()	55
7.7.8	shift()和 unshift()	55
7.7.9	toString()	56
7.7.10	indexOf()和 lastIndexOf()	56
7.7.11	forEach()	57
7.7.12	map()	57
7.7.13	filter()	57
7.7.14	some()	58
7.7.15	reduce()和 reduceRight()	58
	课后习题	59
第 8 章	函数	60
8.1	函数的概念	60
8.2	定义函数	60
8.3	调用函数	61
8.4	变量的作用域	61
8.5	方法	64
8.6	箭头函数	65
	课后习题	66
第 9 章	浏览器对象模型 BOM	67
9.1	什么是 BOM	67
9.2	window 对象	67
9.3	screen 对象	69
9.4	location 对象	70
9.5	history 对象	72
9.6	navigator 对象	73
第 10 章	文档对象模型 DOM	75
10.1	HTML DOM 构建	75
10.2	元素的查找	77

10.2.1	getElementById() 方法.....	78
10.2.2	getElementsByTagName()方法.....	78
10.2.3	getElementsByClassName()方法.....	78
10.3	节点对象的属性.....	78
10.4	节点对象的操作.....	79
10.4.1	改变 HTML 内容.....	79
10.4.2	改变 CSS 样式.....	80
10.4.3	添加新的 HTML 元素.....	80
10.4.4	删除已有的 HTML 元素.....	81
10.5	DOM 事件.....	82
第 11 章	AJAX 技术.....	84
11.1	AJAX 简介.....	84
11.2	XHR 对象.....	84
11.2.1	创建.....	84
11.2.2	请求.....	85
11.2.3	响应.....	86
11.3	JSON.....	87
11.3.1	JSON 对象语法.....	87
11.3.2	JSON 数组语法.....	88
	课后习题.....	88
	课后习题参考答案.....	89
	参考文献.....	94

第1章 JavaScript 概述

1.1 JavaScript 性质

JavaScript 是一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。JavaScript 的解释器被称为 JavaScript 引擎，最早在 HTML(标准通用标记语言下的一个应用)网页上使用，现已广泛应用于客户端的脚本语言。

JavaScript 是属于网络的脚本语言，在 Web 应用开发中被广泛用于为网页添加各式各样的动态功能，为用户提供更流畅美观的浏览效果。通常 JavaScript 脚本是通过嵌入在 HTML 中来实现自身功能的。

- (1) JavaScript 是一种解释性脚本语言(代码不进行预编译)。
- (2) JavaScript 主要用来向 HTML 页面添加交互行为。
- (3) JavaScript 可以直接嵌入 HTML 页面，但写成单独的 js 文件利于结构和行为的分离。
- (4) JavaScript 在绝大多数浏览器的支持下，可以在多种平台下运行，如 Windows、Linux、MAC、Android、iOS 等。

JavaScript 脚本语言同其他语言一样，有它自身的基本数据类型、表达式和算术运算符及程序的基本程序框架。JavaScript 提供了四种基本的数据类型和两种特殊的数据类型(分别用来处理数据和文字)，而变量提供存放信息的地方，表达式则可以完成较复杂的信息处理。

1.2 JavaScript 使用场景

JavaScript 脚本语言由于其效率高、功能强大等特点，在表单数据合法性验证、网页特效、交互式菜单、动态页面、数值计算等方面获得了广泛的应用，甚至出现了完全使用 JavaScript 编写的基于 Web 浏览器的 Unix 操作系统 JS/UIX，以及无需安装即可使用的中文输入法程序 JustInput。可见，JavaScript 脚本的编程能力不容小觑。

1. 表单数据合法性验证

使用 JavaScript 脚本语言能有效验证客户端提交的表单上数据的合法性, 若数据合法则进行下一步操作, 否则返回错误提示信息, 如图 1.1 所示。

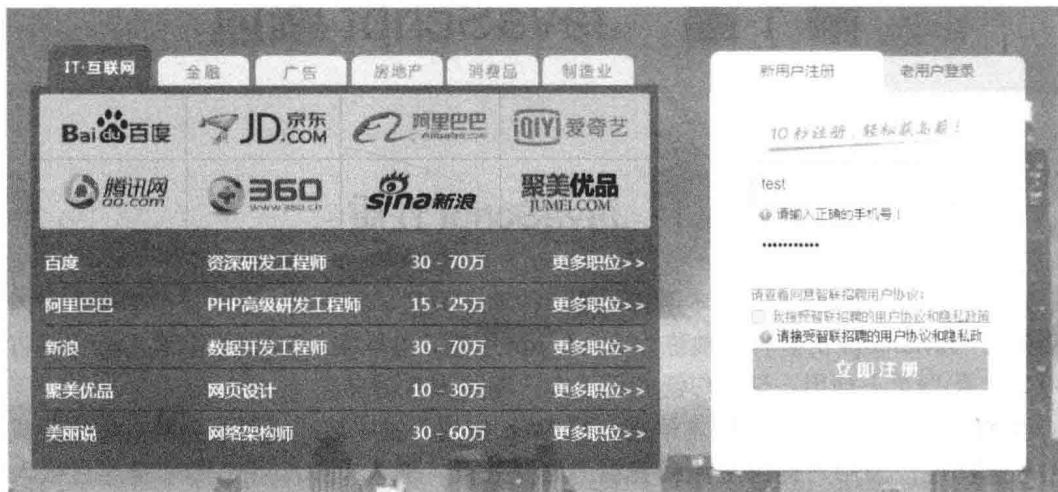


图 1.1 JavaScript 用于表单校验

2. 网页特效

使用 JavaScript 脚本语言, 结合 DOM 和 CSS 能创建绚丽多彩的网页特效, 如各种闪烁文字、滚动的广告图片、页面轮换效果等, 如图 1.2 所示。



图 1.2 网页特效

3. 交互式菜单

使用 JavaScript 脚本语言可以创建具有动态效果的交互式菜单,它完全可以与 Flash 制作的页面导航菜单相媲美,如图 1.3 所示。



图 1.3 交互式菜单

1.3 JavaScript 的未来

尽管每年都面临各种质疑和批评,但 JavaScript 仍然是世界上发展最快的编程语言之一。除此之外,根据 really.com 网站收集的数据,从雇主的角度来看,JavaScript 是前三名最受欢迎的语言之一。根据 StackOverflow 在 2017 年所做的调查显示,JavaScript 是最常用的语言,如图 1.4 所示。

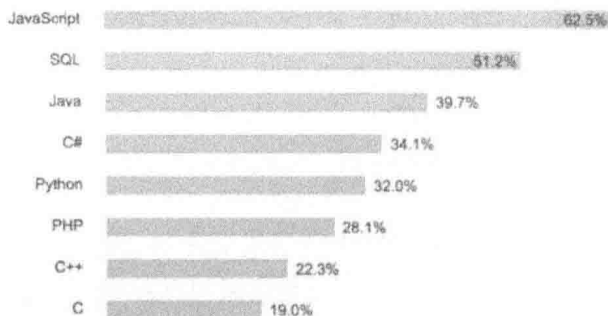


图 1.4 StackOverflow 编程语言排名

根据这些数据,可以看出 JavaScript 不太可能在短期内消失。在开发交互式网页时,它仍然是最好的选择之一,而且它是所有主要浏览器都支持的编程语言。

另一个重要的细节是,JavaScript 已经从一个可以将一些交互性带入网页的工具,发展到一个可以进行高效服务器端开发的工具。Node.js 是一个开放源码的运行环境,允许使用 JavaScript 创建服务器端代码。几十个基于 Node.js 的框架(如 Meteor 和 Derby)使这种技术适用于几乎任何类型的项目,并提供构建高度可扩展的 Web 应用程序所需的功能。

课后习题

一、选择题

1. JavaScript 是_____。

- A. 客户端脚本语言
- B. 客户端标记语言
- C. 服务器端脚本语言
- D. 服务器端标记语言

2. 关于 JavaScript 的作用,不正确的是_____。

- A. 访问数据库
- B. 制作网页特效
- C. 实现客户端表单验证
- D. 实现网页交互操作

二、简答题

1. 简述 JavaScript 的作用。

2. 完成 JavaScript 的入门程序——“hello world”的编写。

第2章 JavaScript 核心语法

2.1 词法结构

JavaScript 是一门简单的语言，也是一门复杂的语言。说它简单，是因为入门简单，学会使用它只需片刻功夫；而说它复杂，是因为要做到灵活运用、真正掌握，则需要数年时间。实际上，前端工程师很大程度上就是指 JavaScript 工程师。前端入门容易精通难，说的是前端，更指的是 JavaScript。

JavaScript 的词法结构是一套基础性规则，是一种标准，用来描述如何使用 JavaScript 来编写程序。作为词法的基础，它规定了诸如变量名是什么样的，怎么写注释，以及程序语句之间如何分隔等。

2.2 字符集

JavaScript 程序是用 Unicode 字符集编写的。Unicode 是 ASCII 和 Latin-1 的超集，并支持几乎所有在用的编程语言。ECMAScript 3 要求 JavaScript 的实现必须支持 Unicode 2.1 及后续版本，ECMAScript 5 则要求支持 Unicode 3 及后续版本。

2.2.1 区分大小写

JavaScript 是区分大小写的语言，也就是说，关键字、变量、函数名和所有的表示符 (identifier) 都必须采取一致的大小写形式。比如，关键字 “for” 必须写成 “for”，而不能写成 “For” 或者 “FOR”。同样，“username”、“Username”、“userName”、“USERNAME” 是四个不同的变量名。

但需要注意的是，HTML 并不区分大小写(尽管 XHTML 区分大小写)。由于它和客户端 JavaScript 联系紧密，因此这点区别很容易混淆。许多的客户端 JavaScript 对象和属性与它们所表示的 HTML 标签和属性同名。在 HTML 中，这些标签和属性名可以是大写也可以是小写，而在 JavaScript 中则必须是小写。例如，在 HTML 中设置事件处理程序时，onclick

属性可以写成 `onClick`，但在 JavaScript 代码(或者 XHTML 文档)中，必须使用小写的 `onclick`。

2.2.2 空格、换行符和格式控制符

JavaScript 会忽略程序中标识(token)之间的空格。多数情况下，JavaScript 同样会忽略换行符。由于可以在代码中随意使用空格和换行，因此可以采用整齐、一致的缩进来形成统一的编码风格，从而提高代码的可读性。

除了可以识别普通的空格符(`\u0020`)，JavaScript 还可以识别如下这些表示空格的字符：水平制表符(`\u0009`)、垂直制表符(`\u000B`)、换页符(`\u000C`)、不中断空白(`\u00A0`)、字节序标记格式控制符(`\uFEFF`)，以及在 Unicode 中所有 Zs 类别的字符。JavaScript 将如下字符识别为行结束符：换行符(`\u000A`)、回车符(`\u000D`)、行分隔符(`\u2028`)、段分隔符(`\u2029`)。回车符加换行符在一起被解析为一个单行结束符。

Unicode 格式控制字符(Cf 类)，比如“从右至左书写标记”(`\u200F`)和“从左至右书写标记”(`\u200E`)，控制着文本的视觉显示，这对于一些非英语文本的正确显示来说是至关重要的，这些字符可以用在 JavaScript 的注释、字符串直接量和正则表达式直接量中，但不能用在标识符(比如变量名)中。但有个例外，零宽连接符(`\u200D`)和零宽非连接符(`\uFEFF`)是可以出现在标识符中的，但不能作为标识符的首字符。上面也提到了，字节序标记格式控制符(`\uFEFF`)被当成了空格来对待。

2.2.3 Unicode 转义序列

JavaScript 定义了一种特殊序列，使用 6 位 ASCII 字符代表任意 Unicode 内码。这些 Unicode 转义序列均以 `\u` 为前缀，其后跟随 4 位十六进制数。转义字符可以用作字符串直接量、正则表达式直接量和标识符(关键字除外)。

例如：“`café`”和“`ca\u00e9`”两个字符串常量是等价的。

Unicode 转义写法也可以出现在注释中，但由于 JavaScript 会将注释忽略，因此它们只是被当成上下文中的 ASCII 字符处理，而不会被解析为其对应的 Unicode 字符。

2.2.4 标准化

Unicode 允许使用多种方法对同一个字符进行编码。比如，字符“é”可以使用 Unicode 字符 `\u00E9` 表示，也可以使用普通的 ASCII 字符 `e` 跟随一个语调符 `\u0301` 来表示。在文本编辑器中，这两种编码的显示结果是一模一样的，但它们的二进制编码表示是不一样的，在计算机里也不相等。Unicode 标准为所有字符定义了一个首选的编码格式，并给出了一个标准化的处理方式将文本转换为一种合适比较的标准格式，JavaScript 会认为它正在解析的程序代码已经是这种标准格式，不会对其标识符、字符串或正则表达式做标准化处理。

2.3 注 释

JavaScript 支持两种格式的注释。在行尾“//”之后的文本都会被 JavaScript 当做注释忽略掉。此外，“/*”和“*/”之间的文本也会被当做注释，这种注释可以跨行书写，但不能有嵌套的注释。下面都是合法的 JavaScript 注释：

```
//这里是单行注释
```

```
/*这里是一段注释*/ //这里是另一段注释
```

```
/*  
*这又是一段注释  
*这里可以注释多行  
*/
```

2.4 标识符和保留字

任何一种计算机语言都离不开标识符和保留字，下面将详细介绍 JavaScript 的标识符和保留字。

标识符用来对变量和函数名进行命名，或者用做某些循环语句中跳转位置的标记。标识符必须以字母、下划线(_)、或美元符(\$)开始，后续的字符可以是字母、数字、下划线或美元符。注意 JavaScript 允许标识符中出现 Unicode 字符全集中的字母和数字，也就是说可以用非英语语言或数字符号来书写标识符。例如：identifier、userName、User_Name、_sys_val、身高、\$change 等为合法的标识符，而 2mail、room#、class 为非法的标识符。

保留字是语言中定义具有特殊含义的标识符，保留字不能作为标识符使用。JavaScript 语言中定义了一些具有专门意义和用途的保留字，这些保留字称为关键字。下面列出了 JavaScript 语言中的关键字：break、delete、function、return、typeof、case、do、if、switch、var、catch、else、in、this、void、continue、false、instanceof、throw、while、debugger、finally、new、true、const、with、default、for、null 和 try。还有一些保留字是在未来 JavaScript 版本使用的，它们主要有：class、enum、export、extends、import 和 super。

2.5 变 量

变量是存储信息的容器，是一个值的符号名称，可以通过名称来获得对值的引用。比如，JavaScript 变量可用于存放值(比如 $x=2$)和表达式(比如 $z=x+y$)。变量可以使用短名称(比如 x 和 y)，也可以使用描述性更好的名称(比如 age 、 sum 、 $totalvolume$)。

- (1) 变量必须以字母开头。
- (2) 变量也能以 $\$$ 和 $_$ 符号开头(不过不推荐这么做)。
- (3) 变量名称对大小写敏感(y 和 Y 是不同的变量)。

注意：JavaScript 语句和 JavaScript 变量都对大小写敏感。

2.6 HTML 中的引用

HTML 中引用 JavaScript 主要有三种方式。

第一种方式：外部引用远程 JavaScript 文件，有以下两种方法。

- (1) 相对路径：

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

- (2) 绝对路径：

```
<script type="text/javascript" src="http://common.blogs.com/js/test.js"></script>
```

第二种方式：直接写在页面上。例如：

```
<script type="text/javascript"><br/>var x=0<br/>function fn(args){<br/>//...<br/>}<br/>//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="46 761 921 803" data-label="Text"><p>加上“<code><![CDATA[</code>”和“<code>]]&gt;</code>”是为了兼容 XHTML，它是推荐的写法。而在 HTML 时代一般用“<code>&lt;!--</code>”和“<code>//--&gt;</code>”。</p></div><div data-bbox="90 810 555 828" data-label="Text"><p>第三种方式：在一些 HTML 控件中使用。例如：</p></div><div data-bbox="129 837 643 878" data-label="Text"><pre>&lt;body onload="alert('loaded');"&gt;<br/>input type="text" name="username" onclick = "alert(this.value);"/&gt;</pre></div><div data-bbox="56 968 778 992" data-label="Page-Footer"><p>此为试读，需要完整PDF请访问：<a href="http://www.ertongbook.com">www.ertongbook.com</a></p></div>
```