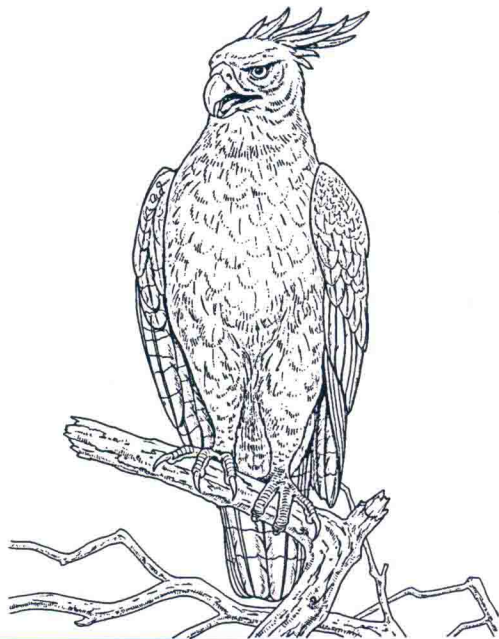




工业和信息化“十三五”  
人才培养规划教材



# JavaScript

前端开发 | 程序设计教程 **微课版**

JavaScript Front-end Development

李玉臣 臧金梅 © 主编

张晓诺 贾殿燕 © 副主编



二维码扫一扫，同步观看**教学视频**，重点难点**轻松掌握**

**86** 个同步案例，**10** 个综合案例，真正做到**学以致用**

提供配套教学课件、电子教案、教学大纲、源代码、自测题库、习题答案、实训指导



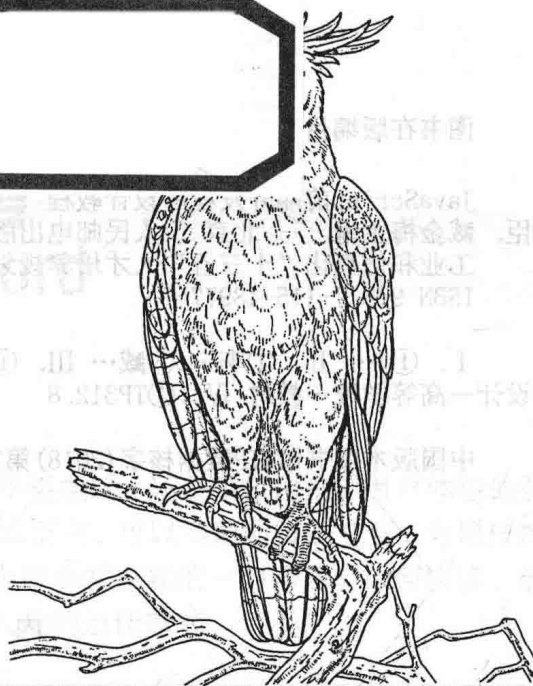
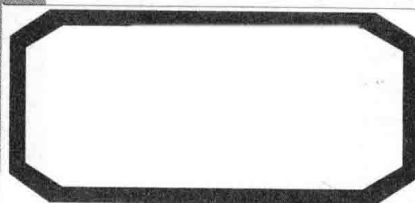
中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



工业和信息化“十三五”  
人才培养规划教材



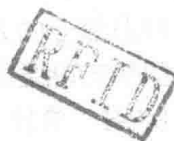
# JavaScript

前端开发 | 程序设计教程 **微课版**

JavaScript Front-end Development

李玉臣 臧金梅 ◎ 主编

张晓诺 贾殿燕 ◎ 副主编



人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

JavaScript前端开发程序设计教程：微课版 / 李玉臣，臧金梅主编. — 北京：人民邮电出版社，2018.10（2019.6重印）  
工业和信息化“十三五”人才培养规划教材  
ISBN 978-7-115-48977-7

I. ①J… II. ①李… ②臧… III. ①JAVA语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第169657号

## 内 容 提 要

JavaScript 是一门广泛应用于 Web 前端开发的脚本语言，能为网页添加各式各样的动态效果，为用户提供流畅美观的浏览效果，具有简单、易学的特点。

本书系统地讲解了 JavaScript 编程的基础知识和实用技术。全书共 10 章，内容包括初识 JavaScript、JavaScript 语言基础、流程控制语句、JavaScript 函数、对象、BOM 和 DOM、事件及事件处理、表单和表单对象、Ajax 初步、JavaScript 特效。为了加深读者对知识的理解，本书在全面、系统地知识讲解的基础上，配备了精彩的同步案例及综合案例，有助于读者对知识和技能的理解。

本书既可以作为普通高等学校、高职高专院校计算机相关专业的教材，也可以作为 JavaScript 爱好者及相关技术人员自学的参考资料。

- 
- ◆ 主 编 李玉臣 臧金梅
  - 副 主 编 张晓诺 贾殿燕
  - 责任编辑 马小霞
  - 责任印制 马振武
  
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
天津翔远印刷有限公司印刷
  
  - ◆ 开本：787×1092 1/16  
印张：15.25 2018 年 10 月第 1 版  
字数：376 千字 2019 年 6 月天津第 4 次印刷
- 

定价：45.00 元

读者服务热线：(010)81055256 印装质量热线：(010)81055316  
反盗版热线：(010)81055315  
广告经营许可证：京东工商广登字 20170147 号

# 前言

## Foreword

随着互联网行业的发展，无论是计算机、平板电脑，还是手机用户，用户体验的要求都越来越高，JavaScript 作为一种流行的脚本语言，可以实现网页的交互，为用户提供流畅的网页浏览效果，是目前 Web 前端开发中非常重要的一门语言。如何快速、全面、系统地了解并掌握它，已成为 Web 开发人员的迫切需求。

### 本书特色

#### 1. 案例贯穿

案例教学是当前非常有效的教学方法之一。本书将 JavaScript 脚本知识和案例有机结合，通过先易后难、从基础到高级的方式，循序渐进讲解 JavaScript 的基础知识和应用技术，并且通过“案例贯穿”的形式展开，在讲解知识点时，共配备了 86 个“同步案例”，让读者明白所学的知识点能用来解决什么问题。

然后，全书共用 10 个“综合案例”有效引导读者将本章学过的内容串起来，培养读者分析问题与解决问题的能力，以此真正做到知识点与案例相辅相成，使读者学以致用。

#### 2. 微课视频

为帮助读者对重点知识的理解和掌握，本书每章重点、难点部分都配备微课视频，读者只需用手机扫描二维码，即可同步观看视频教学。

#### 3. 巩固训练

本书在每一章的最后都提供了训练习题，方便读者及时巩固和验证自己的学习效果。

### 内容安排

本书在章节安排上充分考虑到实际的教学和学习过程，使其更加符合人的一般认知规律。从 JavaScript 语言的发展历史及应用范围开始介绍，再通过数据类型和流程控制语句的讲解，让读者掌握 JavaScript 语言基本编程思路，然后对函数、对象、BOM 和 DOM、事件等 JavaScript 的核心知识点进行重点介绍，最后介绍了几种常用的 JavaScript 的网页特效。全书通过大量有针对性的同步案例及综合案例来帮助读者更方便、更清晰地理解教材内容、知识点和技能点。

### 教学建议

本书作为教材使用时，建议采用理论实践一体化教学模式，课堂教学建议为 40~45 学时，上机指导建议为 40~45 学时。各章主要内容和学时建议分配如下，教师可以根据实际教学情况进行调整。

学时分配表

章节	内容	课堂学时	实践学时
第 1 章	初识 JavaScript, 包括 JavaScript 简介、编写 JavaScript 的工具、JavaScript 在 HTML 中的应用	2	2
第 2 章	JavaScript 语言基础, 包括数据类型、变量与常量、运算符与表达式	2	2
第 3 章	流程控制语句, 包括消息对话框、注释、条件分支语句、循环控制语句、continue 和 break 语句	2	2
第 4 章	JavaScript 函数, 包括函数的定义、预定义函数、自定义函数	4	4
第 5 章	对象, 包括面向对象程序设计、内置对象、自定义对象、数组对象	8	8
第 6 章	BOM 和 DOM, 包括认识 BOM、Window 对象、Navigator 对象、Document 对象、DOM 对象、DOM 对象的方法和属性、访问 DOM 对象、操纵 DOM 对象	8	8
第 7 章	事件及事件对象, 包括事件的概念、事件对象、处理鼠标事件、处理键盘事件、处理页面事件、处理文本编辑事件	4	4
第 8 章	表单和表单对象, 包括表单和表单对象介绍、访问表单和表单对象、操作表单对象	4	4
第 9 章	初识 Ajax, 包括 Ajax 概念、Ajax 编程基础、Ajax 应用案例	4	4
第 10 章	JavaScript 特效, 包括页面导航栏特效、页面文本特效、图形特效	4	4
学时总计		42	42

由于编者水平有限, 书中难免有不足之处, 欢迎读者来函给予宝贵意见, 我们将不胜感激。编者电子邮箱: wflyc028@163.com。

编者

2018年6月

# 目录

## Contents

### 第1章 初识 JavaScript

1.1 JavaScript 简介	2	1.3.1 直接将 JavaScript 代码嵌入 HTML 中	5
1.1.1 JavaScript 的发展概况和特点	2	1.3.2 链接外部的 JavaScript	6
1.1.2 JavaScript 的应用	4	1.3.3 直接在 HTML 标签中使用	7
1.1.3 支持 JavaScript 的浏览器	4	1.4 【综合案例 1】: 随机抽奖小程序	7
1.2 编写 JavaScript 的工具	5	【知识点提炼】	8
1.3 JavaScript 在 HTML 中的应用	5	【训练 1】	8

### 第2章 JavaScript 语言基础

2.1 数据类型	10	【同步案例 2-5】: 比较两个数的大小	19
2.1.1 基本数据类型	10	2.3.3 赋值运算符	20
2.1.2 引用数据类型	11	【同步案例 2-6】: 赋值运算符使用示例	21
2.1.3 特殊数据类型	11	2.3.4 逻辑运算符	22
【同步案例 2-1】: 输出 JavaScript 支持的最大数值和最小数值	12	【同步案例 2-7】: 逻辑运算符与关系运算符配合使用示例	22
2.2 变量与常量	13	2.3.5 条件运算符	23
2.2.1 标识符	13	【同步案例 2-8】: 实现选择正确或错误结果的示例	23
2.2.2 关键字	13	2.3.6 其他运算符	24
2.2.3 变量的定义和赋值	14	【同步案例 2-9】: 通过 typeof 测试变量类型示例	24
【同步案例 2-2】: 变量定义和赋值使用示例	15	2.4 表达式	25
2.2.4 变量的作用域	16	2.5 【综合案例 2】: 挑选出随机生成的两个数中的最大数	26
2.2.5 常量	16	【知识点提炼】	27
【同步案例 2-3】: 通过常量 PI 求圆的周长	16	【训练 2】	28
2.3 运算符	17		
2.3.1 算术运算符	18		
【同步案例 2-4】: 计算账单金额	18		
2.3.2 关系运算符	19		

## 第3章 流程控制语句

3.1 消息对话框	31	【同步案例 3-7】：根据成绩给出学生的考评等级	42
3.1.1 警示对话框	31	3.4 循环控制语句	44
【同步案例 3-1】：打开页面，首先弹出一个问候对话框	31	3.4.1 for 循环语句	44
3.1.2 确认对话框	32	【同步案例 3-8】：通过 for 循环求 1 到 100 的累加和	45
【同步案例 3-2】：在对话框中通过单击不同按钮来显示不同的内容	32	3.4.2 while 循环语句	46
3.1.3 提示对话框	33	【同步案例 3-9】：通过 while 循环求 1 到 100 的奇数累加和	46
【同步案例 3-3】：根据提示输入内容并且在页面上显示	33	3.4.3 do...while 循环语句	48
3.2 注释	34	【同步案例 3-10】：通过 do...while 循环求 1 到 100 的偶数累加和	48
3.2.1 单行注释符	34	3.4.4 循环语句嵌套	49
3.2.2 多行注释符	35	【同步案例 3-11】：输出“*”组成的直角三角形图案	50
3.3 条件分支语句	35	3.5 continue 和 break 语句	51
3.3.1 if 语句	35	【同步案例 3-12】：通过 continue 和 break 来控制程序是否继续运行	51
【同步案例 3-4】：通过 if...else 语句判断两个数中的最大值	38	3.6 【综合案例 3】：随机答题的小游戏	52
【同步案例 3-5】：判断今天是星期几	39	【知识点提炼】	54
3.3.2 switch 语句	40	【训练 3】	54
【同步案例 3-6】：通过 switch 语句判断今天是星期几	41		
3.3.3 if 语句嵌套	42		

## 第4章 JavaScript 函数

4.1 什么是函数	58	能够在页面上输出字符串	62
4.2 预定义函数	58	【同步案例 4-4】：定义函数，实现计算商品总价	62
4.2.1 什么是预定义函数	58	4.3.2 自定义函数的调用	63
4.2.2 常用的预定义函数	58	【同步案例 4-5】：通过函数名直接调用函数示例	63
【同步案例 4-1】：isNaN()函数应用示例	59	【同步案例 4-6】：通过超链接调用函数示例	64
【同步案例 4-2】：parseFloat()和parseInt()函数应用示例	60	4.3.3 函数的参数和返回值	65
4.3 自定义函数	61	【同步案例 4-7】：通过函数统计商品的总价	65
4.3.1 自定义函数的创建	61		
【同步案例 4-3】：定义一个函数，			

【同步案例 4-8】：通过函数返回值 求出两个数中最大数	67
4.3.4 函数变量的作用域	68
【同步案例 4-9】：函数的变量作用域 示例	68
4.3.5 函数的嵌套	69
【同步案例 4-10】：求出 $1+(1+2)+$	

$(1+2+3)+\dots+(1+2+\dots+n)$ 的值	69
4.4 【综合案例 4】：编写可以根据输入 数值实现不同层数的三角形“*” 图案的函数	70
【知识点提炼】	71
【训练 4】	72

## 第 5 章 对象

5.1 面向对象程序设计	75
5.1.1 面向对象编程思想	75
5.1.2 对象	75
5.1.3 类	76
5.1.4 JavaScript 对象框架	76
5.2 内置对象	76
5.2.1 Object 对象类	77
5.2.2 Date 对象类	77
【同步案例 5-1】：读取当前的日期并 在页面上显示出来	79
【同步案例 5-2】：计算自己活了 多少天?	79
5.2.3 String 对象类	80
【同步案例 5-3】：设计一个提取字符 串中数字的函数	91
【同步案例 5-4】：输入一个字符串， 然后反向输出到页面上	92
5.2.4 Math 对象	93
【同步案例 5-5】：计算圆的面积	94
【同步案例 5-6】：求圆周率的 4 次方， 并取四舍五入整数显示	95
5.2.5 Number 对象	96
【同步案例 5-7】：输出 JavaScript 能够 处理的数值区间	96
5.3 自定义对象	97
5.3.1 通过 Object 类创建对象	97
【同步案例 5-8】：通过 Object 创建 对象示例	98
5.3.2 通过字面量创建对象	99
【同步案例 5-9】：通过字面量对象	

创建对象示例	99
5.3.3 通过构造函数 (Constructor) 创建对象	100
【同步案例 5-10】：通过构造函数 (Constructor) 创建对象示例	104
5.3.4 通过 Function 对象定义方法	105
【同步案例 5-11】：通过 Function 对象 显式地创建方法示例	106
【同步案例 5-12】：通过 Function 对象 隐式地创建方法示例	107
5.3.5 通过原型对象 (prototype) 定义方法	108
【同步案例 5-13】：通过原型对象 (prototype) 访问共享方法示例	108
5.3.6 通过 for...in 语句访问对象的 属性	110
【同步案例 5-14】：通过 for...in 语句 循环遍历对象的属性	110
5.3.7 with 语句访问对象的属性和 方法	111
【同步案例 5-15】：通过 with 语句循环 遍历对象的属性	111
5.3.8 继承	112
【同步案例 5-16】：子类继承父类的 属性和方法示例	112
5.4 数组对象	114
5.4.1 什么是数组	114
5.4.2 创建数组对象	114
【同步案例 5-17】：创建一个具有 4 个 元素的数组，并为各个元素赋值	115

5.4.3 数组的属性和方法	115
【同步案例 5-18】：对数组中的元素进行排序	116
【同步案例 5-19】：将两个数组连接成一个数组	117
5.4.4 数组的访问	118
【同步案例 5-20】：使用 for...in 语句遍历数组	119



5.5 【综合案例 5】：每天信息提示栏	120
5.6 【综合案例 6】：倒计时广告牌	122
【知识点提炼】	124
【训练 5】	124

## 第 6 章 BOM 和 DOM



6.1 认识 BOM	127
6.2 Window 对象	127
【同步案例 6-1】：在主窗口中通过单击超链接打开新窗口	129
【同步案例 6-2】：设置时间提醒器	130
【同步案例 6-3】：延时 5 秒关闭窗口	131
6.3 Navigator 对象	132
【同步案例 6-4】：显示当前浏览器的信息	133
【同步案例 6-5】：显示当前窗口占据显示器的区域大小	134
6.4 Document 对象	135
【同步案例 6-6】：退出页面时显示浏览当前页面的时间	135
【同步案例 6-7】：打开新窗口并输出一则通知信息	137
6.5 DOM 对象	138
6.5.1 什么是 DOM	138
6.5.2 DOM 的节点树	138
6.5.3 DOM 的节点类型	139
6.6 DOM 对象的分类	140
6.7 访问 DOM 对象	143
6.7.1 访问集合对象的方法	143
【同步案例 6-8】：显示当前 HTML 文档中所有的标签名	143
6.7.2 访问 HTML 当前页面指定	

元素的方法	144
【同步案例 6-9】：动态显示实时时间	144
【同步案例 6-10】：获取当前页面上所有的超链接	145
6.8 操纵 DOM 对象	146
6.8.1 创建 DOM 元素节点	146
【同步案例 6-11】：页面上显示欢迎词	147
6.8.2 删除 DOM 元素节点	147
【同步案例 6-12】：删除当前页面上所有的水平线	148
6.8.3 复制 DOM 元素节点	149
【同步案例 6-13】：实现复制表单示例	149
6.8.4 移动 DOM 元素节点	151
【同步案例 6-14】：将当前页面上第一个元素移到末尾	151
6.8.5 替换 DOM 元素节点	153
【同步案例 6-15】：替换元素节点示例	153
6.9 【综合案例 7】：创建根据不同来宾显示不同内容的欢迎栏	154
【知识点提炼】	156
【训练 6】	157

## 第7章 事件及事件处理

7.1 事件的概念	160	显示不同内容的对话框	172
7.1.1 认识事件和事件处理	160	【同步案例 7-7】：取消组合键的全选功能	173
7.1.2 事件绑定	161	7.5 处理页面事件	174
【同步案例 7-1】：单击页面上两个按钮，显示不同的问候语	162	7.5.1 页面加载与卸载事件	174
【同步案例 7-2】：使用动态绑定显示问候语	163	【同步案例 7-8】：网页加载时对图片进行缩小设置	176
7.2 事件对象	165	7.5.2 页面大小事件	177
7.2.1 Event 对象	165	7.6 处理文本编辑事件	178
7.2.2 事件流	165	7.6.1 处理文本选中区域	178
【同步案例 7-3】：显示触发事件名称	166	【同步案例 7-9】：页面显示鼠标指向的单词	179
【同步案例 7-4】：阻止事件的默认行为	167	7.6.2 文本的复制、剪切和粘贴操作	180
7.3 处理鼠标事件	168	【同步案例 7-10】：禁止复制粘贴方式输入密码	180
7.3.1 鼠标事件 	168	7.7 【综合案例 8】：显示按钮的说明性文字	182
7.3.2 处理鼠标事件	169	7.8 【综合案例 9】：实现返回首页、后退和前进按钮示例	183
【同步案例 7-5】：鼠标移入和移出时在页面显示不同的图形	169	【知识点提炼】	184
7.4 处理键盘事件	170	【训练 7】	185
7.4.1 键盘事件 	170		
7.4.2 键盘的键码值	170		
【同步案例 7-6】：按下不同按键时			

## 第8章 表单和表单对象

8.1 表单和表单对象	188	表单 	192
8.1.1 定义表单	188	【同步案例 8-2】：通过表单 id 属性访问表单示例	192
8.1.2 文本框	188	8.2.2 通过表单名称来访问	
8.1.3 单选按钮	189	表单 	193
8.1.4 复选框	189	【同步案例 8-3】：通过表单名称 name 属性访问表单示例	193
8.1.5 列表或菜单	189	【同步案例 8-4】：随机生成指定位数的验证码	194
8.1.6 按钮	190	8.2.3 通过表单标签名来访问	
【同步案例 8-1】：创建会员注册表单	190	表单	196
8.2 访问表单和表单元素	192		
8.2.1 通过表单的 ID 来访问			

【同步案例 8-5】：通过表单标签名 访问表单示例	196
8.3 操作表单对象	197
8.3.1 禁用和启用表单对象	197
【同步案例 8-6】：表单对象启用禁用 示例	197
8.3.2 表单对象获得和失去焦点	198
【同步案例 8-7】：表单对象获得失去 焦点示例	198
8.3.3 提交表单	199

【同步案例 8-8】：表单提交案例	200
8.3.4 重置表单	201
【同步案例 8-9】：重置表单示例	201
8.3.5 验证表单	202
【同步案例 8-10】：验证表单文本框 提交内容的合法性	202
8.4 【综合案例 10】：在线调查问卷	203
【知识点提炼】	205
【训练 8】	205

## 第 9 章 Ajax 初步

9.1 什么是 Ajax	209
9.2 Ajax 编程基础	209
9.2.1 初始化 XMLHttpRequest	209
9.2.2 XMLHttpRequest 对象的常用 属性	210
9.2.3 XMLHttpRequest 对象的常用 方法	210

9.2.4 FormData 对象	211
9.3 Ajax 应用案例	211
9.3.1 不刷新页面提交表单数据	211
9.3.2 通过 FormData 上传文件	214
【知识点提炼】	216
【训练 9】	216

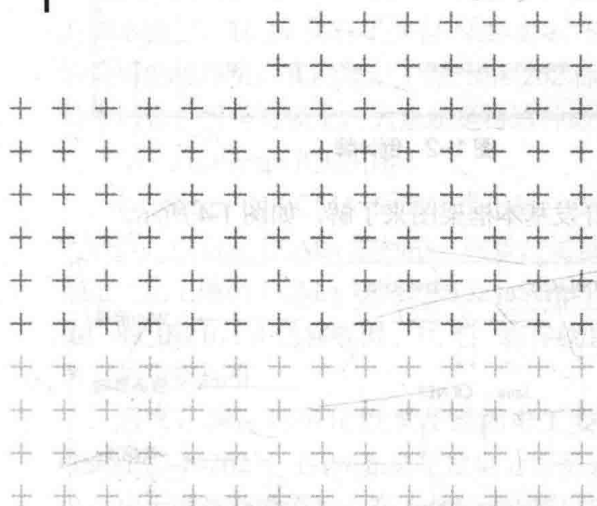
## 第 10 章 JavaScript 特效

10.1 页面导航栏特效	219
10.1.1 树形目录式导航菜单设计	219
10.1.2 带有注释的导航菜单设计	220
10.1.3 下拉式导航菜单设计	221
10.2 页面文本特效	224
10.2.1 以打字效果滚动显示文本	224
10.2.2 随机显示文字牌	226

10.2.3 自动滚屏的文本	227
10.3 图形特效	229
10.3.1 自动变形的图片特效	229
10.3.2 鼠标单击切换图片	230
10.3.3 网页动态选择图片	231

# 第1章

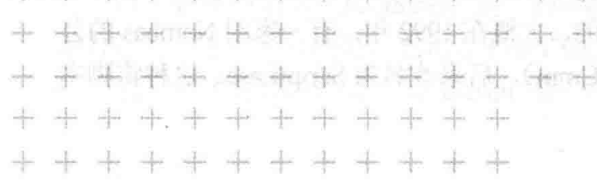
## 初识JavaScript



■ 什么是 JavaScript? JavaScript 能做什么? 怎么使用 JavaScript? 可能每个初次接触 JavaScript 的人都会有这些疑问, 我们一一进行解答。

### 学习目标:

- 了解JavaScript的发展历史及主要特点
- 了解JavaScript的应用范围
- 掌握JavaScript的语法规范以及在HTML文档中的应用



## 1.1 JavaScript 简介

JavaScript (JS) 是基于对象和事件驱动的客户脚本语言，主要是用来进行 Web 前端开发的，如图 1-1 所示的可以查看某个日期星期几的日历，图 1-2 所示的查看某个日期距离 2025 年 1 月 1 日多长时间的倒计时牌，图 1-3 所示的弹球小游戏等，都是典型的使用 JavaScript 开发的网页特效。



图 1-1 日历

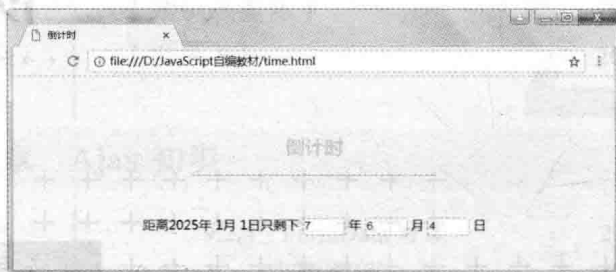


图 1-2 倒计时

什么是 Web 前端开发？可以通过常见的项目开发基本框架图来了解，如图 1-4 所示。

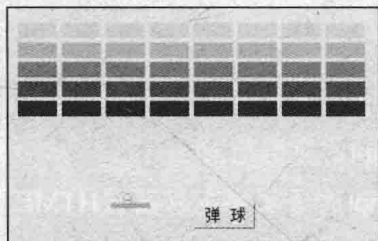


图 1-3 弹球小游戏

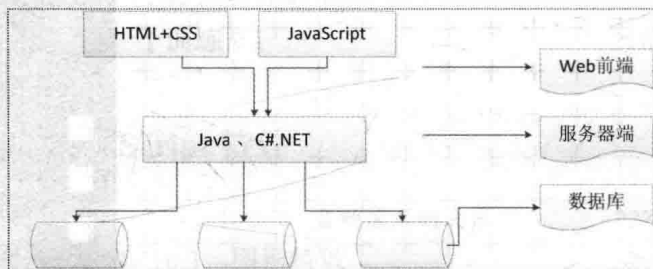


图 1-4 项目架构示意图

如图 1-4 所示，项目开发基本框架一般分为三层。第一层是 Web 前端页面开发，这个页面是用 HTML、CSS 和 JavaScript 开发的，其中 HTML+CSS 实现了前端页面的结构和样式，JavaScript 程序代码实现动态交互、网页特效等功能部分，这一层也就是常说的 Web 前端开发；中间一层是 Web 后台开发，也叫服务器端的程序开发，主要是进行业务处理；最后一层就是数据库了。在 Web 前端开发中，把纯粹的 HTML+CSS 开发的页面称为静态页面，这个页面是死的，没有用户交互功能，也没有什么特效，用户体验度低，要解决这个问题就需要用 JavaScript 来处理，这也是本书主要学习的内容。

### 1.1.1 JavaScript 的发展概况和特点

JavaScript 是怎么来的？它的特点是什么？要解决这些问题，要先了解一下 JavaScript 的发展历史。

#### 1. JavaScript 的历史

##### (1) JavaScript 语言诞生

最早的 JavaScript 语言思想是从嵌入式语言来的，大概在 1992 年，有一家叫 Nombas 的公司开发出了嵌入式脚本语言 C-- (C-minus-minus, Cmm)，后来改名为 ScriptEase。这种将脚本

嵌入到网页中的设计方法也就成为 JavaScript 诞生的理论基础。在 1995 年,网景(Netscape)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。Netscape Navigator 2.0 与 Sun 公司合作开发的一个脚本语言,最开始的时候叫 LiveScript。为了营销的便利,借助于当时如日中天的 Java,将其更名为 JavaScript,JavaScript 1.0 版本就这样诞生了。

### (2) JavaScript 标准的制定

JavaScript 诞生后,成功地进行了推广,刺激了微软,所以微软也决定向浏览器进军,并发布了一个 JavaScript 克隆版,叫 JScript,搭载到 IE 浏览器。微软进来以后,加上后来 CEnvi 的 ScriptEase,就有三种不同的 JavaScript 版本。所以这个时候就需要一个统一的标准来进行语法和特性的统一,JavaScript 标准的制定就提上了日程。

在 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)。当时第 39 技术委员会(TC39)承担了制定一个标准化语法和语义的“通用、跨平台、中立于厂商”的脚本语言。TC39 集合了来自 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,共同制定了 ECMA-262 标准。该标准的出生,诞生了 ECMAScript 这种全新脚本语言,但是习惯上,大家还是将这种语言叫作 JavaScript。

### (3) JavaScript 的应用推广

JavaScript 的 ECMA 标准化推出后,各个公司都参照 JavaScript 的标准进行了浏览器引擎的修改,以适应 ECMA 的标准。网景公司虽然是最早推出 JavaScript 的,但是整整晚了一年才推出“完全遵循 ECMA 规范”的 JavaScript,而微软早在一年前就推出了“完全遵循 ECMA 规范”的 IE4.0,让微软取得了优先,这样就直接导致一个恶果:微软的 JScript 成为 JavaScript 语言的事实标准。

后来,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)。各 Web 浏览器也分别努力将 ECMAScript 作为 JavaScript 实现的基础。

由于网景的慢节奏,在 JavaScript 推广中,促成了微软的 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。所以,1999 年之后,所有的网页都是基于 JScript 来开发的。而网景在微软强大攻势下,于 1998 年全面溃败。但微软也没有一支独大,同年成立的 Mozilla 项目中的 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。另一个同微软构成竞争的是谷歌(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,抢占了大块的市场份额。后来苹果也成为浏览器的竞争者,苹果开发出了 Safari 浏览器,作为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)都运行着 Safari 浏览器,随着苹果的产品不断的发展,Safari 浏览器也分得了浏览器的大量份额。

从以上 JavaScript 的分析我们可以得知,JavaScript 是一门独立的脚本语言,虽然与 Java 语言名字相似,但二者没有一点关系,只是借了一下“名字”而已。

## 2. JavaScript 的特点

(1) 解释型语言:JavaScript 是一种解释型语言,其源代码不需要编译就可以通过浏览器解释运行。在编写代码时,它可以和 HTML 代码结合在一起解释执行。

(2) 基于对象的语言:JavaScript 是一种基于对象的语言,在运行时,可以运用对象的属

性和方法来实现各种功能，这个特点使 JavaScript 变得很强大。它可以使用内置对象，也可以使用自定义对象来实现比较复杂的功能。

(3) 数据安全性：JavaScript 是一种安全的语言，它不允许访问本地的硬盘，也不能将数据存在网络服务器，不允许对网络文档进行修改和删除，只能通过浏览器进行进行信息浏览和动态的交互，防止数据的丢失。

(4) 跨平台性：JavaScript 在运行时，只依赖于浏览器，与操作环境无关，只要有支持 JavaScript 的浏览器就可以正确地运行，对操作平台无要求。

就因为有了以上四大基本特点，JavaScript 在软件开发，特别是 Web 前端开发受到越来越多的关注，其应用范围也就越来越广。

## 1.1.2 JavaScript 的应用

JavaScript 语言具体可以应用到以下几个方面。

### 1. 验证用户输入的内容

JavaScript 语言可以实现在客户端对用户输入的数据进行验证，如在注册时，提示用户名的格式是否符合要求，设置密码时两次设置是否相同等。

### 2. 实现页面的特效文字

使用 JavaScript 语言可以实现页面文字的多种效果，如旋转、翻滚等。

### 3. 实现页面的动画效果

在浏览页面时，我们会发现一些类似下雨、下雪的动画效果，这些效果使页面更加生动，这些动画效果也可以通过 JavaScript 来实现。

### 4. 实现窗口浮动的广告

浏览页面时经常发现一些浮动的广告窗口，这些浮动的广告窗口也可以通过 JavaScript 来实现。除此之外，JavaScript 还广泛的应用在其他很多方面，例如 JavaScript 虽然是客户端脚本语言，但是也可以来开发服务器端的应用，现在比较时兴的 HTML5 APP 也是 HTML5 结合了 JavaScript 来开发的。

## 1.1.3 支持 JavaScript 的浏览器

众所周知，JavaScript 最初开发出来就是为了嵌套在浏览器中使用的，浏览器对 JavaScript 支持也是 JavaScript 代码能够正常解析运行的基础，那么，支持 JavaScript 浏览器有哪些呢？其现在在市场上主流的浏览器都支持 JavaScript，如图 1-5 所示的几款浏览器，都对 JavaScript 有很好的支持。



图 1-5 支持 JavaScript 的浏览器

## 1.2 编写 JavaScript 的工具

从原理上来说，我们可以用任何一种文本编辑工具来编写 JavaScript 程序，如记事本、Notepad++、EditPlus 等，但这一类的编辑工具在写代码时很不方便，因为 JavaScript 是可以嵌入 HTML 网页文档中的，所以我们可以用任何一款 HTML 网页文件的编辑工具来完成 JavaScript 程序的编写，如 Dreamwaver、HBuilder、WebStorm 等。运用这些软件工具来编写程序的好处是一是代码自动提示，减少我们编写代码的失误和提高编写代码的速度，二是这些工具都有强大的纠错能力，帮助我们减少错误，三是提供了编写代码的模板，可以帮助我们提高速度和质量。这些软件编辑工具可以从网上很容易找到，安装在计算机上就可以使用它们开始我们的 JavaScript 征程了。其操作界面如图 1-6 和图 1-7 所示。

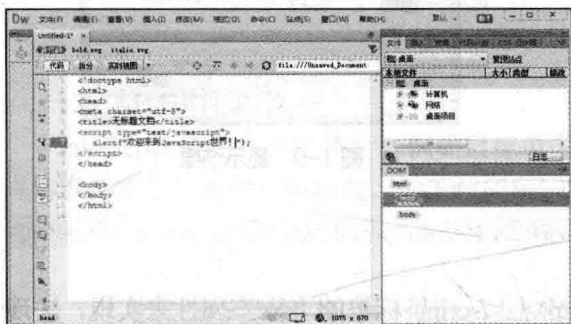


图 1-6 JavaScript 操作界面 (1)



图 1-7 JavaScript 操作界面 (2)

## 1.3 JavaScript 在 HTML 中的应用

通过前面的介绍，我们知道 JavaScript 主要是用来嵌入 HTML 中使用的，那么到底怎么使用呢？JavaScript 在 HTML 中的应用是很灵活的，但大致上有下面三种使用方式。

### 1.3.1 直接将 JavaScript 代码嵌入 HTML 中

JavaScript 代码是可以直接在 HTML 中编写的，怎么编写的，其实很简单，我们只要提供一对 `<script>...</script>` 即可，在这对标记中编写的代码浏览器就不会把它认为是 HTML 代码了，当然，在实际编写时，在这对标记中还有其他一些属性，如表 1-1 所示。

表 1-1 &lt;script&gt;标记属性说明

属性	属性说明
type	设置所使用的脚本语言，默认是“text/javascript”
src	设置引入外部文件的路径位置，如“js/hello.js”
defer	表示在 HTML 文档全部加载完毕后再执行脚本文件



微课 1-2

JavaScript 在 HTML  
中的应用

常用的在 HTML 中嵌入脚本的语法格式如下。

```
<script type="text/javascript">...</script>
```

提示

`<script>` 标记可以放在 `<head></head>` 中，也可以放在 `<body></body>` 中。为了在网页加载时能够先解析 JavaScript 代码，通常一般将 JavaScript 代码放在 `<head>` 标签中。

例如：使用 Dreamwaver 工具编写案例，在 HTML 中直接嵌入 JavaScript 脚本，在页面上弹出窗口显示“Hello! 欢迎来到 JavaScript 世界”，如图 1-8 所示，显示结果如图 1-9 所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>无标题文档</title>
6 <script type="text/javascript">
7     alert("Hello,欢迎来到JavaScript世界!");
8 </script>
9 </head>
10 <body>
11 </body>
12 </html>
```

图 1-8 在 HTML 中直接嵌入 JavaScript 脚本

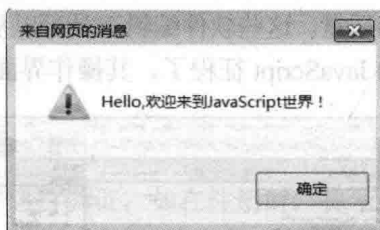


图 1-9 显示结果

### 1.3.2 链接外部的 JavaScript

链接外部的 JavaScript 脚本文件是通过 `<script>...</script>` 标记的“src”属性来实现，实现步骤如下。

(1) 建立一个 JavaScript 文件，JavaScript 文件的扩展名是 .js，并且在单独建立 JavaScript 文件时不需要添加 `<script>...</script>` 标签，直接写文件内容即可，如下代码所示。

```
alert("Hello, 欢迎来到JavaScript世界!");
```

(2) 将建立好的以 .js 为扩展名的文件保存到链接它的 HTML 文件相同的目录下。

(3) 在 HTML 文件中使用 `<script>` 标签的“src”属性链接该文件，如下代码所示。

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

例如，在 HTML 文件 hello.html 的同一路径下创建一个 JavaScript 文件“hello.js”，然后使用 `<script>` 标签的“src”属性链接该文件，如图 1-10 所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>无标题文档</title>
6 <script type="text/javascript" src="hello.js">
7
8 </script>
9 </head>
10 <body>
11 </body>
12 </html>
```

图 1-10 链接外部的 JavaScript

提示

在使用 `<script>` 标签的“src”属性链接外部 .js 文件时，外部 .js 文件要同 HTML 文件在同一目录下，因为“src”的路径文件一般使用相对路径来实现。