



工业和信息化“十三五”
人才培养规划教材



JavaScript

前端开发|模块化教程

JavaScript Development Modular Tutorial

赵建保 ◎ 主编

刘琳 贺辉平 邱尚明 彭华 ◎ 副主编



模块化的写作手法，突出**实践动手能力**，更符合学习要求

突出**前端工程师**职业能力培养

融入 **jQuery UI** 和 **Bootstrap UI** 组件开发模式

直戳**前端组件开发**核心技术

成果导向理念、**任务驱动**教学



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|模块化教程

JavaScript Development Modular Tutorial

赵建保 © 主编

刘琳 贺辉平 邱尚明 彭华 © 副主编

人民邮电出版社

北京

图书在版编目 (CIP) 数据

JavaScript前端开发模块化教程 / 赵建保主编. --
北京: 人民邮电出版社, 2019.2
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-49916-5

I. ①J… II. ①赵… 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253191号

内 容 提 要

JavaScript 是前端组件开发的核心技术, 前端组件是构建 Web 应用界面的基础。本书以前端工程师岗位工作任务为起点, 循序渐进, 以 HTML5、CSS3、JavaScript 为技术支撑, 以 Visual Studio Code 为开发环境, 将网页前端开发过程的任务系统化、规范化和组件化。全书设计开发了轮播图、折叠面板、选项卡、相册、滚动监听、表单验证、银行客服电话查询、微信运动步数统计图等 23 个工作任务, 包含了 JavaScript 核心技术、DOM、Canvas、JSON、Ajax 和性能优化等主题, 覆盖了 Web 项目开发中的常用组件的 HTML 结构、CSS 样式和 JavaScript 交互行为技术。同时注重梳理相关联的知识, 提炼设计模式, 剖析组件的工作机制, 以支持学习者更好地理解 and 运用相关技术适应前端组件项目开发的需求。

本书主要面向高等院校的数字媒体技术、软件技术、计算机应用技术、教育技术等专业的学生, 可作为“JavaScript 程序设计”“HTML5 Web 开发”“Web 前端开发”等课程的教材和参考书, 也可作为培训机构 Web 技术培训教材。

◆ 主 编 赵建保

副 主 编 刘 琳 贺辉平 邱尚明 彭 华

责任编辑 范博涛

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>

北京市艺辉印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 22.25

2019 年 2 月第 1 版

字数: 550 千字

2019 年 2 月北京第 1 次印刷

定价: 59.80 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

在作者十余年的数字媒体应用技术专业建设和 JavaScript 课程教学实践中,每每到网上书店或者实体书店选定教材,面对琳琅满目的 JavaScript 图书时我便油然而生感激和期待。感激有这么多技术大咖、一线教师、从业人员深耕技术,并不辞劳苦地分享各自的研究、开发实践和教学心得。然而已经出版的 JavaScript 图书多以技术为主导,又或者以某个项目为主导,多侧重于介绍孤立的知识和技术,没能融入前端开发工作过程中,由此激发了我编著此书的动力和决心。

我的初心是,以学习者胜任前端工程师岗位工作任务为目标,循序渐进,掌握 HTML5、CSS3、JavaScript 等技术的综合应用,熟悉 JavaScript 开发环境,适应网页前端开发工作的规范化、模块化和组件化岗位要求。我的思路是,以任务导入激发学习者的学习积极性,以工作过程必需的任务成果和目标维持学习者的学习动力和压力,以核心知识帮助学习者构建自己新的认知结构,任务实施过程划分为编写 HTML、编写 CSS、编写 JavaScript 和浏览器测试四大步骤,以强化训练提升学习者的知识和能力的迁移能力。期待这样的设计更有利于学习者提升 JavaScript 前端开发能力,更有利于学习者掌握 JavaScript 基本知识、基本技能、基本思想和基本开发经验,更有利于学习者提高分析问题、解决问题、发现问题和提出问题的能力,以及演绎思维和归纳思维的能力。

全书由赵建保负责整体设计、编写和统稿,刘琳、邱尚明、贺辉平、彭华参与教材设计与编写,参与编写的人员还有沈佳乐、曹盼盼、赖成发、胡铭畅、罗妩媚、石燕平、严梅燕、丁伟杰、蔡嘉威、沈泳鑫、陈响钦、邓金霞、朱进旺、吕晓纯、陆洁莹、李丽彤、邓洁美、谢涌彬、苏新栋。本书得到了广东省高职教育品牌专业项目的支持,企业一线前端开发人员黄亮、邱景生、杨起捷等参与了项目任务设计并给予了全程指导,还提出了很多宝贵建议,在此表示衷心的感谢。

由于 IT 行业发展迅猛,作者项目实践和知识视野有限,经验不够丰富,书中不足之处在所难免,恳请读者提出宝贵的意见和建议。联系方式(E-mail):mpcer@163.com。

编者

2018年10月

目录

CONTENTS

任务 1 搭建 JavaScript 开发环境

1.1 任务导入	2
1.2 成果目标	2
1.3 核心知识	3
1.3.1 JavaScript 演进	3
1.3.2 JavaScript 介绍	4
1.3.3 Web 页面渲染过程	5
1.3.4 Visual Studio Code 介绍	6
1.3.5 Visual Studio Code 快捷键	6
1.3.6 EMMET 语法	9
1.3.7 Node.js 介绍	10
1.3.8 http-server 介绍	10
1.3.9 在 HTML 中使用 JavaScript	11
1.3.10 高性能 JavaScript	12
1.3.11 JavaScript 执行顺序	12
1.3.12 脚本位置	13
1.3.13 组织脚本	13
1.3.14 无阻塞脚本	14
1.3.15 选取 DOM 对象	17
1.3.16 addEventListener	20
1.3.17 读写 HTML DOM style 对象	
属性	20
1.3.18 cssText	21
1.4 任务实施	22
1.4.1 安装和配置 Visual Studio Code	22
1.4.2 安装常用扩展	23
1.4.3 Chrome 浏览器	24
1.4.4 Chrome 开发者工具	24
1.4.5 安装 Node.js	25
1.4.6 安装与配置 http-server	25
1.4.7 编写 HTML 和 CSS	25

1.4.8 编写 JavaScript	27
1.4.9 测试页面	28
1.5 强化训练	28
1.6 学习成果评量	29

任务 2 斑马线表格制作

2.1 任务导入	31
2.2 成果目标	31
2.3 核心知识	32
2.3.1 表格常用标签	32
2.3.2 表格斑马线原理	33
2.3.3 读写 HTML DOM className 属性	33
2.3.4 严格模式 (use strict)	34
2.3.5 定义变量	35
2.3.6 常量	35
2.3.7 变量命名规则	35
2.3.8 JavaScript 语法规则	36
2.3.9 变量类型	37
2.3.10 变量作用域	38
2.3.11 避免变量污染	40
2.3.12 闭包函数	40
2.3.13 JavaScript 转义字符	40
2.3.14 相等操作符	41
2.3.15 toNumber	41
2.3.16 使用 typeof 检测类型	42
2.3.17 使用 constructor 检测类型	42
2.3.18 使用 toString() 检测封装类型	42
2.3.19 事件委托	43
2.4 任务实施	44
2.4.1 编写 HTML	44
2.4.2 编写 CSS 样式	46
2.4.3 编写 JavaScript	47

2.4.4 测试页面.....	48
2.5 强化训练.....	49
2.6 学习成果评量.....	49

任务3 弹出消息框..... 50

3.1 任务导入.....	51
3.2 成果目标.....	51
3.3 核心知识.....	51
3.3.1 基于负边距的垂直居中.....	51
3.3.2 基于 transform 属性的垂直居中.....	52
3.3.3 元素动画制作.....	53
3.3.4 警告对话框 alert().....	53
3.3.5 确认对话框 confirm().....	54
3.3.6 提示对话框 prompt().....	54
3.3.7 BOM 介绍.....	54
3.3.8 window 对象.....	54
3.3.9 location 对象.....	55
3.3.10 screen 对象.....	56
3.3.11 history 对象.....	57
3.4 任务实施.....	57
3.4.1 编写 HTML.....	57
3.4.2 编写 CSS 样式.....	58
3.4.3 编写 JavaScript.....	60
3.4.4 测试页面.....	60
3.5 强化训练.....	61
3.6 学习成果评量.....	61

任务4 图片缩放特效..... 62

4.1 任务导入.....	63
4.2 成果目标.....	63
4.3 核心知识.....	63
4.3.1 DOM 编程.....	63
4.3.2 DOM 访问与修改.....	64
4.3.3 DOM 遍历.....	64
4.3.4 innerHTML 对比 DOM 方法.....	65
4.3.5 字符串连接.....	65
4.3.6 HTML 集合 length.....	66

4.3.7 减少浏览器重排与重绘.....	67
4.4 任务实施.....	69
4.4.1 编写页面结构.....	69
4.4.2 编写 CSS 样式.....	71
4.4.3 编写 JavaScript.....	72
4.4.4 测试页面.....	73
4.5 强化训练.....	73
4.6 学习成果评量.....	74

任务5 网页换肤..... 75

5.1 任务导入.....	76
5.2 成果目标.....	76
5.3 核心知识.....	76
5.3.1 网页换肤原理.....	76
5.3.2 HTML 文档对象模型.....	77
5.3.3 HTML DOM 节点树.....	77
5.3.4 document 对象属性和方法.....	78
5.3.5 element 对象属性和方法.....	79
5.3.6 获取元素属性 getAttribute().....	82
5.3.7 设置元素属性 setAttribute().....	83
5.3.8 本地数据存储方案.....	83
5.3.9 WebStorage.....	84
5.3.10 WebStorage 基本属性和方法.....	84
5.3.11 使用 sessionStorage 对象.....	84
5.3.12 使用 localStorage 对象.....	86
5.3.13 使用 storage 事件.....	86
5.3.14 cookie 介绍.....	87
5.3.15 cookie 构成.....	87
5.3.16 写入 cookie 信息.....	88
5.3.17 读取 cookie 信息.....	90
5.4 任务实施.....	91
5.4.1 编写 HTML.....	91
5.4.2 编写 CSS.....	92
5.4.3 编写 JavaScript.....	94
5.4.4 测试页面.....	95
5.5 强化训练.....	95
5.6 学习成果评量.....	95

任务6 下拉广告96

6.1 任务导入.....	97
6.2 成果目标.....	97
6.3 核心知识.....	97
6.3.1 transition 属性.....	97
6.3.2 超时调用 setTimeout()方法.....	98
6.4 任务实施.....	99
6.4.1 编写 HTML.....	99
6.4.2 编写 CSS 样式.....	100
6.4.3 编写 JavaScript 代码.....	101
6.4.4 测试页面.....	102
6.5 强化训练.....	102
6.6 学习成果评量.....	103

任务7 轮播图.....104

7.1 任务导入.....	105
7.2 成果目标.....	105
7.3 核心知识.....	105
7.3.1 间歇调用 setInterval().....	105
7.3.2 避免常见 JavaScript 错误.....	106
7.3.3 理解 Error 对象.....	106
7.3.4 错误处理思路.....	107
7.3.5 使用浏览器控制台调试程序.....	108
7.3.6 使用断点调试程序.....	109
7.3.7 使用 try-catch 处理异常.....	110
7.4 任务实施.....	111
7.4.1 编写 HTML.....	111
7.4.2 编写 CSS 样式.....	113
7.4.3 编写 JavaScript.....	116
7.4.4 测试页面.....	117
7.5 强化训练.....	117
7.6 学习成果评量.....	118

任务8 滚动公告119

8.1 任务导入.....	120
---------------	-----

8.2 学习成果..... 120

8.3 核心知识..... 120

8.3.1 HTML 事件模型.....	120
8.3.2 DOM0 级事件模型.....	121
8.3.3 DOM2 级事件模型.....	122
8.3.4 IE 事件模型.....	123

8.4 任务实施 124

8.4.1 编写 HTML.....	124
8.4.2 编写 CSS 样式.....	125
8.4.3 编写 JavaScript.....	126
8.4.4 测试页面.....	127

8.5 强化训练..... 128

8.6 学习成果评量 128

任务9 贷款计算器..... 129

9.1 任务导入 130

9.2 成果目标..... 130

9.3 核心知识..... 130

9.3.1 表单类型.....	130
9.3.2 表单结构.....	132
9.3.3 form 对象.....	133
9.3.4 表单事件.....	134
9.3.5 表达式与操作符.....	135
9.3.6 转换为数字.....	136
9.3.7 设置小数位数.....	137
9.3.8 Math 对象.....	137
9.3.9 条件语句.....	138
9.3.10 for 循环.....	141
9.3.11 while 循环.....	142
9.3.12 do-while 循环.....	142
9.3.13 for-in 循环.....	142
9.3.14 优化循环性能.....	143
9.3.15 forEach().....	143

9.4 任务实施 143

9.4.1 编写 HTML.....	143
9.4.2 编写 CSS 样式.....	145
9.4.3 编写 JavaScript.....	147
9.4.4 测试页面.....	147

9.5 强化训练	148
9.6 学习成果评量	148

任务 10 计算器 149

10.1 任务导入	150
10.2 成果目标	150
10.3 核心知识	150
10.3.1 函数介绍	150
10.3.2 定义函数	151
10.3.3 嵌套函数	152
10.3.4 调用函数	152
10.3.5 函数的实参和形参	155
10.3.6 将对象属性用作实参	157
10.3.7 实参类型	157
10.3.8 作为值的函数	157
10.3.9 自定义函数属性	158
10.3.10 slice()	158
10.3.11 isNaN()	158
10.4 任务实施	159
10.4.1 编写 HTML	159
10.4.2 编写 CSS 样式	160
10.4.3 编写 JavaScript	162
10.4.4 测试页面	164
10.5 强化训练	165
10.6 学习成果评量	165

任务 11 投票 166

11.1 任务导入	167
11.2 成果目标	167
11.3 核心知识	167
11.3.1 匿名函数	167
11.3.2 数据存取方式	168
11.3.3 对象成员	169
11.3.4 函数作用域	170
11.3.5 闭包	171
11.3.6 闭包函数	173
11.3.7 递归函数	174
11.4 任务实施	174

11.4.1 编写 HTML	174
----------------	-----

11.4.2 编写 CSS 样式	175
------------------	-----

11.4.3 编写 JavaScript	176
----------------------	-----

11.4.4 测试页面	177
-------------	-----

11.5 强化训练	177
-----------	-----

11.6 学习成果评量	178
-------------	-----

任务 12 折叠面板 179

12.1 任务导入	180
12.2 成果目标	180
12.3 核心知识	180
12.4 任务实施	181
12.4.1 编写 HTML	181
12.4.2 编写 CSS	182
12.4.3 编写 JavaScript	183
12.4.4 测试页面	184
12.5 强化训练	184
12.6 学习成果评量	185

任务 13 银行客服电话查询 186

13.1 任务导入	187
13.2 成果目标	187
13.3 核心知识	187
13.3.1 Ajax 简介	187
13.3.2 Ajax 原理	188
13.3.3 HTTP 请求	188
13.3.4 HTTP 状态码	189
13.3.5 定义 XMLHttpRequest 对象	190
13.3.6 建立 XMLHttpRequest 连接	191
13.3.7 跟踪状态	192
13.3.8 中止请求	192
13.3.9 Ajax 请求与响应模板	192
13.3.10 获取数据	193
13.3.11 获取纯文本	194
13.3.12 使用 Ajax 加载 HTML	194
13.3.13 使用 Ajax 加载 JSON	194

13.3.14 获取 JavaScript 脚本	194
13.3.15 使用 Ajax 加载其他服务器的 数据	194
13.3.16 获取头部信息	195
13.3.17 JSONP 工作原理	195
13.4 任务实施	196
13.4.1 编写 HTML	196
13.4.2 编写 CSS 样式	196
13.4.3 编写 JSON	197
13.4.4 编写 JavaScript	199
13.4.5 测试页面	200
13.5 强化训练	200
13.6 学习成果评量	200
任务 14 省、市、区联动菜单...	201
14.1 任务导入	202
14.2 成果目标	202
14.3 核心知识	202
14.3.1 下拉列表 select	202
14.3.2 HTML DOM Option 对象	203
14.3.3 select add()方法	203
14.3.4 JSON 简介	203
14.3.5 JSON 语法	204
14.3.6 JSON 与 XML 比较	205
14.3.7 访问 JSON 对象值	206
14.3.8 遍历 JSON 对象	206
14.3.9 修改 JSON 值	207
14.3.10 删除对象属性	207
14.3.11 解析 JSON 对象	207
14.3.12 JSON.stringify()	207
14.3.13 JSON 转换为 JavaScript 对象	208
14.4 任务实施	208
14.4.1 编写 HTML	208
14.4.2 编写 CSS	209
14.4.3 JSON 数据准备	210
14.4.4 编写 JavaScript	211
14.4.5 测试页面	212

14.5 强化训练	213
14.6 学习成果评量	213
任务 15 滚动监听.....	214
15.1 任务导入	215
15.2 成果目标	215
15.3 核心知识	215
15.3.1 滚动监听	215
15.3.2 CSS 脚本化	215
15.3.3 访问 CSS 行内样式	216
15.3.4 使用 styleSheets 对象	217
15.3.5 计算样式	218
15.3.6 元素尺寸	220
15.3.7 window.scrollTo	220
15.4 任务实施	221
15.4.1 编写 HTML	221
15.4.2 编写 CSS 样式	222
15.4.3 编写 JavaScript	223
15.4.4 测试页面	224
15.5 强化训练	224
15.6 学习成果评量	225
任务 16 视频播放器	226
16.1 任务导入	227
16.2 成果目标	227
16.3 核心知识	227
16.3.1 HTML video 标签	227
16.3.2 HTML 音频/视频方法	228
16.3.3 HTML 音频/视频属性	228
16.3.4 HTML 音频/视频事件	229
16.3.5 浏览器支持的视频格式	230
16.3.6 浏览器视频能力检测	231
16.3.7 实现播放列表功能	231
16.4 任务实施	232
16.4.1 编写 HTML	232
16.4.2 编写 CSS 样式	233
16.4.3 编写 JavaScript	235
16.4.4 测试页面	236

16.5 强化训练.....	236	17.4 任务实施.....	258
16.6 学习成果评量.....	236	17.4.1 编写 HTML.....	258
任务 17 刮刮乐.....	237	17.4.2 编写 CSS 样式.....	258
17.1 任务导入.....	238	17.4.3 编写 JavaScript.....	259
17.2 成果目标.....	238	17.4.4 测试页面.....	260
17.3 核心知识.....	238	17.5 强化训练.....	260
17.3.1 事件基础.....	238	17.6 学习成果评量.....	260
17.3.2 事件流.....	239	任务 18 微信运动步数统计图....	261
17.3.3 事件冒泡.....	239	18.1 任务导入.....	262
17.3.4 事件捕获.....	240	18.2 成果目标.....	262
17.3.5 事件对象.....	240	18.3 核心知识.....	262
17.3.6 IE 中的事件对象.....	243	18.3.1 HTML5 canvas.....	262
17.3.7 跨浏览器的事件对象.....	245	18.3.2 canvas 坐标.....	263
17.3.8 共享 onload 事件.....	247	18.3.3 canvas 绘图步骤.....	263
17.3.9 事件委托.....	248	18.3.4 canvas 绘制渐变色.....	263
17.3.10 事件类型.....	249	18.3.5 canvas 绘制文本.....	263
17.3.11 UI 事件.....	249	18.3.6 canvas 绘制直线.....	264
17.3.12 焦点事件.....	250	18.3.7 canvas 绘制矩形.....	264
17.3.13 鼠标事件.....	250	18.3.8 canvas 绘制圆形.....	265
17.3.14 键盘事件.....	251	18.3.9 canvas 绘制曲线.....	265
17.3.15 鼠标/键盘事件对象属性.....	251	18.4 任务实施.....	265
17.3.16 鼠标/键盘事件方法.....	252	18.4.1 编写 HTML.....	265
17.3.17 框架/对象 (Frame/Object) 事件.....	252	18.4.2 编写 JavaScript.....	266
17.3.18 表单事件.....	253	18.4.3 测试页面.....	267
17.3.19 剪贴板事件.....	253	18.5 强化训练.....	267
17.3.20 打印事件.....	253	18.6 学习成果评量.....	268
17.3.21 拖动事件.....	253	任务 19 相册.....	269
17.3.22 多媒体 (Media) 事件.....	254	19.1 任务导入.....	270
17.3.23 动画事件.....	254	19.2 成果目标.....	270
17.3.24 过渡事件.....	255	19.3 任务实施.....	270
17.3.25 其他事件.....	255	19.3.1 编写 HTML.....	270
17.3.26 模拟事件过程.....	255	19.3.2 编写 CSS 样式.....	273
17.3.27 模拟鼠标事件.....	256	19.3.3 编写 JavaScript.....	276
17.3.28 模拟键盘事件.....	257	19.3.4 测试页面.....	277
17.3.29 globalCompositeOperation 属性.....	257	19.4 强化训练.....	278
		19.5 学习成果评量.....	278

任务 20 选项卡.....279

20.1 任务导入.....	280
20.2 成果目标.....	280
20.3 核心知识.....	281
20.3.1 选项卡 HTML 模型.....	281
20.3.2 重置 ul 属性.....	281
20.3.3 浮动 (float).....	282
20.3.4 绝对定位 (absolute).....	283
20.3.5 选项卡切换原理.....	283
20.4 任务实施.....	283
20.4.1 编写 HTML.....	283
20.4.2 编写 CSS.....	286
20.4.3 编写 JavaScript 脚本.....	288
20.4.4 浏览器测试.....	289
20.5 强化训练.....	289
20.6 学习成果评量.....	289

任务 21 JavaScript 抽奖器.....290

21.1 任务导入.....	291
21.2 成果目标.....	291
21.3 核心知识.....	291
21.3.1 数据存储.....	291
21.3.2 数组.....	292
21.3.3 创建数组.....	292
21.3.4 数组元素的读和写.....	293
21.3.5 数组元素的添加和删除.....	293
21.3.6 稀疏数组.....	293
21.3.7 数组长度.....	294
21.3.8 数组遍历.....	294
21.3.9 ECMAScript3 数组方法.....	294
21.3.10 ECMAScript5 数组方法.....	296
21.4 任务实施.....	298
21.4.1 编写 HTML.....	298
21.4.2 编写 CSS.....	299
21.4.3 编写 JavaScript.....	300
21.4.4 测试页面.....	301
21.5 强化训练.....	301

21.6 学习成果评量.....302

任务 22 座位预订程序.....303

22.1 任务导入.....	304
22.2 成果目标.....	304
22.3 核心知识.....	304
22.3.1 面向对象背景.....	304
22.3.2 对象的属性和方法.....	305
22.3.3 创建对象方法——字面量 语法.....	305
22.3.4 创建对象方法——构造函数 语法.....	305
22.3.5 添加和删除属性.....	307
22.3.6 访问对象.....	307
22.3.7 this 关键字.....	307
22.3.8 OOP 相关概念.....	308
22.3.9 浏览器内置对象.....	308
22.3.10 浏览器对象模型 BOM.....	309
22.3.11 document 对象.....	309
22.3.12 window 对象.....	311
22.3.13 navigator 对象集合.....	313
22.3.14 screen 对象.....	313
22.3.15 history 对象.....	314
22.3.16 location 对象.....	314
22.3.17 字符串对象.....	314
22.3.18 日期对象.....	315
22.3.19 数组对象.....	317
22.3.20 逻辑对象.....	317
22.3.21 算术对象.....	318
22.3.22 Number 对象.....	318
22.3.23 Form 对象.....	319
22.3.24 iframe 对象属性.....	319
22.4 任务实施.....	320
22.4.1 编写 HTML.....	320
22.4.2 编写 CSS 样式.....	321
22.4.3 编写 JavaScript.....	323
22.4.4 测试页面.....	324
22.5 强化训练.....	324

22.6 学习成果评量.....	325	23.3.11 后向引用.....	331
任务 23 注册表单验证.....	326	23.3.12 零宽断言.....	331
23.1 任务导入.....	327	23.3.13 其他语法.....	332
23.2 成果目标.....	327	23.3.14 常用简易规则.....	332
23.3 核心知识.....	327	23.3.15 常用正则表达式.....	333
23.3.1 正则表达式的概念.....	327	23.4 任务实施.....	338
23.3.2 正则表达式的工作原理.....	328	23.4.1 编写 HTML.....	338
23.3.3 定义正则表达式.....	328	23.4.2 编写 CSS 样式.....	339
23.3.4 元字符.....	329	23.4.3 编写 JavaScript.....	341
23.3.5 反义字符.....	329	23.4.4 测试页面.....	342
23.3.6 限定字符.....	330	23.5 强化训练.....	343
23.3.7 转义字符.....	330	23.6 学习成果评量.....	343
23.3.8 字符分支.....	330		
23.3.9 字符分组.....	330	参考文献.....	344
23.3.10 贪婪匹配和懒惰匹配.....	331		

Chapter

1

任务 1

搭建 JavaScript 开发环境

JavaScript



1.1 任务导入

在 Web 标准中，HTML 定义了页面结构和内容，CSS 定义了页面布局和外观，如颜色、字体、边框、边距和版式布局等，而 JavaScript 定义了页面交互行为，比如元素交互、表单验证、网页游戏等。如今，缺乏动态和交互性的网站已经没有市场，网站必须以新颖的方式与用户互动，页面交互性和动态性已经是商业 Web 应用开发中前端开发的必备能力。随着浏览器 JavaScript 引擎性能的提升，未来在 Web 图像、音频及视频处理、虚拟现实、游戏开发等方面前途无量。要想成为 Web 开发工程师，掌握 JavaScript 必不可少。

正式开始学习 JavaScript 前端开发之前，需要先配置满足 Web 前端开发和测试的学习环境，主要是 JavaScript 编辑器和 Web 浏览器，其中 JavaScript 编辑器用于编写 HTML、CSS 和 JavaScript 前端代码，Web 浏览器用于 Web 应用的开发测试。通过本任务，您将学会如何配置 JavaScript 前端编码和测试环境，并体验鼠标 mouseover、mouseout 事件时表格行背景变色效果的开发，图 1-1 所示为表格行悬停提示效果，从此开启您的前端征程。



图1-1 表格行悬停提示效果

1.2 成果目标

本任务旨在学会搭建前端开发环境，熟悉前端页面开发的基本操作流程，熟悉 Visual Studio Coder 的使用，培养前端开发的意识和兴趣。

知识目标	技能目标	素质目标
1. 了解 JavaScript 发展历程 2. 了解 Web 页面渲染过程 3. 了解 Visual Studio Code 4. 理解 EMMET 语法 5. 理清 DOM 对象选取模式 6. 理解 addEventListener 参数	1. 安装和配置 Visual Studio Code 2. 使用 Chrome 开发者工具 3. 安装与配置 http-server 4. 前端页面开发流程 5. 合理确定 JavaScript 在 HTML 文档中的位置 6. 善用 addEventListener 7. 善用 style 对象操作样式	1. 能梳理 JavaScript 演进 2. 感受前端开发过程 3. 培养前端开发的思维习惯

1.3 核心知识

1.3.1 JavaScript 演进

理解 JavaScript 的最好方法之一，就是了解 JavaScript 的历史。

JavaScript 为互联网而生，紧随着浏览器的出现而问世。回顾它的历史，就要从浏览器的历史说起。1990 年 12 月 25 日，英国计算机科学家伯纳·李爵士 (Tim Berners-Lee) 发明了万维网 (World Wide Web)。1993 年，美国国家超级计算机应用中心 (NCSA) 开始开发一个独立的浏览器 Mosaic。1994 年 10 月，Mosaic 通信公司 (Mosaic Communications) 成立，不久后改名为 Netscape。1994 年 12 月，Navigator 发布了 1.0 版，市场份额一举超过 90%。Netscape 同微软公司在浏览器市场的竞争异常激烈，争相给浏览器添加新功能，获取竞争优势。1995 年 2 月，Netscape 公司发布 Netscape Navigator 2 浏览器，该公司的布兰登·艾奇 (Brendan Eich) 为 Navigator 2 浏览器开发了 LiveScript 脚本语言，主要目的是处理表单数据验证，避免由服务器端验证导致的延时问题。LiveScript 语法借鉴 Java，函数借鉴 Scheme，原型继承借鉴 Self，正则表达式特性则借鉴 Perl。当时正逢 Sun 公司的股票飞涨，为搭上 Java 成功的顺风车，Netscape 将 LiveScript 改名为 JavaScript。由于 JavaScript 1.0 获得了巨大成功，Netscape 公司随即在 Netscape Navigator 3 中又发布了 JavaScript 1.1 版本。

在 Netscape Navigator 3 发布后不久，微软不甘示弱，紧跟着 Netscape 在 Internet Explorer 3 中加入 JavaScript 脚本语言，为了避免与 Netscape 的 JavaScript 产生纠纷，微软特意将其命名为 JScript，这种语言和 JavaScript 很像，浏览器大战就此爆发。自微软在操作系统中内置 Internet Explorer 3，Netscape 就面临着即将丧失浏览器脚本语言主导权的局面。1996 年 11 月，网景公司决定将 JavaScript 提交给国际标准化组织 ECMA，希望 JavaScript 能够成为国际标准，以此抵抗微软。1997 年，欧洲计算机制造商协会 (ECMA) 以 JavaScript 1.1 为蓝本制订了 ECMA-262 新脚本语言的标准，并命名为 ECMAScript，这个版本就是 ECMAScript 1.0 版，它规定了浏览器脚本语言的标准。之所以不叫 JavaScript，一方面是由于商标的关系，Java 是 Sun 公司的商标，根据一份授权协议，只有 Netscape 公司可以合法地使用 JavaScript 这个名字，且 JavaScript 已经被 Netscape 公司注册为商标；另一方面也是想体现这门语言的制定者是 ECMA，不是 Netscape，这样有利于保证这门语言的开放性和中立性。因此，ECMAScript 和 JavaScript 的关系是，前者是后者的规格，后者是前者的一种实现。1998 年 6 月，ECMAScript 2.0 版发布。1999 年 12 月，ECMAScript 3.0 版发布，成为 JavaScript 的通行标准，得到了广泛支持。2007 年 10 月，ECMAScript 4.0 版草案发布，对 3.0 版做了大幅升级，由于 4.0 版的目标过于激进，各方对于是否通过这个标准发生了严重分歧。以 Yahoo、Microsoft、Google 为首的大公司，反对 JavaScript 的大幅升级，主张小幅改动；以 JavaScript 创造者 Brendan Eich 为首的 Mozilla 公司，则坚持当前的草案。2008 年 7 月，由于对于下一个版本应该包括哪些功能，各方分歧太大，争论过于激进，ECMA 开会决定，中止 ECMAScript 4.0 的开发，将其中涉及现有功能改善的一小部分发布为 ECMAScript 3.1，而将其他激进的设想扩大范围，放入以后的版本，由于会议的气氛，该版本的项目代号起名为 Harmony (和谐)。2009 年 12 月，ECMAScript 5.0 版正式发布。Harmony 项目则一分为二，一些较为可行的设想定名为 JavaScript.next 继续开发，后来

演变成 ECMAScript 6, 2015 年 6 月, ECMAScript 6 发布了正式版本, 并更名为 ECMAScript 2015。如图 1-2 所示为 JavaScript 演进路线。



图1-2 JavaScript演进路线

随着 JavaScript 再次成为关注的焦点, 很多杰出的编程人员致力于改善 JavaScript 解释器, 极大地改善了其运行阶段的性能。随着 Gmail、GoogleMaps 这一类客户端应用和 Ajax 的相继出现, 大大促进了 JavaScript 开发模式的变革, 也把 JavaScript 催生成为一种成熟的、某些方面独一无二的、拥有强大原型体系的面向对象语言。

1.3.2 JavaScript 介绍

程序设计语言分为解释型和编译型两大类。Java 和 C++ 等语言需要一个编译器 (Compiler)。编码器是一种程序, 能够把用 Java 等高级语言编写出来的源代码翻译为直接在计算机上执行的文件, 代码错误在编译阶段就能被发现。解释型程序设计语言不需要编译器, 运行仅需要解释器。JavaScript 语言则由 Web 浏览器负责完成有关的解释和执行工作。JavaScript 是 Web 页面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能的解释型脚本语言, 不但可用于编写客户端的脚本程序, 由 Web 浏览器解释执行, 还可以编写在服务器端执行的脚本程序, 在服务器端处理用户提交的信息并动态地向客户端浏览器返回处理结果。浏览器中的 JavaScript 解释器将直接读入源代码并执行, 代码错误只能等到解释器执行到有关代码时才能被发现。如果浏览器中没有解释器, JavaScript 代码就无法执行。

JavaScript 脚本语言的主要特点如下。

(1) 解释性。不同于一些编译性程序语言 (如 C、C++), JavaScript 源代码不需要经过编译, 而是直接嵌入在 HTML 页面中, 使得前端页面支持用户交互并响应相应事件, 在浏览器中运行时被解释。

(2) 基于对象。许多功能运用脚本环境对象的方法与脚本的相互作用来完成。

(3) 事件驱动。JavaScript 可以直接对用户页面的操作行为 (鼠标、键盘、手势等) 做出响应, 无须经过 Web 服务器处理。

(4) 跨平台。JavaScript 依赖于浏览器本身, 与操作环境无关, 只要计算机能运行浏览器并支持 JavaScript 的浏览器, 就可以正确执行。

(5) 安全性。JavaScript 是一种安全性语言, 不允许访问本地硬盘, 不能将数据存入服务器上, 不允许对网络文档进行修改和删除, 只能通过浏览器实现信息浏览或动态交互, 以防止数据丢失和篡改。

JavaScript 由一项主要适用于浏览器客户端的计算机技术, 从一个简单的输入验证器逐渐发展成为一种多功能的强大的程序设计语言, 甚至连服务端也能用它来编写, 能够处理复杂的计算

和交互,拥有闭包、匿名函数等,甚至元编程特性。现在已经具备了与浏览器窗口及其内容等几乎所有方面交互的能力。在互联网发展的早期,JavaScript 就已经成为了支撑网页内容交互体验的基础技术,现在 JavaScript 毫无疑问已经成为了 Web 的核心技术,能够地理定位、播放视频和音频、绘图等,速度更快,性能更强。

JavaScript 是世界上最流利的编程语言之一。使用 JavaScript 不仅可以开发浏览器应用程序,也可以开发手机应用(APP)和服务器端程序。JavaScript 可实现以下功能。

(1) 创建拥有强大而丰富功能的 Web 应用程序,多数运行于 Web 浏览器,基于 HTML5 可开发应用缓存、本地存储、本地数据库等 Web 应用。

(2) 使用 Node.js 编写服务器端脚本。

(3) 开发移动设备应用程序。比如基于 HTML5+ 的移动 APP 开发。

1.3.3 Web 页面渲染过程

Web 技术的根基是 HTML、CSS 和 JavaScript,其中 HTML 控制内容和结构,CSS 控制表现,而 JavaScript 则控制交互行为。换句话说,JavaScript 是让 HTML 和 CSS 协同运作的黏合剂。一个完整的 JavaScript 实现应该包括以下三个不同部分:核心(ECMAScript)、文档对象模型(DOM)和浏览器对象模型(BOM),如图 1-3 所示。

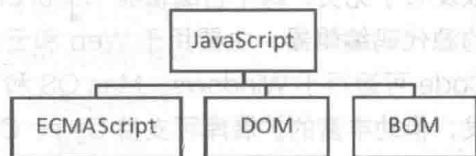


图 1-3 JavaScript 实现

ECMAScript 规定了 JavaScript 的语法、类型、语句、关键字、保留字、操作符和对象,它是语言的核心部分,包括变量、函数、循环等,独立于浏览器,并可以在其他环境中使用。

文档对象模型(DOM)是用于操作 HTML 和 XML 文档的应用程序接口(Application Programming Interface, API),提供了一种与 HTML、XML 文档交互的方式。在浏览器中,主要用来操作 HTML 文档。HTML 和 DOM 把整个页面映射为一个多层节点结构。HTML 或 XML 页面中的每个组成部分都是某种类型的节点,这些节点又包含着不同类型的数据。DOM1 级(DOM Level1)于 1998 年 10 月成为 W3C 推荐标准,包括 DOM 核心(DOM Core)和 DOM HTML。DOM 核心规定如何映射基于 XML 的文档结构,以便简化对文档中任意部分的访问和操作。DOM HTML 模块添加了针对 HTML 的对象和方法。DOM2 包括 DOM 视图、DOM 事件、DOM 样式、DOM 遍历和范围。浏览器对象模型(BOM)实际上是一个与浏览器环境有关的对象集合,使开发人员可以控制浏览器显示的页面以外的部分,BOM 只处理浏览器窗口和框架。

Web 页面渲染过程如图 1-4 所示,图中实线表示先后关系,虚线表示渲染过程可能调用的模块。如页面下载时需要使用网络和存储,计算布局和绘图时需要使用 2D/3D 图形模块生成可视化结果。具体渲染过程是:网页内容输入到 HTML 解释器,HTML 解释器在解析后构建 DOM 树,如果遇到 JavaScript 代码则交给 JavaScript 引擎去处理,如果网页包含 CSS 则交给 CSS 解释器去解释,没有被定义 CSS 的元素渲染引擎将默认样式应用到 HTML 元素上。当 DOM 建立的时候,渲染引擎接收来自 CSS 解释器的样式信息,构建一个新的内部绘图模型。该模型由布局模块计算模型内部各元素的位置和大小信息,最后由绘图模块完成从模型到图像的绘制。

C、C++、Java 等典型的编程语言,执行代码前必须先编译,而 JavaScript 只需直接在网页中编写代码,再在浏览中加载网页,浏览器 JavaScript 引擎就能直接执行代码。运行时,JavaScript 不会修改服务器端的 HTML 文件,而是操控浏览器基于 HTML 文件构建的 DOM 树。