

- 智慧职教学习平台
- 微课
- 教学课件
- 教学设计
- 案例素材

扫描二维码
了解本书的
配套资源



JavaScript 与 jQuery 案例教程

► 主编 郑丽萍 刘万辉

高等教育出版社

JavaScript 与 jQuery 案例教程

► 主编 郑丽萍 刘万辉

内容提要

本书内容涵盖 JavaScript 基本语法、BOM 编程、DOM 编程、事件、原生 Ajax 实现、HTML5 相关 API、本地存储、jQuery 基础、jQuery 中的 DOM 操作、jQuery 制作动画、jQuery 插件、jQuery 与 Ajax、jQuery UI 布局等客户端交互特效制作技术。本书内容讲解由浅入深，每章内容都与案例紧密结合，有助于读者理解和应用知识、学以致用，大大加强读者实践动手操作的能力。所选案例具有极强的扩展性，能够给读者以启发，使读者举一反三，实现非常实用的效果。

本书为新形态一体化教材，与本书配套的数字课程将在“智慧职教”（www.icve.com.cn）网站上线，学习者可以登录网站进行学习，也可以通过扫描书中二维码观看教学视频，详见“智慧职教使用指南”。

本书可作为高职高专计算机相关专业的教学用书，也可作为相关领域的培训教材或参考用书。

图书在版编目（CIP）数据

JavaScript 与 jQuery 案例教程 / 郑丽萍, 刘万辉主
编. — 北京：高等教育出版社，2018.5
ISBN 978-7-04-049510-2

I. ①J… II. ①郑… ②刘… III. ①JAVA 语言-程序设计-高等职业教育-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字（2018）第 042930 号

策划编辑 吴鸣飞
插图绘制 杜晓丹

责任编辑 吴鸣飞
责任校对 陈 杨

封面设计 赵 阳
责任印制 尤 静

版式设计 马敬茹

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 涿州市星河印刷有限公司
开 本 787mm × 1092mm 1/16
印 张 22.75
字 数 410 千字
购书热线 010-58581118
咨询电话 400-810-0598

网 址	http://www.hep.edu.cn
	http://www.hep.com.cn
网上订购	http://www.hepmall.com.cn
	http://www.hepmall.com
	http://www.hepmall.cn
版 次	2018 年 5 月第 1 版
印 次	2018 年 5 月第 1 次印刷
定 价	45.30 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 49510-00

智慧职教服务指南

基于“智慧职教”开发和应用的新形态一体化教材，素材丰富、资源立体，教师在备课中不断创造，学生在学习过程中享受过程，新旧媒体的融合生动演绎了教学内容，线上线下的平台支撑创新了教学方法，可完美打造优化教学流程、提高教学效果的“智慧课堂”。

“智慧职教”是由高等教育出版社建设和运营的职业教育数字教学资源共建共享平台和在线教学服务平台，包括职业教育数字化学习中心（www.icve.com.cn）、职教云（zjy.icve.com.cn）和云课堂（APP）三个组件。其中：

- 职业教育数字化学习中心为学习者提供了包括“职业教育专业教学资源库”项目建设成果在内的大规模在线开放课程的展示学习。

- 职教云实现学习中心资源的共享，可构建适合学校和班级的小规模专属在线课程（SPOC）教学平台。

- 云课堂是对职教云的教学应用，可开展混合式教学，是以课堂互动性、参与感为重点贯穿课前、课中、课后的移动学习 APP 工具。

“智慧课堂”具体实现路径如下：

1. 基本教学资源的便捷获取

职业教育数字化学习中心为教师提供了丰富的数字化课程教学资源，包括与本书配套的电子课件（PPT）、微课、动画、教学案例、实验视频、习题及答案等。未在 www.icve.com.cn 网站注册的用户，请先注册。用户登录后，在首页或“课程”频道搜索本书对应课程“JavaScript 与 jQuery 案例教程”，即可进入课程进行在线学习或资源下载。

2. 个性化 SPOC 的重构

教师若想开通职教云 SPOC 空间，可将院校名称、姓名、院系、手机号码、课程信息、书号等发至 1548103297@qq.com（邮件标题格式：课程名+学校+姓名+SPOC 申请），审核通过后，即可开通专属云空间。教师可根据本校的教学需求，通过示范课程调用及个性化改造，快捷构建自己的 SPOC，也可灵活调用资源库资源和自有资源新建课程。

3. 云课堂 APP 的移动应用

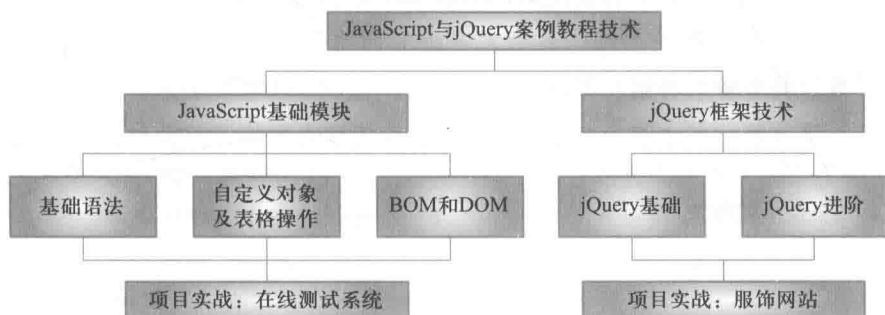
云课堂 APP 无缝对接职教云，是“互联网+”时代的课堂互动教学工具，支持无线投屏、手势签到、随堂测验、课堂提问、讨论答疑、头脑风暴、电子白板、课业分享等，帮助激活课堂，教学相长。

前言

本书全面、系统地介绍了 JavaScript 和 jQuery 知识，以创建出有趣、更富交互性、对用户更友好的 Web 网站。

本书为了让读者更快地上手，特别设计了适合初学者的学习方式，用准确的语言总结概念，用直观的图示演示效果，用详细的注释解释代码。知识点的介绍一般放在每节开始位置，以使零基础的读者了解相关概念，顺利入门。

本书内容设计框架如下图：



本书前 6 章讲解如何将 JavaScript 应用于浏览器中，后 3 章讲解 jQuery，以使得编写脚本的过程更快、更容易。其中，前 3 章介绍 JavaScript 的含义，以及 JavaScript 的基础语法。详细介绍了变量、函数、数据类型、流程控制语句等结构化程序设计，利用 JavaScript 的内置对象，如数组、日期和字符串等来管理复杂的数据，简化程序的设计，脚本化表单和其他控件，创建专业水准的 Web 应用程序。第 4 章自定义对象及表格操作，介绍了自定义对象的定义和创建；动态生成表格的行和单元格，使用复选框动态地选中多行并删除多行信息，界面美化及焦点行高亮显示等功能实现学生成绩管理案例。第 5 章 BOM 和 DOM，介绍了 BOM 和 DOM 的常用属性和方法的应用，包括元素的获取、增加、删除、修改、替换和遍历，以及事件的触发和处理、DOM 的优化和拓展，使用新的 HTML5 相关 API 等。第 6 章项目实战：JavaScript 实现在线测试系统，以一个综合项目案例介绍页面布局，用户登录及密码修改界面的设计、居中、遮罩锁屏等效果；表单的严谨验证，并给予友好的提示，原生 Ajax 应用实现远程验证，动态生成表格和样式，实现表格隔行换色的效果，实现随机抽题并显示题目信息；提交后计算总分，显示总成绩和正确答案等功能。第 7 章 jQuery 基础，介绍了常用选择器的应用，jQuery 实现 DOM 元素的查找、过滤、属性操作，节点的添加、删除、复制和替换，事件的处

理。第8章 jQuery 进阶，详细介绍了 jQuery 动画的制作，jQuery 所支持的针对服务器发起 Ajax 请求的各种方法及 jQuery 插件的开发和使用。第9章项目实战：jQuery UI 实现服饰网站，以一个综合的项目案例实现服饰网站的制作，介绍 jQuery UI 库的用法。创建客户端用户界面 (UI) 曾是一项非常烦琐的任务，而如果使用 jQuery UI 库，这些任务则变得出奇简单。即使没有丰富 JavaScript 编程经验的普通开发者，也可以用 jQuery UI 库创建出专业的用户界面。jQuery UI 库包括对话框、选项卡、accordion 和日期选择器等小组件；本书使用 jQuery UI 插件实现了服饰网站的按钮美化，菜单、主体选项卡制作；登录注册模态窗口界面实现；jQuery Ajax 实现远程登录功能等。

通过本书的学习，读者可掌握使用 JavaScript 及 jQuery 框架进行 Internet 客户端应用程序开发的知识，达到增强处理 HTML 页面显示效果、增加页面操作方便性的目的，跟踪行业新技术。本书以培养职业能力为核心，以工作实践为主线，以项目为导向，采用案例式教学，兼顾界面布局样式与交互性，增加课程内容的视觉冲击力。

阅读本书之前，需要了解如何使用 HTML 和 CSS 来构建 Web 页面。

本书由郑丽萍、刘万辉任主编。参加本书编写的有郑丽萍、刘万辉、管曙亮、黄丽萍、王志勃、陆彩霞、张趁香、冯钢、郜继红、殷鹏岚。

由于编者水平有限，书中难免存在不妥之处，敬请读者原谅，并提出宝贵意见。

编 者

2018年2月

目 录

第1章 JavaScript 初探	1	【技能目标】	23
学习目标	1	任务描述：实现猜数字游戏	24
【知识目标】	1	知识准备	26
【技能目标】	1	2.1 数据类型	26
任务描述：实现商品详情展示	2	2.1.1 数值型	26
知识准备	2	2.1.2 字符串型	27
1.1 JavaScript 简述	2	2.1.3 布尔型	28
1.1.1 JavaScript 简史	2	2.1.4 特殊数据类型	28
1.1.2 JavaScript 引擎的工作		2.1.5 数据类型的自动转换	30
原理	3	2.2 变量	30
1.1.3 JavaScript 代码执行		2.2.1 关键字	30
顺序	4	2.2.2 变量的定义与命名	31
1.1.4 JavaScript 的组成	5	2.2.3 变量的声明与赋值	31
1.1.5 JavaScript 的主要特点	5	2.3 函数	32
1.1.6 JavaScript 相关应用	6	2.3.1 函数的定义与调用	32
1.2 JavaScript 编程起步	8	2.3.2 带有返回值的函数	35
1.2.1 选择 JavaScript 脚本		2.3.3 变量的作用域	37
编辑器	8	2.4 内置函数	38
1.2.2 引入 JavaScript 脚本代码		2.5 运算符与表达式	40
到 HTML 文档中的方法	12	2.5.1 表达式	40
1.2.3 常用输出语句	16	2.5.2 运算符	40
任务实施	18	2.6 基本语句	46
任务训练	20	2.6.1 注释语句	46
【理论测试】	20	2.6.2 流程控制语句	46
【实训内容】	21	2.6.3 异常处理语句	52
第2章 JavaScript 语言基础	23	2.6.4 prompt() 方法	53
学习目标	23	任务实施	54
【知识目标】	23	任务训练	56
		【理论测试】	56

【实训内容】	57	3.4.7 字符串对象应用于表单	81
【实训拓展】	57	3.5 JavaScript 正则表达式	85
第3章 常用内置对象	59	3.5.1 正则表达式	85
学习目标	59	3.5.2 创建正则表达式	85
【知识目标】	59	3.5.3 正则表达式对象的方法	86
【技能目标】	59	3.5.4 正则表达式中的常用 符号	87
任务描述：实现简洁美观的计算器	60	3.6 拓展：用户注册的实现	90
知识准备	60	任务实施	96
3.1 数组（Array）对象	60	任务训练	103
3.1.1 新建数组	60	【理论测试】	103
3.1.2 动态数组	61	【实训内容】	104
3.1.3 数组对象的常用属性与 方法	61	【实训拓展】	105
3.1.4 二维数组	64	第4章 自定义对象及表格操作	107
3.2 日期（Date）对象	66	学习目标	107
3.2.1 新建日期	66	【知识目标】	107
3.2.2 日期对象的属性与方法	67	【技能目标】	107
3.2.3 定时器函数	69	任务描述：学生成绩管理	108
3.2.4 动态改变元素样式	69	知识准备	108
3.2.5 日期对象应用案例	70	4.1 JavaScript 自定义对象	108
3.3 数学（Math）对象	73	4.1.1 JavaScript 对象	108
3.3.1 使用数学对象	73	4.1.2 JavaScript 自定义对象 建立方案	109
3.3.2 数学对象的属性与方法	74	4.1.3 学生对象的建立	114
3.3.3 数字的格式化与产生 随机数	75	4.1.4 使用表单动态创建 学生对象	115
3.3.4 数学对象应用案例	75	4.1.5 遍历数组输出展示学生 对象信息	118
3.4 字符串（String）对象	77	4.2 动态表格	119
3.4.1 使用字符串对象	77	4.2.1 表格建立	119
3.4.2 字符串相加	77	4.2.2 动态插入行和单元格	119
3.4.3 在字符串中使用单引号、 双引号及其他特殊字符	78	4.2.3 动态删除某行	123
3.4.4 比较字符串是否相等	78	4.2.4 动态选中多行并删除	124
3.4.5 字符串对象的属性与 方法	79	任务实施	131
3.4.6 字符串对象应用案例	80		

6.5.2	用户登录的样式设置	200
6.5.3	JavaScript 实现用户登录 居中显示效果	202
6.5.4	JavaScript 实现用户登录 验证功能	202
6.6	在线测试系统：用户密码修改 的实现	204
6.6.1	用户密码修改的内容结构 描述	204
6.6.2	用户密码修改的样式 设置	204
6.6.3	JavaScript 实现用户密码 修改居中和锁屏功能	206
6.7	JavaScript 实现原生 Ajax 应用	207
6.7.1	Ajax 简介	207
6.7.2	原生 JavaScript 实现 Ajax 的步骤	208
6.7.3	Ajax 实现在线测试系统 远程验证—登录功能	215
6.7.4	JavaScript 实现用户按 角色登录功能	216
6.7.5	拓展：数据库操作实现 ..	216
6.8	在线测试系统：在线测试功能 模块的实现	218
6.8.1	在线测试功能模块的内容 结构描述	219
6.8.2	在线测试功能模块的样式 设置	219
6.8.3	JavaScript 实现在线测试 功能	219
6.8.4	JavaScript 实现在线测试 功能：页脚模块的 实现	223

任务训练	223
【理论测试】	223

【实训内容】	224
【实训拓展】	224

第7章 jQuery 基础 225

学习目标	225
------------	-----

【知识目标】	225
--------------	-----

【技能目标】	225
--------------	-----

任务描述：实现文字和图片提示

效果	226
----------	-----

知识准备	226
------------	-----

7.1 jQuery 简介 226

7.1.1 jQuery 环境配置	227
-------------------------	-----

7.1.2 jQuery 应用的创建	229
--------------------------	-----

7.1.3 jQuery 对象与 DOM 对象的 转换	232
--------------------------------------	-----

7.2 jQuery 常用选择器的应用 233

7.2.1 基本选择器	233
-------------------	-----

7.2.2 层次选择器	235
-------------------	-----

7.2.3 过滤选择器	241
-------------------	-----

7.3 jQuery 控制页面 250

7.3.1 DOM 元素的遍历	250
-----------------------	-----

7.3.2 DOM 元素属性操作及 节点添加	254
---------------------------------	-----

7.3.3 DOM 节点删除、复制 和替换	258
--------------------------------	-----

7.4 jQuery 事件 261

7.4.1 事件绑定与解除	263
---------------------	-----

7.4.2 复合事件	266
------------------	-----

任务实施	267
------------	-----

任务训练	270
------------	-----

【理论测试】	270
--------------	-----

【实训内容】	271
--------------	-----

第8章 jQuery 进阶 273

学习目标	273
------------	-----

【知识目标】	273	知识准备	319
【技能目标】	273	9.1 jQuery UI 插件	319
任务描述: jQuery 实现本地验证和		9.1.1 jQuery UI 简介	319
远程验证	274	9.1.2 jQuery UI 使用:	
8.1 jQuery 动画	275	datepicker	320
8.1.1 元素的显示与隐藏	275	9.1.3 jQuery UI 使用:	
8.1.2 回调函数的使用	278	dialog	322
8.1.3 自定义动画	280	9.2 服饰网站: 页眉的设计与	
8.2 jQuery 与 Ajax	289	实现	326
8.2.1 Ajax 简介	289	9.2.1 页眉元素及 CSS 样式	
8.2.2 load() 方法	290	代码设计	326
8.2.3 \$.get() 和 \$.post()		9.2.2 jQuery UI 使用:	
方法	293	button	328
8.2.4 \$.ajax() 方法	294	9.3 服饰网站: 页面主体部分的	
8.2.5 表单序列化	296	设计与实现	329
8.2.6 \$.getScript() 和		9.3.1 页面主体部分设计	329
\$.getJSON() 方法	297	9.3.2 菜单的设计	329
8.3 jQuery 插件	301	9.3.3 主体选项卡的设计	333
8.3.1 插件简介	301	9.3.4 页脚模块的实现	339
8.3.2 开发 plugin	301	9.4 服饰网站: 商品详情页面的	
8.3.3 第三方插件的使用	306	设计与实现	340
任务实施	311	9.4.1 详情页面页眉设计	340
任务训练	315	9.4.2 详情页面主体的内容	
【理论测试】	315	结构描述	340
【实训内容】	316	9.4.3 详情页面主体的样式	
【实训拓展】	316	设置	343
第9章 项目实战: jQuery UI		9.4.4 详情页面主体的功能	
实现服饰网站	317	实现	345
学习目标	317	任务训练	347
【知识目标】	317	【实训内容】	347
【技能目标】	317	【实训拓展】	348
任务描述: 实现服饰网站	318	参考文献	349

第 1 章

JavaScript 初探



学习目标

【知识目标】

- 了解 JavaScript 简史。
- 了解 JavaScript 引擎的工作原理。
- 掌握 JavaScript 的组成。
- 掌握 JavaScript 的主要特点和相关应用。
- 巩固学习 HTML 和 CSS 的使用方法。

【技能目标】

- 能够使用 JavaScript 脚本编辑器。
- 能够改变 JavaScript 代码的执行顺序。
- 能够使用多种方案引入 JavaScript 脚本代码到 HTML 文档中。
- 能够使用常用的输出语句。
- 能够获取元素并改变元素的内容。



任务描述：实现商品详情展示

作为初学者，体验 JavaScript 的编写方法与技巧，实现文字切换效果，体验浏览商品简要信息，如图 1-1 所示。当单击图片后，显示商品的详细信息，如图 1-2 所示。



图 1-1 单击图片前的页面效果



图 1-2 单击图片后的页面效果



微课：1-1
JavaScript 简介



知识准备

1.1 JavaScript 简述

1.1.1 JavaScript 简史

ECMAScript 是实现 JavaScript 所依据的标准，也用来描述语言版本。（例

如, ECMAScript5) 是一种由欧洲计算机制造商协会 (ECMA) 通过的脚本程序设计标准。

1997 年, 为了解决 JavaScript 各版本间语法、特性等方面的混乱, 推出了 ECMA-262 规范版本。ECMA-262 标准 Edition 3 提供了对字符串处理、错误定义和数值输出等方面的更新, 同时增加了对 try...catch 异常处理、正则表达式、新的控制语句等方面的完美支持。它标志着 ECMAScript 成为了一门真正的编程语言, 以 ECMAScript 为核心的 JavaScript 脚本语言得到了迅猛的发展。JavaScript 3.0 是一个巨大的成功, 在业界得到了广泛支持, 成为通行标准, 奠定了 JavaScript 语言的基本语法, 以后的版本是对其的完全继承。直到今天, 初学者一开始学习 JavaScript, 就是在学习 JavaScript 3.0 的语法。

2009 年 12 月, ECMAScript 5 版正式发布。ECMAScript 5 与 ECMAScript 3 基本保持兼容, ECMAScript 5 支持 Opera 11.60、Internet Explorer 9 * (IE9)、Firefox 4、Safari 5.1 *、Chrome 13 等浏览器, 因此在 PC 端开发时, 要注意 IE 9 以下版本浏览器的兼容性, 在移动端开发时, 就可以忽略这个问题。

2015 年 6 月通过的 ECMAScript 6, 是 JavaScript 语言的下一代标准, 其目标是使得 JavaScript 语言可以用来编写复杂的大型应用程序, 成为企业级开发语言。基本上没有浏览器能够完全支持, 一般不直接使用。若需要, 由转换器转换为被浏览器接受的 ECMAScript 5。

1.1.2 JavaScript 引擎的工作原理

JavaScript 解析引擎能够“读懂”JavaScript 代码, 并准确地给出代码运行结果。例如, 当编写了“var a = 1+1;”这样一段代码, JavaScript 引擎做的事情是看懂(解析)这段代码, 并将 a 的值变为 2。JavaScript 引擎就是直接解析并将代码运行结果输出的 JavaScript 的解释器。图 1-3 所示为 JavaScript 的执行过程。

JavaScript 引擎根据规则解析 JavaScript 代码。ECMAScript 定义了这些规则。标准的 JavaScript 引擎就会根据这套文档去实现。注意, 这里强调了标准, 因为也有不按照标准来实现的, 如 IE 的 JavaScript 引擎。这也是为什么 JavaScript 会有兼容性问题的原因。

JavaScript 引擎是浏览器的组成部分之一。浏览器同时还要做很多别的事情, 如解析页面、渲染页面、cookie 管理、历史记录等。一般情况下, JavaScript 引擎(如 IE 9 的 Chakra、Firefox 的 TraceMonkey、Chrome 的 V8 等)都是浏览器开发商自行开发的。从而也可看出, 不同浏览器都采用了不同的 JavaScript 引擎。

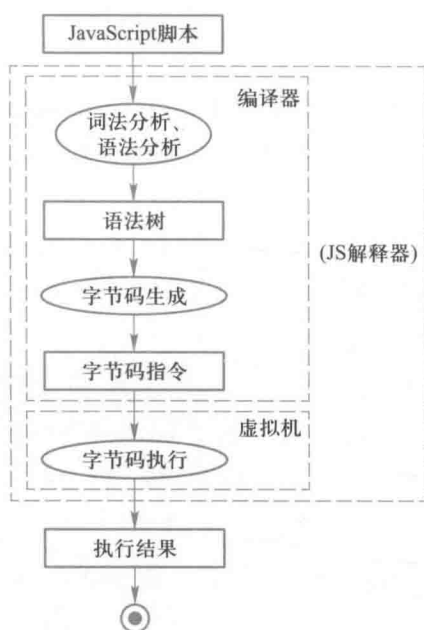


图 1-3 JavaScript 的执行过程

1.1.3 JavaScript 代码执行顺序

HTML 文档在浏览器中的解析过程是按着文档流从上到下逐步解析页面结构和信息的。JavaScript 代码作为嵌入的脚本应该也算 HTML 文档的组成部分，JavaScript 代码在装载时的执行顺序也是根据脚本标签<script>的出现顺序来确定的。如果通过脚本标签<script>的 src 属性导入外部 JavaScript 文件脚本，那么它也将按照其语句出现的顺序来执行，不会因为外部 JavaScript 文件而延期执行。

为了安全起见，一般在页面初始化完毕之后才允许 JavaScript 代码执行，这样可以避免网速对 JavaScript 执行的影响，同时也避开了 HTML 文档流对 JavaScript 执行的限制。

如果在一个页面中存在多个 window.onload 事件处理函数，则只有最后一个才是有效的。为解决该问题，可以把所有脚本或调用函数都放在同一个 onload 事件处理函数中。例如：

```
window.onload=function(){  
    f1();  
    f2();  
    f3();  
}
```

而且通过这种方式可以改变函数的执行顺序，方法是简单地调整 onload 事件处理函数中调用函数的排列顺序。

除了页面初始化事件外，还可以通过各种交互事件（如鼠标事件、键盘事件及时钟触发器等方法）来改变 JavaScript 代码的执行顺序。

1.1.4 JavaScript 的组成

JavaScript 由以下 3 部分组成。

- 核心（ECMAScript）描述了该语言的语法和基本对象。
- 浏览器对象模型（Browser Object Model, BOM）描述了与浏览器进行交互的方法和接口。
- 文档对象模型（Document Object Model, DOM）描述了处理网页内容的方法和接口。

BOM 是浏览器对象模型，用来获取或设置浏览器的属性、行为，如新建窗口、调整窗口大小、关闭窗口、浏览历史记录等。DOM 是文档对象模型，定义了 JavaScript 操作 HTML 文档的接口，提供了访问 HTML 文档（如 body、form、div、textarea 等）的途径以及操作方法，可以用来获取或设置文档中标签的属性，如获取或者设置 input 表单元素的 value 值。

浏览器载入 HTML 文档后，将整个文档规划成由节点构成的节点树，文档中每个部分都是一个节点。例如，`<div id="div1" class="div1">DOM 示例</div>`，其中`<div>`标签是元素节点，“id”和“class”是属性节点，“DOM 示例”是文本节点。

1.1.5 JavaScript 的主要特点

JavaScript 是一种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，主要用于创建具有交互性较强的动态页面。其具有以下主要特点。

1. 解释型脚本语言

JavaScript 是一种解释型脚本语言，在嵌入 JavaScript 脚本的 HTML 文档载入时被浏览器逐行解释，可大量节省客户端与服务器端进行数据交互的时间。

2. 基于对象的语言

JavaScript 是一种基于对象的语言，提供了大量的内置对象，如 String、Number、Boolean 等，它还具有一些面向对象的基本特征，可以根据需要创建自己的对象，从而进一步扩大其应用范围，编写功能强大的 Web 文档。

3. 简单性

JavaScript 的基本结构类似 C 语言。它采用小程序段的方式编程，并提供

了简易的开发平台和便捷的开发流程，因此可以嵌入到 HTML 文档中供浏览器解释执行。同时，JavaScript 的变量类型是弱类型，不强制检查变量的类型，即可以不定义其变量的类型。

4. 相对安全性

JavaScript 是一种相对安全性语言。它不允许访问本地硬盘，也不能将数据存入服务器，不允许对网络文档进行修改和删除，而只能通过浏览器实现信息浏览或动态交互，从而能有效地防止数据被篡改。

5. 动态性

JavaScript 是动态的。它可以直接对用户或客户的输入作出响应，无须经过 Web 服务程序。它对用户的响应，是采用事件驱动的方式进行的。所谓事件就是指在网页中执行了某种操作所产生的动作。例如，按下鼠标、移动窗口、选择菜单等都可以被视为事件。当事件发生后，可能会引起相应的事件响应，即事件驱动。

6. 跨平台性

JavaScript 依赖于浏览器本身，与操作系统环境无关，只要操作系统能运行浏览器并且浏览器支持 JavaScript，就可以正确执行。

综上所述，JavaScript 是一种有较强生命力和发展潜力的脚本描述语言，它可以被直接嵌入到 HTML 文档中，供浏览器解释执行，直接响应客户端事件，如验证数据表单合法性，并调用相应的处理方法，迅速返回处理结果并更新页面，实现 Web 交互性和动态的要求，同时将大部分的工作交给客户端处理，将 Web 服务器的资源消耗降到最低。HTML 定义网页的内容；CSS 描述网页的布局；JavaScript 控制网页的行为。本书介绍 JavaScript 如何与 HTML 和 CSS 一起工作。

1.1.6 JavaScript 相关应用

JavaScript 的功能十分强大，可实现多种任务。JavaScript 可用来在数据被送往服务器前对表单输入的数据进行验证，实现如执行计算、检查表单、编写游戏、添加特殊效果、自定义图形选择、创建安全密码等。所有这些功能都有助于增强站点的动态效果和交互性。以下是几种常见的 JavaScript 应用。

1. 验证数据

使用 JavaScript 可以创建动态 HTML 页面，以使用特殊对象、文件和相关数据库来处理用户输入和维护永久性数据。例如，在某个网站注册时必须填写一份表单，输入各种详细信息，如果某个字段输入有误，向 Web 服务器提交表单前，经客户端验证发现错误，提示警告信息，如图 1-4 所示。



微课：1-2
JavaScript 相关应用