

国家骨干高职院校建设成果·计算机项目化系列教材

JavaScript与jQuery 实战教程

卢淑萍 樊红珍 主 编
徐红波 王涛 黄伟民 副主编



BOOKASK.COM

清华大学出版社
北 京

内 容 简 介

本书系统全面地介绍了 JavaScript 和 jQuery 知识,内容涵盖 JavaScript 基本语法、BOM 编程、DOM 编程、事件、jQuery 基础、jQuery 中的 DOM 操作、jQuery 制作动画、jQuery 插件及 jQuery 与 Ajax 等客户端交互特效制作技术。本书由浅入深,辅以大量的实例说明,操作性和实用性较强。

本书适合作为各大专院校相关专业的网页特效设计课程的教材,也可以作为网页特效设计的培训教材和网站开发的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

JavaScript 与 jQuery 实战教程/卢淑萍,樊红珍主编.--北京:清华大学出版社,2015

国家骨干高职院校建设成果.计算机项目化系列教材

ISBN 978-7-302-38330-7

I. ①J… II. ①卢… ②樊… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 240789 号

责任编辑:刘翰鹏

封面设计:

责任校对:袁 芳

责任印制:

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795764

印 刷 者:

装 订 者:

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.5

字 数:519 千字

版 次:2015 年 2 月第 1 版

印 次:2015 年 2 月第 1 次印刷

印 数:1~ 000

定 价: .00 元

产品编号:058037-01



Web 前端技术发展迅速,主流技术日新月异,HTML、CSS 和 JavaScript 三者共同构成了丰富多彩的网页,它们使网页包含更多活跃的元素和更加精彩的内容。HTML 是一种超文本标记语言,它定义了网页结构,决定了网页内容。CSS 实现了网页结构与表现样式完全分离。JavaScript 则主要实现实时、动态、交互性效果,对用户的操作进行响应,使页面更加实用、友好、人性化,是目前运用最广泛的行为标准语言。jQuery 是一个优秀的 JavaScript 框架,凭借简洁的语法让开发者轻松实现很多以往需要大量 JavaScript 开发才能完成的功能和特效,并对 CSS、DOM、Ajax 等各种标准 Web 技术提供了许多实用而简便的方法,同时很好地解决了浏览器之间的兼容性问题。

本书以实际网站中流行的网页特效为载体,强化 Web 前端工程师所需要掌握的技能,提升动手能力,是一本应用当前流行前端技术实现客户端特效的实用教程,针对“Web 前端工程师”能力,以工作任务为核心重新选择和组织专业知识体系,按照工作过程设计学习情境,是一本体现工学结合思想的教材。与其他同类教材相比,本书具有以下特点。

- ◆ 突出客户端网页特效制作能力的培养。本书按照工学结合教材的编写思路,精心设计各教学环节,让读者在反复动手实践中学会应用所学知识解决实际问题。
- ◆ 教学内容完全根据真实任务来确定。选取的教学内容适用于设计与制作小型的动态网站,即制作包含客户端验证、常见动态效果、界面美观大方的网站,选取的教学内容也是作为大型网站规划与建设的基础。
- ◆ 本书内容由浅入深,并辅以大量的实例说明,操作性和实用性较强。
- ◆ 充分考虑学生认知规律,化解知识难点。本书面向实际应用组织教材内容,通过实例进行讲解、分析。

本书配套教学资源丰富,包括教学课件(PPT)、教学案例、操作视频、案例素材、教案、授课计划、拓展资源和课后练习答案等教学资源,方便教师教学和学生课后练习。读者可登录(<http://61.145.231.44:8080/skills/solver/classView.do?classKey=5542365>)网站获取。为了简化篇幅,书中代码只截取主要部分,完整源代码请见配套教学资源。

本书作者既有高校教学经验丰富的“双师型教师”,又有企业一线工程师。其中卢淑萍老师编写了第1~3章,并对全书进行了统稿;樊红珍老师编写了第8~10章;徐红波老师编写了第4、7章;王涛老师编写了第5、6章;企业工程师黄伟民编写了第11、12章。清华大学出版社编辑对本书结构给予了悉心指导,并提出了宝贵意见,在此表示深深的敬意和谢意。

由于编者水平有限,书中难免存在疏漏之处,敬请各位专家和读者批评、指正。

编者邮箱: 58613802@qq.com.

编 者

2014 年 11 月



BOOKASK.COM



第 1 章 JavaScript 概述	001
任务 1.1 认识 JavaScript	001
1.1.1 JavaScript 的起源	001
1.1.2 JavaScript 的作用	002
1.1.3 JavaScript 的基本特点	004
1.1.4 JavaScript 的组成	005
任务 1.2 在页面显示个人信息	005
1.2.1 选择 JavaScript 脚本编辑器	006
1.2.2 脚本的基本结构	006
1.2.3 在网页中引用 JavaScript 的方式	007
1.2.4 常用的输入/输出语句	008
1.2.5 任务实现	011
任务 1.3 测试综合案例在主流浏览器上的兼容性	011
1.3.1 浏览器之间的冲突	012
1.3.2 标准的制定	012
1.3.3 任务实现	012
小结	013
实训	013
课后练习	016
第 2 章 JavaScript 基础	019
任务 2.1 显示变量数据类型	019
2.1.1 变量	019
2.1.2 数据类型	020
2.1.3 运算符	023
2.1.4 任务实现	025
任务 2.2 根据成绩给出学生考评	026
2.2.1 if 语句	026
2.2.2 switch 语句	027
2.2.3 任务实现	028
任务 2.3 实现猜数游戏	029
2.3.1 for 循环	030



2.3.2	while 循环	031
2.3.3	do...while 循环	031
2.3.4	for...in 循环	031
2.3.5	中断循环语句	031
2.3.6	任务实现	032
任务 2.4	制作简易计算器	032
2.4.1	数值判断函数	033
2.4.2	自定义函数	033
2.4.3	任务实现	036
小结	037
实训	037
课后练习	040
第 3 章	JavaScript 对象编程	043
任务 3.1	在页面动态显示系统时间	043
3.1.1	Date 对象的创建	043
3.1.2	Date 对象的常用方法	044
3.1.3	定时器函数	046
3.1.4	任务实现	046
任务 3.2	制作随机选号页面	047
3.2.1	Math 对象的常用属性	048
3.2.2	Math 对象的常用方法	049
3.2.3	任务实现	049
任务 3.3	制作简单的焦点图效果	050
3.3.1	数组的创建	051
3.3.2	数组的访问	052
3.3.3	数组的常用属性和方法	052
3.3.4	任务实现	054
任务 3.4	验证注册页面信息	057
3.4.1	String 对象的创建	058
3.4.2	String 对象的常用属性和方法	058
3.4.3	任务实现	060
任务 3.5	使用正则表达式验证注册页面信息	065
3.5.1	定义正则表达式	066
3.5.2	正则表达式的操作方法	067
3.5.3	正则表达式的模式	069
3.5.4	任务实现	070
小结	073
实训	073
课后练习	079



第 4 章 BOM 编程	082
任务 4.1 制作弹出窗口特效	082
4.1.1 顶级对象模型参考	083
4.1.2 window 的常用属性	083
4.1.3 window 的常用方法	084
4.1.4 window 的常用事件	086
4.1.5 任务实现	086
任务 4.2 制作网页标题交替显示效果	088
4.2.1 document 对象的常用属性	088
4.2.2 任务实现	089
任务 4.3 制作复选框全选效果	090
4.3.1 document 对象的常用方法	091
4.4.2 任务实现	093
小结	094
实训	094
课后练习	099
第 5 章 DOM 编程	101
任务 5.1 画出页面的文档节点结构图	101
5.1.1 什么是 DOM	102
5.1.2 DOM 的组成	102
5.1.3 DOM 模型中的节点	102
5.1.4 任务实现	102
任务 5.2 使用 Core DOM 方式动态添加表格	103
5.2.1 节点的常用属性和方法	104
5.2.2 查看节点	104
5.2.3 创建和增加节点	108
5.2.4 删除和替换节点	110
5.2.5 任务实现	112
任务 5.3 使用 HTML DOM 方式动态添加表格	113
5.3.1 HTML DOM 对象	114
5.3.2 访问 HTML DOM 对象的属性	115
5.3.3 表格对象	115
5.3.4 任务实现	117
任务 5.4 拓展	118
5.4.1 用 DOM 的文档碎片提高页面的执行效率	118
5.4.2 在特定节点后插入节点	120
小结	121
实训	121
课后练习	125



第 6 章 JavaScript 与 CSS 交互	129
任务 6.1 使用 JavaScript 动态改变导航菜单样式	129
6.1.1 style 属性	130
6.1.2 className 属性	133
6.1.3 任务实现	134
任务 6.2 制作图片轮显效果	136
6.2.1 visibility 属性	137
6.2.2 display 属性	137
6.2.3 任务实现	139
任务 6.3 制作随光标滚动的广告图片	141
6.3.1 获取样式属性值	142
6.3.2 页面事件	144
6.3.3 任务实现	144
任务 6.4 拓展	146
6.4.1 制作 Tab 切换效果	146
6.4.2 制作光标移到某图片上显示该图片对应大图	147
小结	148
实训	149
课后练习	152
第 7 章 jQuery 基础	157
任务 7.1 使用 jQuery 在警告框中显示“Hello World!”	157
7.1.1 jQuery 的优势	158
7.1.2 配置 jQuery 环境	158
7.1.3 jQuery 开发工具和插件	159
7.1.4 任务实现	160
任务 7.2 网站品牌列表的显示与收起	161
7.2.1 基本选择器	162
7.2.2 层次选择器	166
7.2.3 过滤选择器	167
7.2.4 任务实现	179
任务 7.3 拓展	181
7.3.1 jQuery 对象与 DOM 对象	181
7.3.2 表单选择器	183
小结	185
实训	185
课后练习	188
第 8 章 jQuery 中的 DOM 操作	191
任务 8.1 制作图片展示效果	191
8.1.1 获取和设置属性	192



8.1.2 删除属性	192
8.1.3 任务实现	192
任务 8.2 制作留言板前端局部更新效果	194
8.2.1 查找节点	194
8.2.2 创建节点	195
8.2.3 插入节点	196
8.2.4 任务实现	197
任务 8.3 制作电子邮件删除效果	199
8.3.1 删除节点	200
8.3.2 复制节点	202
8.3.3 替换节点	202
8.3.4 遍历节点	202
8.3.5 任务实现	203
任务 8.4 表格隔行变色和当前行变色	204
8.4.1 获取和设置 style 对象的属性	205
8.4.2 获取和设置 class 样式	206
8.4.3 追加和移除样式	206
8.4.4 判断是否含有某个样式	207
8.4.5 任务实现	208
任务 8.5 拓展	209
8.5.1 制作导航栏菜单效果	209
8.5.2 制作层叠图片横向平滑移动轮播切换效果	210
小结	214
实训	214
课后练习	218
第 9 章 jQuery 中的事件	221
任务 9.1 微博字数统计	221
9.1.1 加载 DOM	222
9.1.2 事件绑定	224
9.1.3 事件移除	225
9.1.4 简写事件	225
9.1.5 任务实现	227
任务 9.2 制作图片提示效果	228
9.2.1 事件对象	229
9.2.2 事件冒泡	229
9.2.3 任务实现	231
任务 9.3 改变图片透明度	232
9.3.1 合成事件 hover	233
9.3.2 特殊事件 one	233



9.3.3 任务实现	234
任务 9.4 拓展	234
9.4.1 模拟操作 trigger() 方法	234
9.4.2 事件命名空间	235
小结	236
实训	236
课后练习	240
第 10 章 jQuery 制作动画	242
任务 10.1 FAQ 答案的显示和隐藏	242
10.1.1 show() 和 hide() 方法	243
10.1.2 fadeIn() 和 fadeOut() 方法	244
10.1.3 slideUp() 和 slideDown() 方法	245
10.1.4 任务实现	245
任务 10.2 图片横向移动效果	247
10.2.1 自定义动画方法 animate()	248
10.2.2 动画回调函数	251
10.2.3 停止动画和判断是否处于动画状态	252
10.2.4 任务实现	256
任务 10.3 拓展	261
10.3.1 焦点幻灯片效果	261
10.3.2 背景动画全屏焦点图效果	263
小结	266
实训	266
课后练习	270
第 11 章 jQuery 插件应用	273
任务 11.1 使用 jqZoom 插件实现图片放大镜效果	273
11.1.1 jqZoom 插件简介	274
11.1.2 jqZoom 插件参数	274
11.1.3 任务实现	275
任务 11.2 使用 Fancybox 插件实现图片播放效果	276
11.2.1 Fancybox 插件简介	277
11.2.2 Fancybox 插件参数	277
11.2.3 任务实现	278
任务 11.3 编写表格隔行变色插件	280
11.3.1 插件的种类	280
11.3.2 插件的注意事项	281
11.3.3 jQuery 插件的机制	281
11.3.4 命名空间	282
11.3.5 编写 jQuery 插件	282



11.3.6 任务实现	285
小结	286
实训	286
课后练习	288
第 12 章 项目案例：化妆品网站	289
任务 12.1 案例分析	289
12.1.1 需求概述	289
12.1.2 开发环境	292
12.1.3 案例覆盖的技能点	292
12.1.4 开发技巧	293
任务 12.2 页面交互效果实现	294
12.2.1 首页交互效果	294
12.2.2 产品列表页面交互效果	308
12.2.3 订购页面交互效果	320
12.2.4 用户注册交互效果	325
小结	331
参考文献	332



BOOKASK.COM



BOOKASK.COM

JavaScript 概述

学习目标

- (1) 了解 JavaScript 的发展。
- (2) 了解 JavaScript 的作用及特点。
- (3) 了解 JavaScript 的组成。
- (4) 掌握脚本的基本结构。
- (5) 掌握在网页引入脚本的方式。
- (6) 了解浏览器兼容性问题。

JavaScript 是目前 Web 应用程序开发者使用最广泛的客户端脚本编程语言之一,它不仅可用来开发交互式的 Web 页面,更重要的是它将 HTML、XML、Ajax 和 Java Applet 等功能强大的 Web 对象有机结合起来,使开发人员能快速生成 Internet 或 Intranet 上使用的分布式应用程序。

任务 1.1 认识 JavaScript

1.1.1 JavaScript 的起源

20 世纪 90 年代中期,大部分互联网用户使用 28.8Kbps 的调制解调器连接到网络进行网上冲浪,为解决网页功能简单的问题,HTML 文档已经变得越来越复杂和庞大,更让用户痛苦的是,为了能验证一个表单的有效性,客户端必须与服务器端进行多次的数据交互,甚至等待几十秒之后,服务器端返回的不是“提交成功”,而是错误提示。当时业界已开始考虑开发一种客户端脚本语言,用于解决诸如表单合法性验证等简单而实用的问题。

1995 年,Netscape 公司和 Sun 公司联合开发出 JavaScript 脚本语言,并在 Netscape Navigator 2 中实现了 JavaScript 脚本规范的第一个版本,即 JavaScript 1.0 版,该脚本语言不久就显示了其强大的生机和发展潜力。当时 Netscape Navigator 主宰着 Web 浏览器市场,为了跟上 Netscape 的步伐,Microsoft 在其 Internet Explorer 3 中以 Jscript 为名,发布了一个 JavaScript 1.0 的克隆版本 Jscript 1.0。

1997 年,为了避免无序竞争,同时解决 JavaScript 几个版本中语法、特性等方面的混乱问题,JavaScript 1.1 作为一个草案被提交给欧洲计算机制作商协会(ECMA),经协商后推出了 ECMA-262 标准,其中定义了 ECMAScript 这种全新的脚本语言。然而至今各个浏览



器对 JavaScript 的支持仍未完全遵循该标准。

到了 21 世纪,网上各种广告和滚动提示条越来越多,JavaScript 被很多网页制作者乱用,直到 2005 年年初,Google 公司的网上产品使 Ajax 一时兴起,并受到广泛好评,作为 Ajax 最重要元素之一的 JavaScript 才重新找到了自己的定位。

如今 JavaScript 正朝着提高用户体验、增强网页友好性的方向发展,并越来越受到开发人员的关注,各种 JavaScript 的功能插件层出不穷,网页的功能在它的基础上更是愈加丰富多彩。

1.1.2 JavaScript 的作用

JavaScript 的诞生无疑给网页注入了新的活力,它除了实现普通的表单验证外还可以制作各种漂亮的页面特效,越来越多的网页使用了这一脚本语言。随着 Web 技术的发展和成熟,JavaScript 还被用在服务器的通信上,也就是近年来越来越火的 Ajax 技术。

1. 客户端表单验证

JavaScript 最开始出现的目的就是解决验证方面的问题,这也是 JavaScript 最基本和最重要的作用。表单验证的应用场合比较常见,例如网站中常见的会员注册,填写注册信息时,如果某项信息格式输入错误(例如密码长度位数不够、出生日期输入字符等),表单页面将及时给出错误提示。这些错误在没有提交到服务器端前,在客户端提前进行验证,称为客户端表单验证。这样,用户得到了及时的交互,同时也减轻了网站服务器端的压力。

2. 页面动态效果

在 JavaScript 中,用户可以编写响应鼠标、键盘等事件的代码,创建动态页面特效,从而高效地控制页面内容。例如幻灯片切换特效(见图 1-1)和层的显示隐藏特效(见图 1-2 与图 1-3),它们的应用增强了客户端的体验,使网站更加有动感、有魅力,吸引了更多的浏览者。



图 1-1 幻灯片切换特效



图 1-2 层的隐藏



图 1-3 层的隐藏显示特效

3. 动态改变页面内容

在实际应用中,通常需要动态地在页面中为表格添加一行、删除一行或者改变表格中的内容,这些功能经常在购物网站中使用,用户可以方便地改变购物的数量等,如图 1-4 所示。



商品名称	数量	价格	操作	
防滑真皮休闲鞋	12	¥568.50	删除	修改
抗疲劳神奇钛项圈	2	¥49.00	删除	修改
抗疲劳神奇钛项圈	3	¥49.00	删除	修改
抗疲劳神奇钛项圈	<input type="text" value="6"/>	¥49.00	删除	确定
抗疲劳神奇钛项圈	5	¥49.00	删除	修改
抗疲劳神奇钛项圈	6	¥49.00	删除	修改
抗疲劳神奇钛项圈	7	¥49.00	删除	修改
增加订单				

图 1-4 动态改变表格内容

1.1.3 JavaScript 的基本特点

JavaScript 是一种基于对象和事件驱动的客户脚本语言,并具有相对的安全性,主要用于创建交互性较强的动态页面。其主要特点如下。

(1) 基于对象: JavaScript 是基于对象的脚本编程语言,能通过 DOM(文档结构模型)及自身提供的对象和操作方法来实现所需的功能。

(2) 事件驱动: JavaScript 采用事件驱动方式,能响应键盘、鼠标及浏览器窗口事件等,并执行指定的操作。

(3) 解释性语言: JavaScript 是一种解释性脚本语言,无须专门的编译器进行编译,在嵌入 JavaScript 脚本的 HTML 文档被浏览器载入时逐行地解释,大量地节省了客户端与服务端进行数据交互的时间。

(4) 实时性: JavaScript 事件处理是实时性的,无须经服务器端即可对客户端的事件做出响应,并用处理结果实时更新目标页面。

(5) 动态性: JavaScript 提供简单高效的语言流程,灵活处理对象的各种方法和属性,同时及时响应文档页面事件,实现页面的交互性和动态性。

(6) 跨平台: JavaScript 脚本的正确运行依赖于浏览器,而与具体的操作系统无关。只要客户端装有支持 JavaScript 脚本的浏览器,JavaScript 脚本运行结果就能正确反映在客户端浏览器平台上。

(7) 开发使用简单: JavaScript 的基本结构类似于 C 语言,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,能嵌入 HTML 文档中供浏览器解释执行,同时 JavaScript 的变量类型是弱类型,使用不严格。

(8) 安全性: JavaScript 是客户脚本,通过浏览器解释执行。它不允许直接访问本地计算机,并且不能将数据存到服务器上,它也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,从而有效地防止数据的丢失。

综上所述,JavaScript 是一种有着较强生命力和发展潜力的脚本描述语言,可被直接嵌入 HTML 文档中,供浏览器解释执行;它可直接响应客户端事件,如验证数据表单合法性等,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 交互性和动态性的要求。同时将大部分的工作交给客户端处理,使 Web 服务器资源服务器消耗降到最低。



1.1.4 JavaScript 的组成

尽管 ECMAScript 是一个重要的标准,但它并不是 JavaScript 的唯一部分,也不是唯一被标准化的部分。实际上,一个完整的 JavaScript 是由以下 3 个不同部分组成的,如图 1-5 所示。

1. ECMAScript 标准

ECMAScript 是一种开放的、在国际上广为接受的、标准的脚本语言规范,它不与任何的浏览器绑定,实际上,它也没有提到用于任何用户输入/输出的方法,那么 ECMAScript 标准主要对哪些内容实行了规范呢?简单地说,ECMAScript 主要描述了以下内容。

- (1) 语法。
- (2) 变量和数据类型。
- (3) 关键字、保留字。
- (4) 运算符。
- (5) 逻辑控制语句。
- (6) 对象。

ECMAScript 定义了脚本语言的所有属性、方法和对象,因此在使用 Web 客户端脚本语言编码时一定要遵循 ECMAScript 标准。

2. 浏览器对象模型(BOM)

从 Internet Explorer 3 和 Netscape Navigator 3 开始,浏览器都提供一种被称为 BOM (Browser Object Model)的特性,它可以对浏览器窗口进行访问和操作。利用 BOM 的相关技术,Web 开发者可以移动窗口、改变状态栏以及执行一些与页面内容不相关的操作。

3. 文档对象模型(DOM)

DOM 是 Document Object Model(文档对象模型)的简称,是 HTML 文档对象模型 (HTML DOM)定义的一套标准方法,用来访问和操纵 HTML 文档。DOM 由万维网联盟(World Wide Web Consortium, W3C)定义,最新的浏览器都支持第一级和第二级 DOM(第一级和第二级 DOM 是一种标准),通过 JavaScript,可以重构整个 HTML 文档。

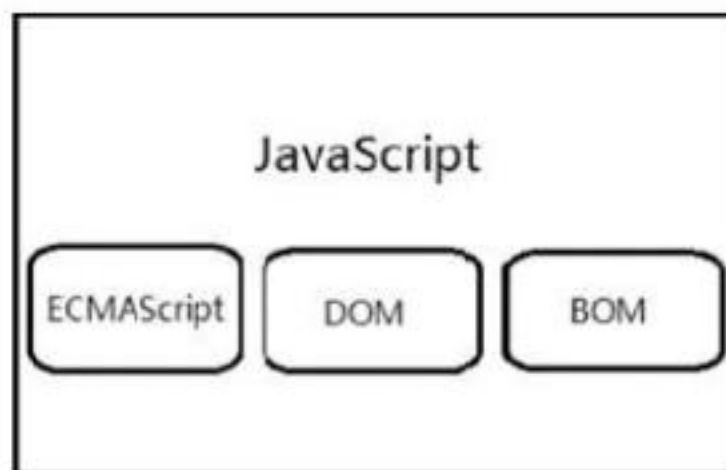


图 1-5 JavaScript 的组成部分

任务 1.2 在页面显示个人信息



任务描述

使用浏览器打开页面时,在页面使用 JavaScript 语言输出个人信息,包括所在班级、学号、姓名、性别和爱好。页面效果如图 1-6 所示。

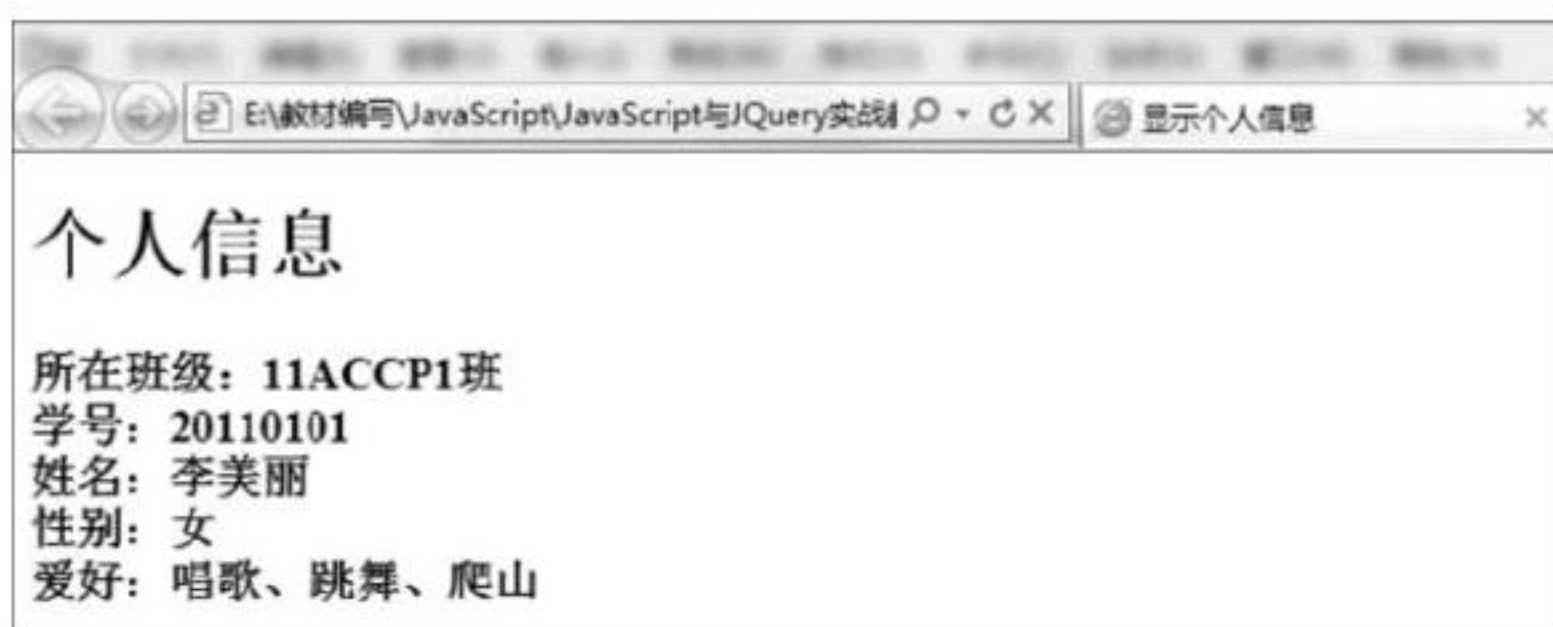


图 1-6 个人信息页面



任务分析

本任务要在页面使用 JavaScript 语言输出个人信息,完成该任务可以使用下列步骤。

- (1) 选择 JavaScript 脚本编辑器编辑脚本代码。
- (2) 嵌入脚本代码到 HTML 文档中。
- (3) 使用 document.write 语句在页面输出信息。
- (4) 使用浏览器浏览该 HTML 文档。

1.2.1 选择 JavaScript 脚本编辑器

编写 JavaScript 脚本代码可以选择普通的文本编辑器,如记事本、UltraEdit、EditPlus 等,只要所选择的编辑器能将所编辑的代码最终保存为 HTML 文档类型(扩展名为 .html、.htm 等)即可。也可以选择专业的网页编辑工具,如 Dreamweaver、FrontPage 等,这些专业网页开发工具内部集成了 JavaScript 脚本的开发环境,使用简单、方便。本书案例采用 Dreamweaver 专业网页开发工具实现。

1.2.2 脚本的基本结构

通常 JavaScript 代码通过<script>标签嵌入 HTML 文档中,可以将多个脚本嵌入一个文档中,只需将每个脚本都封装在<script>标签中。浏览器遇到<script>标签时,将逐行读取内容,直到遇到</script>结束标签为止。浏览器将检查 JavaScript 语句的语法,如果有错误,将提示错误;如果没有错误,浏览器将解释执行语句。

1. 脚本的基本结构

脚本的基本结构如下:

```
<script language="javascript 1.2" type="text/javascript">  
    //JavaScript 语句;  
</script>
```

其中<script>和</script>标签对将 JavaScript 脚本代码进行封装,同时告诉浏览器其间代码为 JavaScript。language 属性指定封装代码的脚本语言及版本,type 属性指定插入的脚本代码类型。



2. 脚本的执行原理

在脚本的执行过程中,浏览器客户端与服务器端采用请求/响应模式进行交互,如图 1-7 所示。

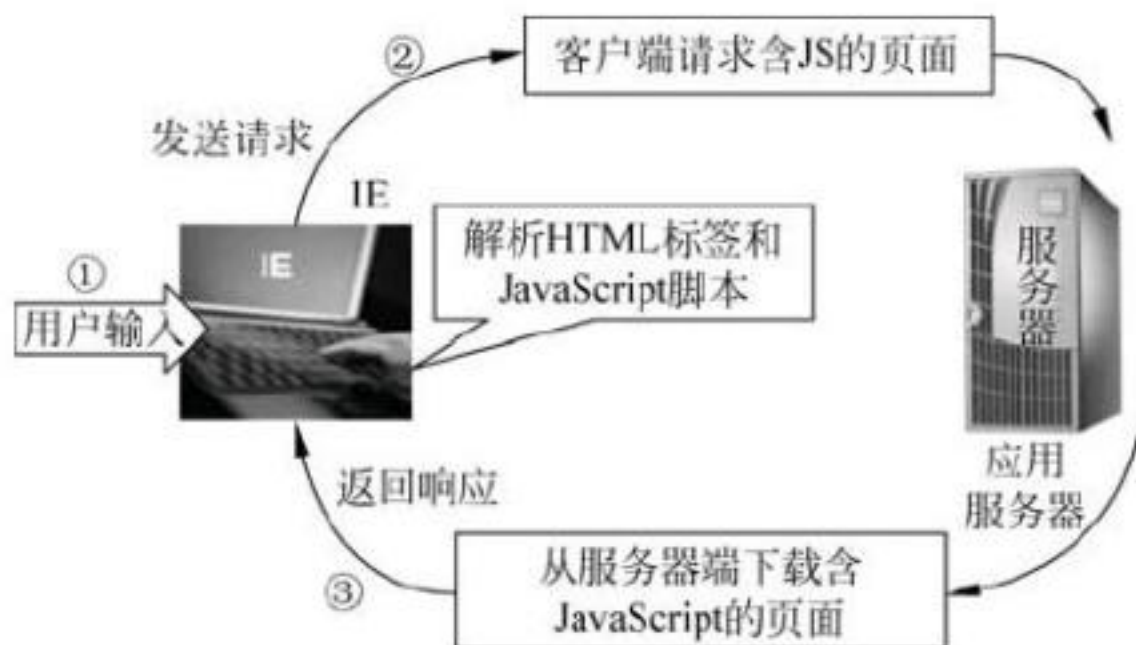


图 1-7 脚本执行原理

下面解析一下这个过程。

(1) 浏览器接收用户请求,即用户在浏览器的地址栏中输入要访问的页面(页面包含 JavaScript 脚本代码)。

(2) 浏览器把请求消息发送到 Web 服务器端,等待服务器端响应。

(3) Web 服务器端响应请求,Web 服务器端找到请求的页面,将整个页面包含 JavaScript 脚本代码作为响应内容,发送回客户端;客户端浏览器打开回应的网页文件内容,从上往下逐行读取并显示其中的 HTML 标签和 JavaScript 脚本代码。

使用客户端脚本有如下好处。

(1) 含脚本的页面只要下载一次即可,减少了不必要的网络通信。

(2) 脚本是从服务器端下载到客户端,然后在客户端进行的,不占用服务器端的资源,客户端分担了服务器端的任务,大大地减轻了服务器的压力,从而间接地提升了服务器的性能。

1.2.3 在网页中引用 JavaScript 的方式

JavaScript 脚本代码可以通过多种方式嵌入 HTML 文档中,主要有以下几种:①使用 `<script>...</script>` 标签对;②链接外部的 JavaScript 文件;③JavaScript 伪 URL 引入。

1. 使用 `<script>...</script>` 标签对

使用 `<script>...</script>` 标签对将 JavaScript 脚本代码嵌入 HTML 文档中,是最常用的方法。

【示例 1-1】

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>script 标签方式</title>
<script type="text/javascript">
  
```




```
    alert("知识是伟大的!");  
</script>  
</head>  
<body>  
</body>  
</html>
```

浏览器载入嵌有 JavaScript 脚本的 HTML 文档时,能自动识别 JavaScript 脚本代码起始标签<script>和结束标签</script>,并将其间的代码解释执行,然后将解释结果返回 HTML 并在浏览器窗口显示。

<script>...</script>的位置并不是固定的,可以包含在<head>...</head>和<body>...</body>中的任何位置。这种方式通常适用于 JavaScript 脚本代码比较少,并且网站中的每个页面使用的 JavaScript 脚本代码均不同的情况。在实际项目开发中,开发人员有时希望在多个页面运行 JavaScript 实现相同的页面效果,这时通常使用外部 JavaScript 文件。

2. 链接外部的 JavaScript 文件

外部 JavaScript 文件是将 JavaScript 写入一个外部文件中,以.js 为后缀保存这个文件,然后通过<script>标签的 src 属性将脚本代码嵌入 HTML 文档中。

【示例 1-2】

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

hello.js 就是外部 JavaScript 文件,src 属性表示指定外部 JavaScript 文件的路径,一般采用相对路径。



注意

外部文件中不能包含<script>标签,把.js 文件放到通常存放脚本的目录中,这样容易管理和维护。

3. JavaScript 伪 URL 引入

通过伪 URL 地址调用语句来引入 JavaScript 脚本代码的格式如下:

JavaScript: JavaScript 脚本代码

【示例 1-3】

```
<h4>通过伪 URL 方式引入 JavaScript 脚本代码:</h4>  
<form>  
<input type="button" value="伪 URL 方式" onclick="javascript:alert('这是另一种方式调用脚本代码')"/>  
</form>
```

1.2.4 常用的输入/输出语句

在 JavaScript 中常用的输入/输出语句有:警告对话框 alert()、提示对话框 prompt() 和输出语句 document.write()。



1. 警告对话框 alert()

alert()方法会创建一个警告对话框,用于将浏览器或文档的警告信息传递给客户。参数可以是变量、字符串或表达式,警告对话框无返回值。

alert()方法的基本语法格式如下:

```
alert("提示信息");
```

(1) 参数是字符串

```
alert("欢迎来到 JavaScript 世界!");
```

效果如图 1-8 所示。

(2) 参数是变量

```
var myName = "Jacky";  
alert("大家好,\n我是" + myName);
```

效果如图 1-9 所示。



说明:

JavaScript 中提供了一些特殊字符,允许在字符串中包括一些无法直接输入的字符,每个字符以反斜杠\开始,其中\n就是转义字符,表示换行,其他常用的转义字符还有\r回车符,\t制表符,\"双引号,\\反双斜杠。

(3) 参数是表达式

```
alert("7 + 8 = " + (7 + 8));
```

效果如图 1-10 所示。



图 1-8 参数是字符串



图 1-9 参数是变量



图 1-10 参数是表达式

2. 提示对话框 prompt()

prompt()方法会弹出一个提示对话框,用于收集客户关于特定问题的反馈信息,提示对话框具有返回值。

prompt()方法的基本语法格式如下:

```
prompt("提示信息","输入框的默认信息");
```

该方法的返回值可以被引用或存储到变量中。

**【示例 1-4】**

```
<script type="text/javascript">
    var sports = prompt("输入你最喜欢的运动:", "打羽毛球");
    alert("你最喜欢的运动是: " + sports);
</script>
```

运行效果如图 1-11 所示,单击“确定”按钮后,对话框如图 1-12 所示。

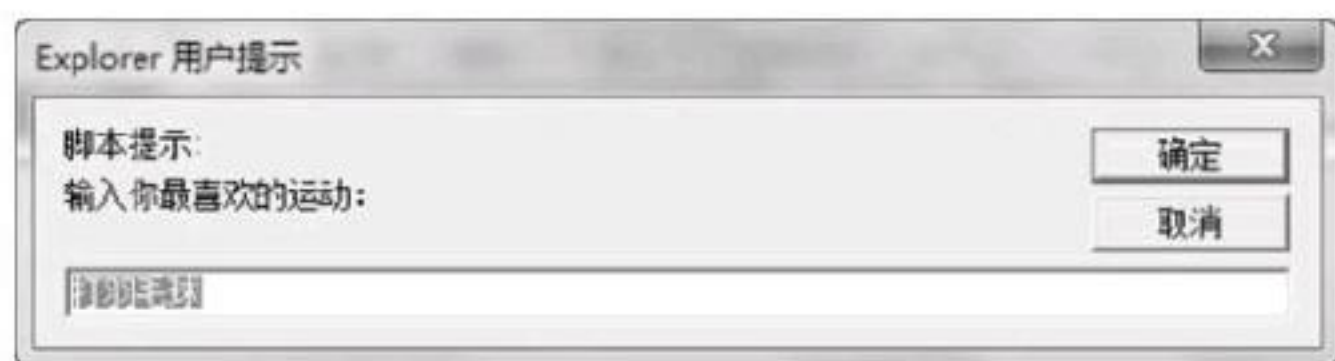


图 1-11 提示对话框

**说明:**

如果用户单击“取消”按钮, `prompt()` 方法将返回 `null`; 如果用户单击“确定”按钮, `prompt()` 方法将返回用户输入的字符串或默认字符串; 如果用户没有设置 `prompt` 方法的第二个参数, 默认值将得到“undefined”的值。

3. 输出语句 `document.write()`

`document.write()` 方法可以向文档写文本、HTML 表达式或 JavaScript 脚本代码。该方法需要一个字符串参数, 它是写到文档 HTML 的内容, 这些字符串参数可以是变量或值为字符串的表达式, 写入的内容常常包括 HTML 标记语言。

`document.write()` 方法的基本语法格式如下:

```
document.write("输出内容");
```

(1) 参数是字符串

```
document.write("JavaScript 的学习非常有趣");
```

(2) 参数是变量

```
var myName = "Jacky";
document.write("大家好,我是" + myName);
```

(3) 参数是 HTML 标签

```
document.write("<h1>网页制作常用工具</h1><ul><li>图像处理 Photoshop 工具</li><li>网  
页编辑 Dreamweaver 工具</li></ul>");
```

其运行效果如图 1-13 所示。



图 1-12 单击“确定”按钮的效果

- 图像处理Photoshop工具
- 网页编辑Dreamweaver工具

1.2.5 任务实现

- (1) 启动网页编辑工具,如 Dreamweaver CS6,新建网页。
- (2) 在<head>标签对中嵌入<script></script>标签。
- (3) 使用 document.write()输出语句输出信息。

```
<script type="text/javascript">
document.write("<h1>个人信息</h1>");
document.write("<h3>所在班级：11ACCP1 班<br/>学号：20110101 <br/>姓名：李美丽<br/>性别：女<br/>爱好：唱歌、跳舞、爬山</h3>");
</script>
```

任务描述

利用不同的浏览器测试综合案例的效果,由于浏览器对代码兼容性不一样,可能在不同的浏览器上看到的页面效果会有所差异。IE 浏览器测试效果如图 1-14 所示,Firefox 浏览器测试效果如图 1-15 所示。





图 1-15 Firefox 浏览器测试效果

任务分析

为了使页面在不同的浏览器中呈现尽可能相同的效果,在编写脚本代码时要遵循 Web 标准,并且要在各主流浏览器上做兼容性测试。

1.3.1 浏览器之间的冲突

由于商业利益的竞争,各浏览器厂商对脚本代码的支持程度不一样,这样带来的结果就是相同的代码在不同浏览器里呈现出来的效果有差异,我们将引起这些差异的问题统称为浏览器兼容性问题。例如,IE 浏览器中,event 对象有 x、y 属性,但是没有 pageX、pageY 属性;Firefox 浏览器中,event 对象有 pageX、pageY 属性,但是没有 x、y 属性。

1.3.2 标准的制定

就在浏览器厂商之间为了商业利益展开激烈竞争时,万维网联盟 W3C 于 1998 年 10 月发布了 DOM Level 1,随后又相继发布了 DOM Level 2 和 DOM Level 3。尽管各个浏览器之间的 JavaScript 至今仍然存在差异,但相对以前的状况已经得到了很大改善,标准化的语句对于 JavaScript 的发展起了关键性的作用,各个浏览器厂商都是围绕着标准进行,这也使得开发者可以将大部分精力用于页面本身功能的开发,而不再是协调浏览器之间的兼容性问题。

1.3.3 任务实现

根据任务分析,实现本任务的具体操作步骤如下:

- (1) 在 IE 浏览器中测试综合案例。
- (2) 在 Firefox 浏览器中测试综合案例。
- (3) 在 Chrome 浏览器中测试综合案例。
- (4) 记录在三大主流浏览器中页面呈现的差异。



小 结

本章介绍了 JavaScript 的起源,说明了什么是 JavaScript、JavaScript 有何作用及 JavaScript 的特点,同时还介绍了 JavaScript 的组成,如何在页面引入 JavaScript 脚本,各浏览器对 JavaScript 的支持不一致导致相同代码在不同浏览器呈现的效果可能有所差异。下面对本章内容做一个小结。

(1) JavaScript 的主要作用体现在客户端表单验证、页面动态效果和动态改变页面内容等方面。

(2) JavaScript 的基本特点: JavaScript 是基于对象和事件驱动,是解释性语言,具有实时性、动态性、跨平台性和安全性,开发使用简单。

(3) JavaScript 由 3 部分组成,分别是 ECMAScript、DOM 和 BOM,其中 ECMAScript 是 JavaScript 的标准规范,BOM 与浏览器窗口进行交互,DOM 可以控制页面的每一个元素,实现页面中的各种动态效果。

(4) 在页面中引入 JavaScript 主要有 3 种方式: ①使用 `<script>...</script>` 标签对; ②链接外部的 JavaScript 文件; ③JavaScript 伪 URL 引入。

(5) 常用的输入/输出语句有: ①警告对话框 `alert()`; ②提示对话框 `prompt()`; ③输出语句 `document.write()`。

实 训

实训目的

- (1) 认识 JavaScript,了解 JavaScript 的组成。
- (2) 掌握在 Web 页面中使用 JavaScript 的方法。
- (3) 掌握常用的输入/输出语句的使用方法。
- (4) 熟悉 JavaScript 的编写工具,并能够运用其编写简单的程序。

实训 1 分别使用 HTML 方式和 JavaScript 脚本方式在页面显示信息

训练要点

- (1) 掌握 HTML 基本结构。
- (2) 掌握在网页中使用 `<script>` 标签嵌入脚本的方式。
- (3) 掌握使用 `document.write()` 方法输出信息。

需求说明

在页面显示如图 1-16 所示的信息。

实现思路及步骤

- (1) 使用 HTML 方式完成

- ① 启动 Dreamweaver CS6,新建网页。
- ② 将网页标题改为“实训 1”。

修改 `<title>` 标签中的内容为: `<title>实训 1</title>`。



图 1-16 显示信息效果

③ 在<body>标签中添加页面内容。

在<body>标签中添加下列代码。

```
<h2>网页主要由三部分组成</h2>
```

```
<h4>
```

```
  <ul>
```

```
    <li>结构</li>
```

```
    <li>表现</li>
```

```
    <li>行为</li>
```

```
  </ul>
```

```
</h4>
```

(2) 使用 JavaScript 脚本方式完成

① 启动 Dreamweaver CS6,新建网页。

② 将网页标题改为“实训 1”。

修改<title>标签中的内容为:<title>实训 1</title>。

③ 在页面添加<script type="text/javascript"></script>标签,并在标签内添加代码。

在<script>标签中添加下列代码。

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

```
document.write("<h2>网页主要由三部分组成</h2>");
```

```
document.write("<h4><ul><li>结构</li><li>表现</li><li>行为</li></ul></h4>");
```

```
</script>
```

实训 2 在 Web 页面引入 JavaScript 文件

训练要点

(1) 学会创建 JavaScript 文件。

(2) 掌握在页面中引入 JavaScript 文件的方法。

需求说明

将实训 1 的脚本代码存放在单独的文件中,并在页面中引入该文件。

实现思路及步骤

(1) 在 Dreamweaver CS6 中,单击“文件”→“新建”→“页面类型”→JavaScript→“创建”命令。

(2) 在新建的文件中输入代码。



输入如下代码。

```
document.write("<h2>网页主要由三部分组成</h2>");  
document.write("<h4><ul><li>结构</li><li>表现</li><li>行为</li></ul></h4>");
```

(3) 保存 JavaScript 文件,保存名为“training1.js”,注意保存路径。

(4) 在页面中使用 `<script type="text/javascript" src="training1.js"></script>` 引入 JavaScript 文件。



注意

这里引用的 JavaScript 文件与当前页面文件存放在同一目录下。

实训 3 使用伪 URL 方式引入 JavaScript 脚本代码

训练要点

- (1) 通过伪 URL 方式引入 JavaScript 脚本代码。
- (2) 掌握 alert() 方法的应用。

需求说明

- (1) 在 Web 页面包含有一个提交按钮。
- (2) 单击“提交”按钮时弹出如图 1-17 所示的页面。



图 1-17 页面内容及效果

实现思路及步骤

- (1) 建立 HTML 页面。
- (2) 给按钮添加 onclick 事件。

在 HTML 页面给按钮添加脚本代码,HTML 代码及脚本代码如下:

```
<input type="button" value="提交" onclick="javascript:alert('若没有输入用户名,请输入用户名');"/>
```

实训 4 使用转义字符在页面输出信息

训练要点

- (1) 学会使用转义字符。
- (2) 使用 alert 语句输出信息。



需求说明

完成如图 1-18 所示的页面效果制作。

实现思路及步骤

- (1) 建立新的页面。
- (2) 使用 alert() 方法输出信息, 字符信息中包含换行转义字符。

参考代码如下:

```
<script type="text/javascript">
    alert("祝愿大家:\n学习进步!\n更上一层楼!");
</script>
```



图 1-18 转义字符输出

实训 5 将用户输入的信息在页面输出

训练要点

掌握 prompt 语句的用法。

需求说明

完成如图 1-19 及图 1-20 所示效果的制作。

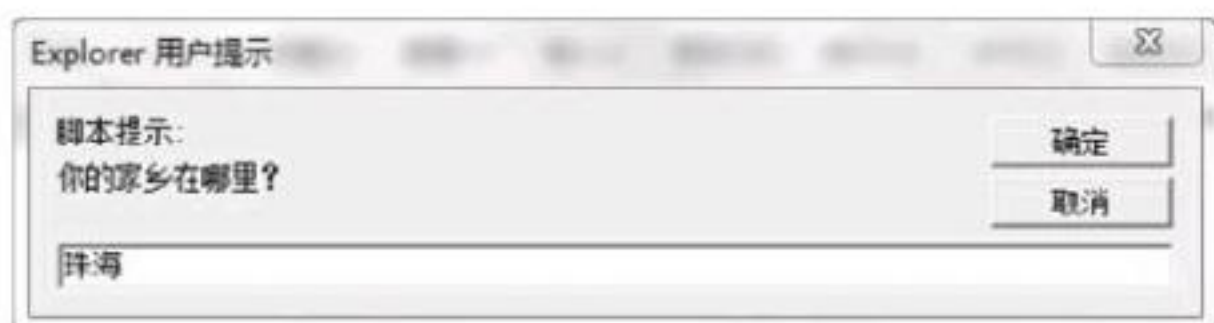


图 1-19 提示信息框



图 1-20 弹出消息框

实现思路及步骤

- (1) 建立新的页面。
- (2) 使用 prompt() 方法接收用户输入。
- (3) 使用 alert() 方法将用户输入的信息显示。

参考代码如下:

```
<script type="text/javascript">
    var city = prompt("你的家乡在哪里?", "珠海");
    alert("你的家乡在:" + city);
</script>
```

课后练习

一、选择题

1. 下列关于向 HTML 页面嵌入 JavaScript 脚本的说法中描述正确的是()。
 - A. JavaScript 脚本只能放置在 HTML 页面<head>与</head>中
 - B. JavaScript 脚本可以放置在 HTML 页面中的任何地方

- ## 二、操作题

1. 使用 JavaScript 语句在页面输出如图 1-21 所示的信息。
2. 使用 JavaScript 语句在页面输出如图 1-22 所示的信息,其中北京为默认值。



JavaScript世界真奇妙

快和我一起来学习吧！

图 1-21 输出信息



图 1-22 输入城市信息

书 码

BOOKASK.COM



本书试读到此结束啦！



JavaScript与jQuery实战教程

作者：卢淑萍, 樊红珍, 主编

出版社：清华大学出版社

通过以下方式阅读更多 [Powered by 书问](#)



● 扫码分享到朋友圈可阅读更多



立即扫码



● 还不过瘾？购买书库畅读卡全本畅读此书！



[查看全部书库](#)



● 购买纸书也可畅读全本哦！

[蔚蓝网](#)

[中国图书网](#)