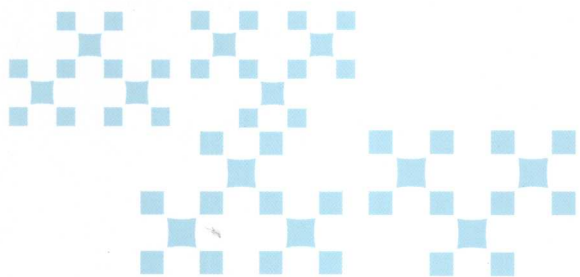


JavaScript+jQuery 程序开发实用教程

李雨亭 吕 婕 王泽璘 编著



注重基础知识及案例的讲解。

- ◆ 教材示例源文件丰富，实例具有普遍性和实用性。
- ◆ 每章都有课后练习题，便于巩固学习成果。
- ◆ 教材有配套的电子教案与习题答案，方便老师教学时使用。

免费赠送
PPT电子课件
及习题答案



清华大学出版社

大学软件学院软件开发系列教材

JavaScript+jQuery 程序开发实用教程

李雨亭 吕 婕 王泽璘 编 著

清华大学出版社
北 京

内 容 简 介

本书循序渐进地介绍了 JavaScript 开发技术。深入分析了 JavaScript 的核心知识,并在此基础上详细讲解了 jQuery 框架的使用。此外,在每个重要知识点讲解的后面,通过丰富典型的案例,使读者进一步巩固所学的知识,提高实际开发能力。

本书内容全面,实例丰富,易于理解,每章的内容都简洁紧凑,从最佳实践的角度入手,为读者更好地使用 JavaScript 及 jQuery 框架开发动态网页提供了很好的指导。

本书适合高等院校计算机科学、软件工程、数字媒体技术、通信及相关专业本、专科作为动态网页程序设计相关课程教材使用,也是打算学习和正从事 JavaScript+jQuery 动态网页设计的开发人员的教材或参考书。

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+jQuery 程序开发实用教程/李雨亭,吕婕,王泽璘编著. —北京:清华大学出版社,2016
大学软件学院软件开发系列教材
ISBN 978-7-302-41907-5

I. ①J… II. ①李… ②吕… ③王… III. ①Java 语言—程序设计—高等学校—教材 IV. ①TP312
中国版本图书馆 CIP 数据核字(2015)第 259930 号

责任编辑:杨作梅

装帧设计:杨玉兰

责任校对:李玉萍

责任印制:刘海龙

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>, 010-62791865

印 装 者:北京鑫丰华彩印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.5 字 数:523 千字

版 次:2016 年 1 月第 1 版 印 次:2016 年 1 月第 1 次印刷

印 数:1~2000

定 价:45.00 元

产品编号:045188-01

前 言

本书立足于 JavaScript 原生语言基础，对其语法、函数和事件等作了详细介绍，并提供了大量实战案例来对重点知识点的应用进行了详细讲解。最后结合最为常用的 jQuery 框架介绍如何使用 JavaScript+jQuery 进行动态网页开发。

本书共分为 16 章，各章主要内容说明如下：

第 1 章为 JavaScript 概述，主要介绍了 JavaScript 脚本语言的主要特征和优、缺点，还介绍了 JavaScript 的具体应用及编辑 JavaScript 脚本语言的几种常用工具。

第 2 章讲述 JavaScript 的数据类型和运算符，主要介绍了 JavaScript 的基本语法，包括数据类型和运算符。

第 3 章讲述 JavaScript 的流程控制，主要介绍了 JavaScript 程序中各种流程结构。

第 4 章讲述 JavaScript 中的函数，主要介绍了 JavaScript 中函数的调用和使用。

第 5 章讲述 JavaScript 中的对象，主要介绍了对象的基本概念和 JavaScript 中常用的内置对象和浏览器对象。

第 6 章讲述 JavaScript 中的事件与事件处理，主要介绍了 JavaScript 中常用的事件及其处理程序的编写。

第 7 章讲述 JavaScript 中的 DOM 编程，在介绍文档对象模型的基础上，详细讲解了如何使用 JavaScript 语言进行 DOM 编程。

第 8 章讲述 CSS 样式表，主要介绍了 CSS 样式表的定义、选择器以及各种属性。

第 9 章讲述 JavaScript 的网页特效，主要介绍了 JavaScript 中各种常见的网页特效的实现方法。

第 10 章为初识 jQuery，主要对 jQuery 的基本情况进行了介绍，并讲解了 jQuery 的安装和配置。

第 11 章讲述 jQuery 选择器，主要介绍了 jQuery 中常用选择器的基本使用方法。

第 12 章讲述 jQuery 中 DOM 的操作，主要介绍了 jQuery 中的各种 DOM 操作。

第 13 章讲述 jQuery 的事件处理，在介绍 jQuery 中的事件处理机制的基础上，详细讲解了 jQuery 中常见的事件处理方法。

第 14 章讲述 jQuery 的动画效果，详细介绍了 jQuery 中的各种动画效果的实现方法。

第 15 章讲述 jQuery 与 Ajax，详细介绍了 jQuery 中的 Ajax 的实现方法。

第 16 章讲述 jQuery 的常用插件，详细介绍了 jQuery 中的各种常用插件的使用方法。

本书由李雨亭、吕婕、王泽磷编著，参加编写的还有孙更新、宾晟、李宗颜、孙海伦、宫生文、解本巨、李海涛、史爱松、李晓娜、王萍萍等。

由于作者水平有限，本书的内容难免会有纰漏和不足之处，恳请各位专家同仁和读者批评指正。

编 者

目 录

第 1 章 JavaScript 概述 1

1.1 JavaScript 简介 2

1.1.1 JavaScript 语言简史 2

1.1.2 JavaScript 的特点 2

1.1.3 JavaScript 在页面中的应用 3

1.2 HTML 页面中嵌入 JavaScript 的方法 4

1.2.1 页面中定义 JavaScript 代码 5

1.2.2 链接外部 JavaScript 文件 6

1.2.3 事件调用 JavaScript 程序 7

1.3 JavaScript 代码的编辑工具 8

1.3.1 纯文本编辑器 9

1.3.2 Dreamweaver 9

课后小结 11

习题 11

第 2 章 JavaScript 的数据类型和运算符 13

2.1 JavaScript 的语法规则 14

2.1.1 区分大小写 14

2.1.2 代码的格式 14

2.1.3 代码的注释 15

2.1.4 常量 15

2.1.5 空白符和换行符 16

2.1.6 标识符 17

2.1.7 保留字 17

2.2 数据类型 18

2.2.1 数值型 18

2.2.2 字符串类型 19

2.2.3 布尔型 20

2.2.4 空值型 20

2.2.5 未定义值 21

2.2.6 类型转换 21

2.3 变量 24

2.3.1 变量的命名 24

2.3.2 变量的声明 24

2.3.3 变量的赋值 26

2.4 运算符 27

2.4.1 算术运算符 28

2.4.2 赋值运算符 31

2.4.3 关系运算符 32

2.4.4 逻辑运算符 34

2.4.5 字符串运算符 36

2.4.6 其他运算符 37

2.4.7 运算符的优先级 38

课后小结 39

习题 39

第 3 章 JavaScript 的流程控制 41

3.1 JavaScript 中的选择结构 42

3.1.1 if 语句 42

3.1.2 if...else 语句 42

3.1.3 嵌套 if...else 语句 44

3.1.4 switch 语句 46

3.2 JavaScript 中的循环结构 48

3.2.1 while 语句 48

3.2.2 do...while 语句 49

3.2.3 for 语句 50

3.2.4 for...in 语句 51

3.2.5 with 语句 52

3.2.6 break 语句 54

3.2.7 continue 语句 55

课后小结	56	5.3.6 navigator 对象	109
习题	56	课后小结	111
第 4 章 JavaScript 中的函数	59	习题	111
4.1 函数的定义	60	第 6 章 JavaScript 中的事件与事件处理	113
4.2 函数的调用	61	6.1 事件及事件处理程序	114
4.3 函数的参数和返回值	62	6.1.1 事件和事件处理程序	114
4.3.1 函数的参数	62	6.1.2 指定事件处理程序	114
4.3.2 函数的返回值	66	6.2 JavaScript 的常用事件	116
4.4 函数的嵌套和递归	67	6.2.1 键盘事件	116
4.5 变量的作用域	70	6.2.2 鼠标事件	118
4.6 JavaScript 中的系统函数	71	6.2.3 Load 事件和 Unload 事件	121
4.6.1 encodeURIComponent 函数	71	6.2.4 Focus 事件和 Blur 事件	123
4.6.2 decodeURI 函数	72	6.2.5 Submit 事件和 Reset 事件	125
4.6.3 parseInt 函数	73	6.2.6 Change 事件和 Select 事件	127
4.6.4 isNaN 函数	74	6.2.7 Error 事件	129
4.6.5 eval 函数	75	课后小结	129
课后小结	76	习题	130
习题	76	第 7 章 JavaScript 中的 DOM 编程	131
第 5 章 JavaScript 中的对象	77	7.1 XML 基础	132
5.1 对象的基本概念	78	7.1.1 XML 的文档结构	132
5.1.1 对象的属性和方法	78	7.1.2 XML 解析器	133
5.1.2 对象的创建和删除	79	7.2 DOM 编程基础	134
5.2 内置对象	81	7.2.1 DOM 文档对象模型	134
5.2.1 String 对象	81	7.2.2 DOM 中的节点	135
5.2.2 Number 对象	85	7.3 使用 DOM 编程	137
5.2.3 Math 对象	88	7.3.1 访问节点	137
5.2.4 Date 对象	90	7.3.2 创建新节点	138
5.2.5 Array 对象	93	7.3.3 添加节点	139
5.3 浏览器对象	98	7.3.4 插入节点	140
5.3.1 window 对象	99	7.3.5 删除节点	142
5.3.2 document 对象	102	课后小结	145
5.3.3 frame 对象	104	习题	145
5.3.4 history 对象	106		
5.3.5 location 对象	108		





第 8 章 CSS 样式表	147	9.4 窗体特效	197
8.1 CSS 简介	148	9.4.1 无边框窗口自动关闭特效	197
8.1.1 CSS 的发展	148	9.4.2 方向键控制窗口的特效	199
8.1.2 CSS 的特点	148	9.4.3 改变窗体颜色	200
8.2 CSS 的定义方式	149	9.5 鼠标特效	201
8.3 CSS 的选择器	150	9.5.1 屏蔽鼠标右键	201
8.3.1 元素选择器	150	9.5.2 获取鼠标位置坐标	202
8.3.2 类选择器	152	9.5.3 根据方向改变鼠标外观	204
8.3.3 ID 选择器	153	9.6 菜单特效	205
8.3.4 属性选择器	155	9.6.1 左键弹出菜单	205
8.3.5 后代选择器	156	9.6.2 下拉菜单	207
8.3.6 子元素选择器	158	9.6.3 滚动菜单	208
8.3.7 相邻兄弟选择器	159	9.7 警告和提示特效	210
8.4 CSS 中的属性	161	9.7.1 进站提示信息	210
8.4.1 字体属性设置	164	9.7.2 单击超链接显示提示框	211
8.4.2 文本属性设置	168	9.7.3 显示停留时间	211
8.4.3 背景属性设置	171	9.8 密码特效	213
8.4.4 边框属性设置	175	9.8.1 弹出式密码保护	213
8.4.5 边距和填充距属性设置	178	9.8.2 检查密码的格式合法性	215
课后小结	179	课后小结	218
习题	179	第 10 章 初识 jQuery	219
第 9 章 JavaScript 网页特效	181	10.1 jQuery 概述	220
9.1 文字特效	182	10.1.1 jQuery 简介	220
9.1.1 跑马灯效果	182	10.1.2 jQuery 的特点	220
9.1.2 打字效果	183	10.2 jQuery 代码编写	221
9.1.3 文字大小变化效果	185	10.2.1 配置 jQuery 开发环境	221
9.1.4 升降文字效果	187	10.2.2 jQuery 代码编写	221
9.2 图片特效	188	10.3 jQuery 对象与 DOM 对象	222
9.2.1 改变页面中图片的位置	188	10.3.1 jQuery 对象简介	222
9.2.2 鼠标拖动改变图片大小	190	10.3.2 jQuery 对象与 DOM 对象	
9.2.3 不断闪烁的图片	191	的相互转换	223
9.3 时间和日期特效	192	课后小结	223
9.3.1 标题栏显示分时间问候语	192	习题	224
9.3.2 显示当前系统时间	193	第 11 章 jQuery 选择器	225
9.3.3 星期查询功能	195	11.1 jQuery 选择器简介	226
		11.2 jQuery 选择器的分类	226



11.2.1 基本选择器	226	13.2 jQuery 中的页面载入事件	270
11.2.2 层次选择器	228	13.3 jQuery 中的事件绑定	272
11.2.3 过滤选择器	229	13.3.1 bind()方法绑定事件	272
11.2.4 表单选择器	235	13.3.2 one()方法绑定事件	273
11.3 jQuery 中元素属性的操作	237	13.3.3 live()方法绑定事件	274
11.3.1 设置元素属性	237	13.4 jQuery 中的事件移除	276
11.3.2 删除元素属性	238	13.5 jQuery 中的事件冒泡	278
11.4 jQuery 中样式类的操作	239	13.6 jQuery 中的模拟事件触发操作	279
11.4.1 添加样式类	239	13.7 jQuery 中的合成事件	282
11.4.2 移除样式类	241	13.7.1 hover()方法	282
11.4.3 交替样式类	242	13.7.2 toggle()方法	283
11.5 jQuery 中样式属性的操作	243	课后小结	284
11.5.1 读取样式属性	243	习题	284
11.5.2 设置样式属性	244	第 14 章 jQuery 的动画效果	285
11.5.3 设置元素偏移	245	14.1 显示与隐藏效果	286
11.6 jQuery 中元素内容的操作	247	14.1.1 隐藏元素的 hide()方法	286
11.6.1 操作 HTML 代码	247	14.1.2 显示元素的 show()方法	287
11.6.2 操作文本	248	14.1.3 交替显示隐藏元素	288
11.6.3 操作表单元素的值	249	14.2 滑动效果	288
11.7 筛选与查找元素集中的元素	250	14.2.1 向上收缩效果	289
课后小结	255	14.2.2 向下展开效果	290
习题	255	14.2.3 交替伸缩效果	290
第 12 章 jQuery 中 DOM 的操作	257	14.3 淡入淡出效果	291
12.1 DOM 树结构	258	14.3.1 淡入效果	291
12.2 创建元素	259	14.3.2 淡出效果	293
12.3 插入元素	259	14.3.3 交替淡入淡出效果	294
12.4 复制元素	261	14.3.4 不透明效果	295
12.5 替换元素	262	14.4 自定义动画效果	295
12.6 包裹元素	263	14.4.1 自定义动画	296
12.7 删除元素	264	14.4.2 动画队列	297
课后小结	266	14.4.3 动画停止和延时	299
习题	266	课后小结	301
第 13 章 jQuery 的事件处理	267	习题	301
13.1 jQuery 中的事件处理机制	268	第 15 章 jQuery 与 Ajax	303
		15.1 Ajax 简介	304





15.2 jQuery 中的 Ajax 方法	306
15.2.1 load()方法	306
15.2.2 \$.get()方法和\$.post()方法	307
15.2.3 \$.getScript()方法和 \$.getJSON()方法	309
15.2.4 \$.ajax()方法	310
15.3 jQuery 中的 Ajax 事件	312
课后小结	315

习题	315
----------	-----

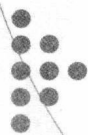
第 16 章 jQuery 常用插件

16.1 jQuery 中的表单插件	318
16.2 jQuery 中的验证插件	320
16.3 jQuery 中的右键菜单插件	325
16.4 jQuery 中的图片弹窗插件	329
课后小结	332
习题	333



第 1 章

JavaScript 概述



学习目标:

随着 Internet 的飞速发展,越来越多的用户每天都要访问各种具有实时的、动态的、可交互的网页效果的 Internet 站点。而早期静态的、缺乏交互的 HTML 页面满足不了这种要求,因此页面上这些动态效果就需要使用 JavaScript 语言来编写实现。本章将介绍 JavaScript 脚本语言的主要特征和优、缺点,还介绍 JavaScript 的具体应用及编辑 JavaScript 脚本语言的几种常用工具。

内容摘要:

- 了解 JavaScript 语言的特点
- 掌握 JavaScript 的具体应用
- 熟练掌握 JavaScript 语言的开发和调试工具

1.1 JavaScript 简介

JavaScript 是由 Netscape 公司开发的一种基于对象和事件驱动并具有安全性能的脚本语言(Scripting Language), 主要用在 Internet(国际互联网或因特网)的客户端上, 是“浏览器”上的程序语言。当 Web 服务器输出内容(包括 JavaScript 的程序代码)到浏览器时, JavaScript 可以操纵浏览器上的一切内容, 在浏览器上提供用户交互、页面美化、增强 Web 页面的“智能性”, 使网页包含更多活跃的元素和更加精彩的内容。

HTML 语言是网页设计中普遍采用的一种超文本标记语言, 但 HTML 自身不能为网页提供动态支持, 也不能接收用户输入, 更不能对用户请求做出反应。JavaScript 可以嵌入到 HTML 页面中使网页具有动态效果, 并具有交互性, 它的出现弥补了 HTML 语言的缺陷。

1.1.1 JavaScript 语言简史

JavaScript 是一种可以用来给网页增加交互性的编程语言。JavaScript 语言的最初名称为 LiveScript, 由 Netscape 公司开发。在 Sun 公司推出 Java 语言之后, Netscape 公司与 Sun 公司合作, 于 1995 年 12 月推出了一种新的语言, 称为 JavaScript 1.0。JavaScript 和 Java 虽然很类似, 但并不一样, Java 是一种比 JavaScript 更复杂的面向对象编程语言, 而 JavaScript 则是相当容易了解的脚本语言。JavaScript 开发者可以不那么注重程序技巧, 所以许多 Java 的特性在 JavaScript 中并不支持。

与当时已经存在的 Perl 等基于服务器的计算机语言相比, 由于 JavaScript 能够在当时的 Netscape Navigator 2.0 浏览器中进行操作, 也就意味着它可以在客户端运行。因此 JavaScript 具有很大的优势, 受到广大程序员的欢迎并飞速发展起来。

不久 Microsoft 公司的浏览器 Internet Explorer 3.0 也开始支持 JavaScript 了。此后, 许多公司相继宣布承认 JavaScript 为 Internet 上的开放式脚本编写标准, 并且把它添加到了自己的产品中。目前, 流行的浏览器都支持 JavaScript。

 **说明:** Microsoft 公司也有自己的 JavaScript, 称为 JScript。JavaScript 和 JScript 基本相同, 只是在一些细节上有出入。

1.1.2 JavaScript 的特点

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言。其特点主要表现在以下几个方面。

1. 简单性

JavaScript 是一种脚本语言, 像其他脚本语言一样, JavaScript 同样也是一种解释性语



言。它的基本结构形式与 C、C++ 十分类似。但它不像这些语言一样需要先编译，而是在程序运行过程中被逐行地解释。它与 HTML 标记结合在一起，从而方便用户的使用操作。

2. 动态性

JavaScript 是动态的，它可以直接对用户或者客户输入做出响应，无须经过 Web 服务程序。它对用户请求的响应采用事件驱动的方式进行。事件驱动就是指在主页中执行了某种操作所产生的动作，就称为“事件”，如按下鼠标、移动窗口、选择菜单等都可以视为事件，当事件发生后，可能会引起相应的事件响应。

3. 基于对象的语言

JavaScript 是一种基于对象的语言，这意味着它能运用已经创建的对象，但不能派生新的对象，也就是没有像 Java 等面向对象程序设计语言所具有的继承、多态等特点，这使得 JavaScript 更容易学习和掌握。

4. 安全性

JavaScript 是一种安全的语言，它不允许访问本地的硬盘，也不能将数据存入到服务器上，更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。从而有效地防止数据的丢失。

5. 跨平台性

JavaScript 依赖于浏览器本身，与操作环境无关，只要是能运行浏览器的计算机并支持 JavaScript 的浏览器就可以正确执行。

正是以上的这些 JavaScript 的特点，使得它在 Web 编程领域中被广泛地普及和运用，具有广阔的发展前景。

1.1.3 JavaScript 在页面中的应用

在使用 JavaScript 语言进行网页制作时，JavaScript 不但可以用于编写客户端的脚本程序，用来实现在 Web 浏览器端解释并执行脚本程序；而且还可以编写在服务器端执行的脚本程序，以实现服务器端处理用户提交的信息，并相应地更新浏览器显示的 Web 服务器程序。



说明： JavaScript 包含两种区分程序的方式：在用户计算机上运行的程序称为客户端(client-side)程序；在服务器上运行的程序称为服务器端(server-side)程序。

1. JavaScript 在客户端的应用

客户端的 JavaScript 脚本程序是被嵌入到 HTML 页面中或从外部连接到 HTML 页面的。当用户使用浏览器请求 HTML 页面时，JavaScript 脚本程序与 HTML 页面一起被下载到客户端，由客户端的浏览器读取 HTML 页面，并解析其中是否含有 JavaScript 脚本。如果有，就解释执行并以页面方式显示出来。整个过程如图 1-1 所示。



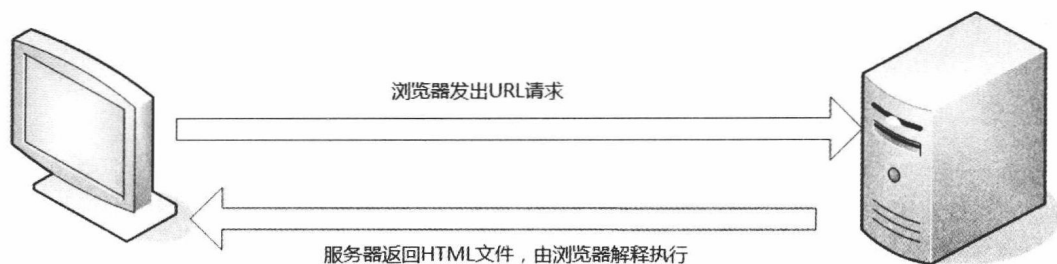


图 1-1 客户端 JavaScript

客户端 JavaScript 可以实现很多功能。在 HTML 页面中使用 JavaScript，可以利用表单元素和超链接使网页直接对用户做出响应；也能够利用警告、提示和确认信息向用户提示所发生的错误；还可以利用 JavaScript 程序改变浏览器窗口的外观等。

例如，使用 JavaScript 客户端程序输出一个简单的“Hello, World!”的消息，实现代码如下：

```
<script type="text/javascript">
var msg ='Hello,World!'
alert(msg);
</script>
```

2. JavaScript 在服务器端的应用

在服务器端，JavaScript 也是嵌入 HTML 页面的，服务器端使用<server></server>标签对。服务器端代码不会被浏览器直接解释，需要通过服务器解释执行。服务器端 JavaScript 有许多扩展的对象、方法、属性和事件。服务器端 JavaScript 经过编译以后生成二进制代码组成.web 文件。

 **说明：** 由于服务器端 JavaScript 是厂商特有的服务端技术，而且使用它的开发人员远远少于客户端 JavaScript。所以本书没有对它进行详细介绍。

1.2 HTML 页面中嵌入 JavaScript 的方法

在开始创建 JavaScript 程序之前，首先需要掌握创建 JavaScript 程序的方法以及如何在 HTML 页面中调用 JavaScript 程序。

为了能够让浏览器识别 HTML 页面中的 JavaScript 脚本代码，脚本代码必须包含在 Script 容器标签内。换言之，要用打开标签<script>开始脚本，用关闭标签</script>来结束脚本。具体格式如下。

```
<script>
  JavaScript 程序
</script>
```

<script></script> 标签对的位置并不是固定的，可以出现在 HTML 页面的<head>



</head> 或者 <body></body> 标签对之间。也可以在一个页面的多个位置，通过 <script></script> 标签对来嵌入多段 JavaScript 代码。

在 HTML 中嵌入 JavaScript 程序可以使用两种方式。<script> 标签有两个可选属性，这两个属性决定 JavaScript 脚本代码是以何种方式嵌入到 HTML 页面的。这两个属性的具体说明如表 1-1 所示。

表 1-1 WTK 目录结构功能描述

属 性	描 述
src	包含 JavaScript 源代码的文件的 URL。文件应以 .js 为扩展名
language	表示在 HTML 中使用哪种脚本语言

如果使用 src 属性，就能够把存储到某个单独文件中的 JavaScript 代码从外部引入进来，并将这个文件加载到 HTML 页面中。如果将 language 属性设置为“JavaScript”，表示 HTML 页面中的脚本语言是 JavaScript，这样就可以直接在 HTML 页面中编写 JavaScript 脚本程序。



说明：这两种属性可以单独使用，也可以并用。

上述两种方法嵌入到 HTML 页面中的 JavaScript 脚本都是在页面载入时开始运行的。这种在页面载入时就运行的脚本，称为实时脚本。如果 JavaScript 脚本是在页面载入后响应用户操作动作时才运行的，这种方式称为延时脚本。延时脚本是通过将 JavaScript 代码定义在函数中实现的。

1.2.1 页面中定义 JavaScript 代码

在 HTML 页面中，可以直接在页面中编写 JavaScript 程序代码。即通过 <script> 标签的 language 属性，来指定页面中所使用的脚本编写语言，具体的格式如下：

```
<script language="JavaScript" >
JavaScript 脚本代码
</script>
```

下面通过一个例子来说明页面中直接定义 JavaScript 代码的应用，如实例 1.1 所示。

实例 1.1 页面中直接定义 JavaScript 代码。其代码如下：

```
<html>
<head>
<title>页面中直接定义 JavaScript 程序</title>
</head>
<body>
<h2>
<script language="JavaScript">
    var str="第一个 JavaScript";
    document.write(str);
</script>
</h2>
</body>
</html>
```

将页面保存为 firstscript.htm 文件，在浏览器中执行，结果如图 1-2 所示。

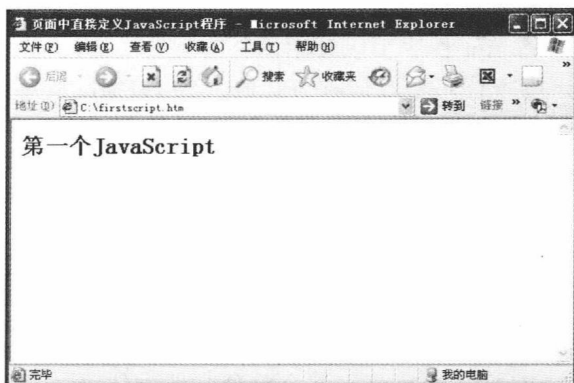


图 1-2 页面中直接定义 JavaScript 代码执行结果

 **说明：** 对于小的脚本和基本 HTML 页面来说，将 JavaScript 程序直接包括进 HTML 文件是很方便的。但当页面需要长且复杂的脚本时，这样做会有些麻烦。

1.2.2 链接外部 JavaScript 文件

为使 HTML 文件以及 JavaScript 脚本的开发和维护更为容易，JavaScript 规范建议用户将 JavaScript 脚本保存于单独的文件中，并利用<script>标签的 src 属性把外部 JavaScript 文件链接到 HTML 文件中。

 **说明：** 外部文件的扩展名应是.js。尽管这不是强制的，但使用该扩展名便于确定每个文件包含的内容。

使用<script>标签的 src 属性的具体格式如下：

```
<script Language="JavaScript" src="aaa.js">
</script>
```

例如，假定创建了一个文件 MyCommonFunctions.js，该文件与 HTML 页面位于同一个目录下，则为了把该外部 JavaScript 文件链接到网页上，应使用以下<script>标记：

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

浏览器会读取这行代码，并把文件的内容作为网页的一部分包含进来。链接外部文件时，不能在<script>标记中放置任何 JavaScript 代码。例如，下面的代码是无效的：

```
<script type="text/javascript" src="MyCommonFunctions.js">
var myVariable;
if ( myVariable == 1 )
{
// do something
}
</script>
```

下面通过一个例子来说明链接外部 JavaScript 文件的应用。

实例 1.2 链接外部 JavaScript 文件。

(1) 创建外部 JavaScript 文件 aaa.js。其代码如下：

```
<!--  
    document.write("这段文字来源于外部文件 aaa.js.");  
-->
```

在使用 JavaScript 脚本时要注意，并不是所有的浏览器都可以正确解析 JavaScript 脚本。因此，在那些浏览器中编写的脚本将处于一种不是被解析而是直接显示的危险，解决的方法是将所有的脚本放在 HTML 注释中来避免这种事件的发生。在代码中通过<!-- -->标识进行注释，若不能够解析 JavaScript 代码的浏览器，则所有在其中的代码均被忽略；若能够解析，则执行并显示其结果。

(2) 创建 HTML 文件。其代码如下：

```
<html>  
<head>  
<title>链接外部 JavaScript 文件</title>  
</head>  
<body>  
<script language="JavaScript" src="aaa.js">  
</script>  
</body>  
</html>
```

将页面保存为 linkscript.htm 文件，在浏览器中执行，结果如图 1-3 所示。



图 1-3 链接外部 JavaScript 文件的执行结果



说明： 使用独立的.js 文件来编写 JavaScript 脚本可以为程序带来很大的方便。实际操作时，可以将经常使用的小程序编写成.js 文件，然后可以在程序中调用。修改这些.js 文件也很方便，而不必在页面的每一处去修改源代码。

1.2.3 事件调用 JavaScript 程序

之前介绍的 JavaScript 代码，在 HTML 文件中都是通过直接调用方式嵌入到页面中

的。此外，还可以在 HTML 标签的事件中调用 JavaScript 程序，如 onclick 事件、onload 事件等。

下面通过一个例子来说明如何通过事件来调用 JavaScript 程序。

实例 1.3 事件调用 JavaScript 程序。其代码如下：

```
<html>
<head>
<title>an onload script</title>
<script language="javascript">
<!--
function done(){
alert("the page has finished loading.")
}
-->
</script>
</head>
<body onload="done()">
  页面原始内容不包含任何脚本语句。
</body>
</html>
```

将页面保存为 load.htm 文件，在浏览器中执行，结果如图 1-4 所示。



图 1-4 事件调用 JavaScript 程序的执行结果



说明： 还有一种简约的格式可以使事件调用 JavaScript 程序，如在链接标记中使用 `<a href="javascript:alert('Hello World') ">Click me</a>` 调用。

1.3 JavaScript 代码的编辑工具

因为 JavaScript 只是纯文本，所以可以使用几乎任何文本编辑器来编辑 JavaScript。JavaScript 文件必须是纯文本格式，只有这样 Web 浏览器才能理解它们。本节将介绍几种比较常用的编写 JavaScript 的工具。