



高职高专计算机技能型紧缺人才培养规划教材

计算机多媒体技术专业

主编 张基宏 耿 壮

JavaScript 脚本程序设计

吴以欣 陈小宁 编

免费提供

教学相关资料

 人民邮电出版社
POSTS & TELECOM PRESS

高职高专计算机技能型紧缺人才培养规划教材
计算机多媒体技术专业

JavaScript 脚本程序设计

吴以欣 陈小宁 编 ←

人民邮电出版社

图书在版编目 (CIP) 数据

JavaScript 脚本程序设计 / 吴以欣, 陈小宁编. —北京: 人民邮电出版社, 2005.5

ISBN 7-115-13323-9

I. J... II. ①吴... ②陈... III. JAVA 语言—程序设计—高等学校: 技术学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 033120 号

内 容 提 要

本书全面系统地介绍了用 JavaScript 语言制作网页的客户端编程技术方法。主要分为 4 大部分: JavaScript 的特点及基本编程方法, JavaScript 用于处理浏览器对象, JavaScript 实用技巧, JavaScript 综合实训项目。读者通过本书的学习, 可以制作出自己的动态网页, 全面提高自己网页设计的基本知识和基本技能。本书为高职高专多媒体技术专业的教材, 也可以作为一般网页制作人员的自学用书。

高职高专计算机技能型紧缺人才培养规划教材

计算机多媒体技术专业

JavaScript 脚本程序设计

-
- ◆ 编 吴以欣 陈小宁
责任编辑 潘春燕
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67170985
北京通州大中印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 376 千字
印数: 1-5 000 册
- 2005 年 5 月第 1 版
2005 年 5 月北京第 1 次印刷

ISBN 7-115-13323-9/TP · 4621

定价: 21.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

高职高专计算机技能型紧缺人才培养
规划教材编委会

主 任 武马群

副 主 任 王泰峰 徐民鹰 王晓丹

编 委 (以姓氏笔画为序)

马 伟 安志远 向 伟 刘 兵 吴卫祖 吴宏雷
余明辉 张晓蕾 张基宏 贺 平 柳 青 赵英杰
施晓秋 姜 锐 耿 壮 郭 勇 曹 炜 蒋方纯
潘春燕

丛书出版前言

目前,人才问题是制约我国软件产业发展的关键。为加大软件人才培养力度和提高软件人才培养质量,教育部继在2003年确定北京信息职业技术学院等35所高职院校试办示范性软件职业技术学院后,又同时根据《教育部等六部门关于实施职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》(教职成[2003]5号)的要求,组织制定了《两年制高等职业教育计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》。示范性软件职业技术学院与计算机应用与软件技术专业领域技能型紧缺人才培养工作,均要求在较短的时间内培养出符合企业需要、具有核心技能的软件技术人才,因此,对目前高等职业教育的办学模式和人才培养方案等做较大的改进和全新的探索已经成为学校的当务之急。

据此,我们认为做一套符合上述一系列要求的切合学校实际的教学方案尤为重要。遵照教育部提出的以就业为导向,高等职业教育从专业本位向职业岗位和就业为本转变的指导思想,根据目前高等职业教育院校日益重视学生将来的就业岗位,注重培养毕业生的职业能力的现状,我们联合北京信息职业技术学院等几十所高职院校和普拉内特计算机技术(北京)有限公司、福建星网锐捷网络有限公司、北京索浪计算机有限公司等软件企业共同组建了计算机应用与软件技术专业领域技能型紧缺人才培养教学方案研究小组(以下简称研究小组)。研究小组对承担计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的79所院校的专业设置情况做了细致的调研,并调查了几十所高职院校计算机相关专业的学生就业情况以及目前软件企业的人才市场需求状况,确定首批开发目前在高职院校开设比较普遍的计算机应用、计算机网络、多媒体技术和计算机软件等4个专业方向的教学方案。

同时,为贯彻教育部提出的要与软件企业合作开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的精神,使高等职业教育培养出的软件技术人才符合企业的需求,研究小组与许多软件企业的专家们进行了反复研讨,了解到目前高职院校的毕业生的实际动手能力和综合应用知识方面较弱,他们和企业需求的软件人才有着较大的差距,到企业后不能很快独挡一面,企业需要投入一定的成本和时间进行项目培训。针对这种情况,研究小组在教学方案中增加了“综合项目实训”模块,以求强化学生的实际动手能力和综合应用前期所学知识的能力,探索将企业的岗前培训内容前移到学校的教学中的实验之路,以此增强毕业生的就业竞争力。

在上述工作的基础上,研究小组于2004年多次组织召开了包括企业专家、教育专家、学校任课教师在内的各种研讨会和方案论证会,对各个专业按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”一步步进行了认真的分析和研讨:

- 列出各专业的岗位群及核心技能。针对教育部提出的以就业为导向,根据目前高职高专院校日益关心学生将来的就业岗位的现状,在前期大量调研的基础上,首先提炼各个专业的岗位群。如对某专业的岗位群进行研究时,首先罗列此专业的各个岗位,以便能正确了解每个岗位的职业能力,再根据职业能力进行有意义的合并,形成各个专业的岗位群,再对每

个岗位群总结和归纳出其核心技能。

- 根据岗位群及核心技能做出教学方案。在岗位群及核心技能明确的前提下，列出此岗位应该掌握的知识点，再依据这些知识点推出应该学习的课程、学时数、课程之间的联系、开课顺序并进行必要的整合，最终形成一套科学完整的教学方案。

为配合学校对技能型紧缺人才的培养工作，在研究小组开发上述4个专业的教学方案的基础上，我们组织编写了这套包含计算机软件、计算机网络、多媒体及计算机应用等4个专业的教材。本套教材具有以下特点：

- 注重专业整体策划的内涵。对各专业系列教材按照“岗位群→核心技能→知识点→课程设置→各课程应掌握的技能→各教材的内容”的思路组织开发教材。

- 按照“理论够用为度”的原则，对各个专业的基础课进行了按需重新整合。

- 各专业教材突出了实训的比例，注重案例教学。每本教材都配备了实验、实训的内容，部分专业的教材配备了综合项目实训，使学生通过模拟具体的软件开发项目了解软件企业的运行环境，体验软件的规范化、标准化、专业化和规模化的开发流程。

为了方便教学，我们免费为选用本套教材的老师提供部分专业的整体教学方案及教学相关资料。

- 所有教材的电子教案。

- 部分教材的习题答案。

- 部分教材中实例制作过程中用到的素材。

- 部分教材中实例的制作效果以及一些源程序代码。

本套教材以各个专业的岗位群为出发点，注重专业整体策划，试图通过对系列教材的整体构架，探索一条培养技能型紧缺人才的有效途径。

经过近两年的艰苦探索和工作，本套教材终于正式出版了，我们衷心希望，各位关心高等职业教育的读者能够对本套教材的不当之处给予批评指正，提出修改意见，也热切盼望从事高等职业教育的教师以及软件企业的技术专家和我们联系，共同探讨计算机应用与软件技术专业的教学方案和教材编写等相关问题。来信请发至 panchunyan@ptpress.com.cn。

编者的话

JavaScript 是一种解释性的、基于对象的、用于网页设计和制作的客户端程序开发语言，由于它简单、易于理解、易于使用，因此，已成为网页制作中最为流行的脚本式程序语言，将它与浏览器的文档对象模型（DOM）结合使用，可以制作出功能强大的动态网页。

本书主要分为 4 大部分：第 1 部分主要介绍了 JavaScript 的特点及编程基础（第 1 章、第 2 章和第 3 章）；第 2 部分主要介绍了 JavaScript 用于浏览器的对象模型（第 4 章和第 5 章）；第 3 部分主要介绍了 JavaScript 用于动态网页及其他实用技术（第 6 章和第 7 章）；第 4 部分是 JavaScript 的综合实训项目。

本书由浅入深，配有大量的实例，所有实例都可用于当前最为流行的两大浏览器——Microsoft Internet Explorer 浏览器和 Netscape 的 Navigator 浏览器较高级版本。本书不仅全面介绍了 JavaScript 的基本编程技术及 JavaScript 的最新版本的功能，而且对于 JavaScript 在不同浏览器中的不同应用技术也进行了一些介绍。

本书实例中的大部分为作者在工作中的实际应用，所有实例均经过验证。计划学时为全时学习的 54 小时，如果能增加实验室上机实训，将取得更好的效果。读者通过本书的学习及大量的示例和实训项目的练习，可以很快地通过 JavaScript 编程进行网页的动态设计。

虽然本书的学习并不要求读者具备大量的编程经验，但是读者如果有使用 HTML 创建过一些网页的经历，将会有助于对本书的学习和理解。

作者电子邮件地址是 syndia_wu@yahoo.com，欢迎读者提出建议和意见。

编者

2005 年 2 月

目 录

第 1 章 JavaScript 简介	1
1.1 什么是 JavaScript.....	1
1.1.1 JavaScript 的发展历史.....	1
1.1.2 JavaScript 的特点.....	2
1.1.3 JavaScript 的作用.....	4
1.2 使用 JavaScript 的环境要求.....	8
1.2.1 硬件要求.....	8
1.2.2 软件要求.....	9
1.3 JavaScript 示例.....	11
1.3.1 编写 JavaScript.....	11
1.3.2 运行 JavaScript 程序.....	14
1.3.3 调试 JavaScript 程序.....	15
习题.....	17
第 2 章 JavaScript 编程基础	19
2.1 数据类型及变量.....	19
2.1.1 数据类型.....	19
2.1.2 常量与变量.....	20
2.2 表达式与运算符.....	23
2.2.1 表达式.....	23
2.2.2 运算符.....	23
2.3 基本语句.....	29
2.3.1 注释语句.....	29
2.3.2 赋值语句.....	30
2.3.3 流程控制语句.....	30
2.4 函数.....	35
2.4.1 定义函数.....	37
2.4.2 使用函数.....	38
2.4.3 函数的参数.....	38
2.5 对象.....	40
2.5.1 什么是对象.....	40
2.5.2 定义对象.....	42
2.5.3 使用对象.....	43

2.6 事件及事件处理程序	44
2.6.1 网页中的事件	45
2.6.2 用 JavaScript 处理事件	46
习题	47
第 3 章 JavaScript 常用内置对象	49
3.1 数组 (Array) 对象	49
3.2 字串 (String) 对象	55
3.3 数学 (Math) 对象	63
3.4 日期 (Date) 对象	73
习题	81
第 4 章 JavaScript 常用文档对象	83
4.1 文档 (document) 对象	83
4.1.1 文档对象的常用属性、方法与事件	84
4.1.2 cookie 属性	87
4.2 窗体 (form) 及其元素对象	89
4.2.1 窗体对象	89
4.2.2 窗体中的元素对象	91
4.3 锚点 (anchor) 与链接 (link) 对象	102
4.3.1 锚点对象	103
4.3.2 链接对象	104
4.4 图像 (image) 对象	106
4.4.1 网页中已有的图像对象	106
4.4.2 网页中新建图像对象	107
4.4.3 图像对象的应用实例	107
习题	112
第 5 章 JavaScript 其他常用窗口对象	114
5.1 屏幕 (screen) 对象	114
5.2 浏览器信息 (navigator) 对象	115
5.3 窗口 (window) 对象	116
5.3.1 窗口对象的常用属性和方法	116
5.3.2 多窗口控制	117
5.3.3 输入/输出信息	125
5.4 网址 (location) 对象	126
5.4.1 网址对象的常用属性和方法	126
5.4.2 网址对象的应用实例	127
5.5 历史记录 (history) 对象	128

5.5.1	历史对象的常用属性和方法	128
5.5.2	历史对象的应用实例	129
5.6	框架 (frame) 对象	130
5.6.1	框架对象的常用属性和方法	130
5.6.2	框架对象的应用实例	131
	习题	140
第 6 章	JavaScript 与动态网页	142
6.1	网页对象模型与网页元素对象	142
6.1.1	HTML 对象模型方法	142
6.1.2	DHTML 对象模型方法	143
6.1.3	W3C 文档对象模型方法	144
6.2	动态网页内容	145
6.2.1	DHTML 对象模型方法	145
6.2.2	W3C 文档对象模型方法	148
6.3	动态网页样式	151
6.3.1	网页样式表简介	151
6.3.2	样式的语法规则	155
6.3.3	层叠式应用规则	156
6.3.4	改变网页元素对象的样式属性	157
6.3.5	改变网页样式表	159
6.3.6	改变外部网页样式文件	161
6.4	动态定位网页对象	161
6.4.1	绝对位置与相对位置	161
6.4.2	z-index	163
6.4.3	坐标位置及尺寸	163
6.4.4	display 与 visibility	166
	习题	168
第 7 章	JavaScript 实用技巧	170
7.1	建立函数库	170
7.2	识别浏览器的方法	176
7.2.1	浏览器信息对象的方法	176
7.2.2	浏览器功能识别的方法	177
7.3	校验用户输入	178
7.4	弹出窗口	182
7.4.1	一般的弹出窗口	182
7.4.2	对话框式的弹出窗口	182
7.4.3	窗口中的“窗口”	185

7.5	下拉菜单	188
7.6	事件冒泡处理	199
7.7	动画技术	200
7.7.1	动画网页对象的内容	201
7.7.2	动画网页对象的尺寸	202
7.7.3	动画网页对象的位置	204
	习题	207
第 8 章	实训	209
8.1	实训 1 JavaScript 基本操作	209
8.2	实训 2 JavaScript 编程基础	211
8.3	实训 3 JavaScript 常用内置对象	217
8.4	实训 4 JavaScript 常用文档对象	224
8.5	实训 5 JavaScript 其他常用窗口对象	229
8.6	实训 6 JavaScript 与动态网页	233
8.7	实训 7 JavaScript 实用技巧	238

第 1 章

JavaScript 简介

本章主要介绍 JavaScript 的一些基础知识。通过本章的学习，读者应该掌握以下内容：

- 什么是 JavaScript
- 使用 JavaScript 的环境要求

1.1 什么是 JavaScript

1.1.1 JavaScript 的发展历史

JavaScript 是 1995 年由美国 Netscape 公司的布瑞登·艾克 (Brendan Eich) 为 Navigator 2.0 浏览器的应用而发明的。它是写在 HTML (超文本标记语言) 文件中的一种基于对象和事件驱动并具有安全性能的脚本语言，当用户在客户端的浏览器中显示该网页时，浏览器就会执行 JavaScript 程序，让用户通过交互式的操作变换网页显示的内容，以实现 HTML 语言所不能实现的一些功能。例如，当鼠标移过某一图片时，图片会产生翻转 (如图 1-1 所示)；当用户在文字框中输入文字后，校验输入的内容，并且产生警告信息；当用户点击某一按钮后，改变网页中的某一区域文字的颜色和大小等。

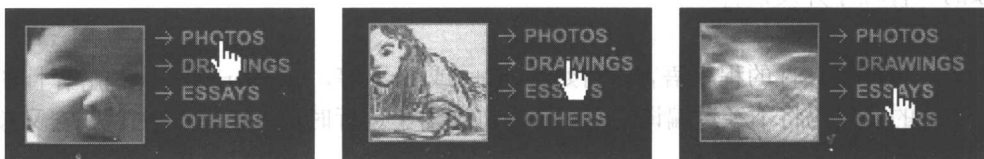


图 1-1 JavaScript 应用实例

当 JavaScript 在 Navigator 2.0 浏览器中成功地应用后不久，美国 Microsoft (微软) 公司也推出了用于 Internet Explorer 浏览器中的、类似 JavaScript 的程序语言，其注册商标名称为 JScript。从此以后，这两个当今最为流行的浏览器——Netscape 的 Navigator 浏览器和 Microsoft 的 Internet Explorer 浏览器，在不断提高浏览器版本的同时，也不断地更新其所用的脚本程序的版本，表 1-1 列出了它们的主要版本变化。

1999 年，作为国际信息和通信系统标准权威的欧洲计算机制造协会 (ECMA)，在 Netscape JavaScript 1.5 版本的基础上制定了“ECMAScript 程序语言的规范书”，又称为“ECMA-262 标准”，该标准已被国际标准组织 (ISO) 采纳通过，作为各种浏览器生产开发所使用的脚本程序的统一标准。实际上，该规范书制定的是上述两种脚本程序语言的“最小兼容性”标准，

也就是说, Netscape 的 Navigator 浏览器将继续使用其注册商标名“JavaScript”的脚本程序, 同时 Microsoft 的 Internet Explorer 浏览器也将继续使用其注册商标名“JScript”的脚本程序, 并且, 它们各自仍然保持着原有的不同于“ECMA-262 标准”的特性。但是, 上述两大公司都已许诺, 从此以后开发的新功能将会遵循 ECMA-262 标准。

表 1-1 两大浏览器及其 JavaScript 版本的比较

计算机公司	浏览器	浏览器版本	JavaScript 版本
Netscape	Navigator	2.x	JavaScript 1.0
Netscape	Navigator	3.x	JavaScript 1.1
Netscape	Navigator	4.0x	JavaScript 1.2
Netscape	Navigator	4.5x	JavaScript 1.3
Netscape	Navigator	4.7x	JavaScript 1.4
Netscape	Navigator	6.x	JavaScript 1.5
Netscape	Navigator	7.x	JavaScript 1.5
Microsoft	Internet Explorer	3.x	JScript 1.x, 3.x, 5.x
Microsoft	Internet Explorer	4.x	JScript 3.x, 5.x
Microsoft	Internet Explorer	5.x	JScript 5.x
Microsoft	Internet Explorer	5.5x	JScript 5.5
Microsoft	Internet Explorer	6.x	JScript 5.6

本书主要介绍的是用于浏览器的通用性脚本式语言, 因此, 本书将使用“JavaScript”一词同时代表上述两种程序语言。

1.1.2 JavaScript 的特点

由上述 JavaScript 的发展历史可以看出, JavaScript 是一种解释性的、用于客户端的、基于对象的一种程序开发语言。

1. 解释性

不同于一些编译性的程序语言, 例如 C、C++ 或 Java 等, JavaScript 是一种解释性的程序语言, 即它的源代码将不经过编译, 而直接在浏览器中运行时被“翻译”, 因此, 它又被称做“脚本式”语言。

由于 JavaScript 的这一特点, 在编写 JavaScript 程序时, 将很难预计运行程序所使用的硬件环境、操作系统以及浏览器等, 因此, 当开发 JavaScript 程序时, 不应该仅使用其最高版本的特性和功能, 还必须考虑到少数人有可能仍然在使用旧版本的浏览器。

另外, 考虑到现在所编写的程序有可能将会使用 5 年或 10 年, 因此, 在编写 JavaScript 程序的过程中, 应尽量兼容各种硬件环境、各种操作系统以及各种不同浏览器的功能和特性, 这样才能够编写出真正实用的、跨平台、跨浏览器的 JavaScript 应用程序。

2. 用于客户端

JavaScript 包含有服务器端应用和客户端应用两个方面, 其中客户端的应用更为广泛, 它也是本书的主要介绍内容。

当 JavaScript 程序用于运行在用户浏览器中时, 它就被称做是“客户端”的程序。明确这一点将有助于我们编写 JavaScript 程序时的构思。

如图 1-2 所示为 Internet 网页显示过程的示意图, 当一个网页在浏览器中通过网址申请后, 服务器端的程序根据用户的申请, 与数据库之间进行存取数据的操作, 然后将用户所需的数据送回浏览器。从上述过程中可以看出, 用于客户端的 JavaScript 不同于运行在服务器端的程序, 例如 Java、.Net 等常用于存取用户所需的数据内容, 而客户端的程序则用于用户输入数据的校验、根据用户的操作改变网页的画面或者进行动画处理等“界面”性的工作。

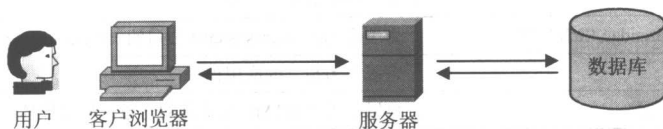


图 1-2 Internet 网页显示过程

3. 基于对象

JavaScript 程序语言是一种基于对象的程序设计语言, 它将显示在浏览器网页中的任何一种元素。例如按钮、文字框、图像等, 都作为“对象”处理, 而网页中各元素之间的关系, 都被描述为各“对象”的层次结构关系, 这种关系被称为“文档对象模型”(DOM)。

如图 1-3 所示为一般浏览器网页的文档对象模型结构图。实际上, JavaScript 程序的核心就是用其基本编程方法对浏览器的文档对象模型进行处理, 使网页中的各元素不再是“静态”不变的元素, 而是可以根据用户的不同要求“动态”地显示出来, 所以 JavaScript 程序是制作动态网页 (DHTML) 的基本工具之一, 这部分内容将在第 6 章中进行详细说明。

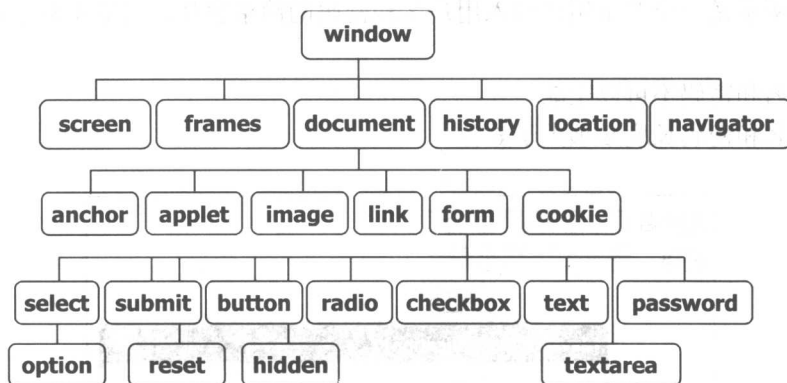


图 1-3 浏览器网页的文档对象模型

4. 与 Java 的比较

在 Netscape 公司发明 JavaScript 的初期, 围绕着选择“什么样的程序语言作为开发浏览器”这一问题, 曾经引起过很大的争议, 其中一方的意见是使用 Java 作为开发浏览器程序的工具, 因为它功能强大, 并且该语言已经发展得较为成熟。但是更多的人赞同使用“脚本式”的语言进行开发, 因为其最大的优点是易学易用, 是一种“轻量级”的程序语言, 因此, 最终出现了“JavaScript”——与 Java 名称很像的脚本式语言。

由于这两种程序设计语言名字相像, 它们使用的编程语法结构也有许多相似之处, 并且它们都是用于 Internet 的应用, 而这些应用又都使用浏览器, 因此很容易被初学者混淆。表

1-2 列出了这两种程序语言的主要区别。

表 1-2 JavaScript 与 Java 的比较

JavaScript	Java
在客户端运行时被解释	由编写者编译后变成机器码，运行在服务器端或客户端
程序源代码嵌入在 HTML 文件中	由 Java 开发的 Applets 与 HTML 无关
没有严格的数据类型	具有严格的数据类型
由美国 Netscape 公司的 Brendan Eich 发明	由美国 Sun Microsystems 公司的 James Gosling 发明
只能在浏览器中应用	可以作为独立的应用程序
只作用于 HTML 的对象元素	可以作用于 HTML 元素外的对象，如多媒体

1.1.3 JavaScript 的作用

在 Internet 的浏览器中运行 JavaScript，主要目的是用于在客户端动态地、与用户交互式地完成一些 HTML 文件所不能实现的功能。本书将在以后的各章节中详细介绍下述 JavaScript 的常用功能。

1. 校验用户输入的内容

对于一般 Internet 的应用软件，用户输入内容的校验常分为两种：格式性校验和功能性校验。其中，功能性校验常常与服务器端的数据库相关联，因此，这种校验必须将网页窗体提交到服务器端后才能进行；而格式性校验可以只发生在客户端，即在窗体提交到服务器端之前完成。JavaScript 常用于对于用户输入的格式性校验。

如图 1-4 所示是一个要求用户输入用户名和密码的简单窗体，它要求对用户的输入进行下述校验。

- (1) 用户名和密码不可以空缺。
- (2) 用户名和密码长度必须大于 6 位。

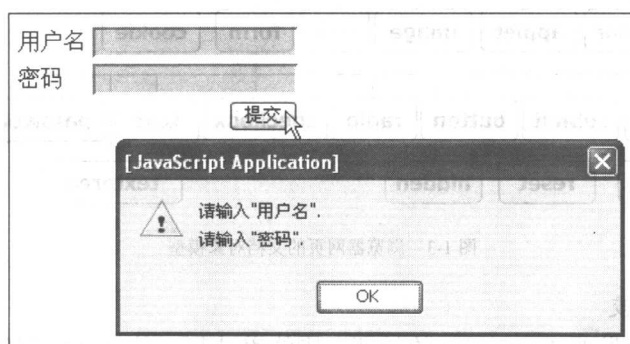


图 1-4 校验用户输入的内容

- (3) 用户名和密码必须是有效字符串，例如，用户输入的内容中不可以包含“#”等特殊字符。
- (4) 密码中必须包含有字符和数字。
- (5) 用户名必须有效，即服务器端的数据库中存在有该用户名。

上述各项校验中,前4项属于格式性的校验,可以由 JavaScript 来完成,第5项则必须通过表单提交后,由服务器端的程序进行校验。

有关校验用户输入内容的具体制作方法,详见第4.2节及第7.3节。

2. 有效地组织网页内容

当网页中需要用户输入的窗体内容很多时,有效地编排、组织用户输入内容,尽量减少用户选项,制作友好的用户界面,是网页设计的一个主要内容,JavaScript 常用于完成这一任务。

例如,对于如图1-5所示的表单,如果用户选择“长方形”项,则要求用户输入“长度”、“宽度”和“颜色”域的内容;如果用户选择“正方形”项,则隐藏“宽度”域,并且,在这种状态中,如果用户选择“提交”按钮提交表单时,“宽度”域的变量名将不会传递给服务器端的程序。

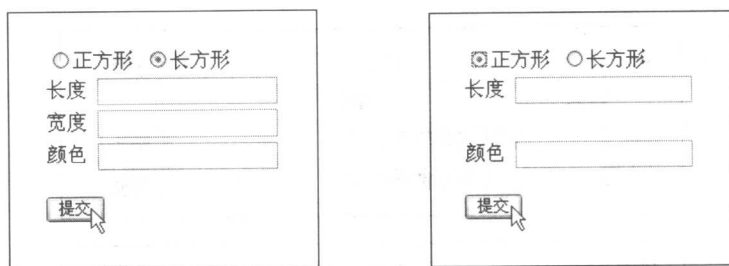


图 1-5 根据用户的选择显示不同的输入域

上述示例的相关内容及其制作方法,详见第6.4节。

3. 动态地显示网页内容

JavaScript 常用于完成不用通过服务器端处理、仅在客户端动态显示网页内容的功能,这样既可以节省网页与服务器端之间的通信,又可以制作出便于用户使用的友好界面。

例如,在网页的一角动态地显示时钟,或显示离指定的日期还差多少天等(详见第3.4节及第7.6节);在两个列表之间移动元素(如图1-6所示,详见第4.2节);当用户在多行文本框中输入内容时,用户每输入一个文字,其上方自动显示文字的个数(如图1-7所示);动态地在表格编辑域中添加、删除记录(如图1-8所示,详见第6.2节)等。

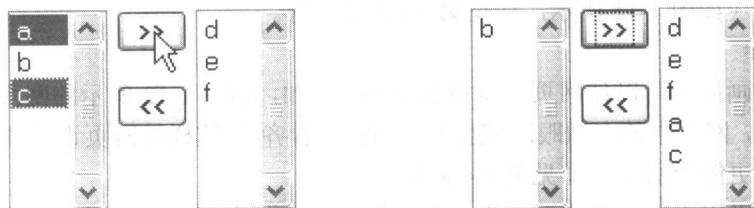


图 1-6 在两个列表之间移动元素

4. 弥补静态网页不能实现的功能

通过 JavaScript 还可以实现一些静态网页不能实现的功能。

例如,当显示多列多行内容的大型表格时,需要像 Microsoft Excel 软件中的“冻结”功

您已输入了 7 个文字 (请小于50个文字) :

基本情况如下:

提交

图 1-7 自动显示用户输入的文字个数

姓名	单位	新增
姓名	单位	
张三	北京欣光软件公司	删除
李四	上海南市食品公司	删除

姓名	王五	单位	天津第四棉纺厂	新增
姓名	单位			
张三	北京欣光软件公司	删除		
李四	上海南市食品公司	删除		
王五	天津第四棉纺厂	删除		

图 1-8 动态编辑表格域

能一样将表头及左边第一列 (或数列) 固定住, 用户在滚动表格内容时, 可以方便地查看表头的内容 (如图 1-9 所示); 对于一些特殊的网页, 如网上测验网页, 不允许用户通过浏览器的工具栏或快捷菜单中的“back (返回)”项返回到上一页 (如图 1-10 所示, 详见第 5.5 节); 对于弹出窗口, 需要将焦点集中在弹出窗口上, 也就是说, 在弹出窗口关闭之前, 用户不能在底窗口中操作 (如图 1-11 所示, 详见第 7.4.2 节) 等。

5. 动画显示

网页中的动画显示可以使网页显得更加生动, 例如, 如图 1-12 所示的网页中, 当光标移向不同的链接时, 网页中的卡通眼珠就会向该链接“看齐”, 这样的网页设计往往比单纯显示网页的文字链接更能吸引人 (详见第 4.4.3 节)。

动画技术不仅可以用于网上游戏、广告、演示等网页的制作, 还可以用于一般应用网页的制作。例如, 当网页的内容很多时, 可以显示一段“请等候……”的动画信息, 或显示一个下载过程的进度条 (如图 1-13 所示); 对于网上测试的网页, 显示一个倒计时或倒计数的计时器或计数器等。有关动画显示的具体制作方法, 详见第 7.6 节。