



职业教育新技术丛书 · 移动应用



移动网页设计

(基于jQuery Mobile)

◎ 赵增敏 主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

职业教育新技术丛书·移动应用

移动网页设计

(基于jQuery Mobile)

◎ 赵增敏 主 编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

jQuery Mobile 是一套移动应用界面开发框架,它通过网页形式来呈现类似于移动应用的用户界面,旨在创建使智能手机、平板电脑和台式机设备都能访问的响应式移动网站和应用程序。为了满足社会和企业对人才的需求,在网页设计课程中适时引入 jQuery Mobile 移动网页设计已是势在必行。本书通过大量实例系统地讲述了 jQuery Mobile 移动网页设计的方法和技巧。本书共 9 章,主要内容包括:移动网站开发基础,页面与对话框,按钮与弹出窗口,工具栏与导航栏,网格、表格和列表视图,面板、可折叠块和筛选器, jQuery Mobile 表单, jQuery Mobile 事件,综合设计实例等。

本书坚持“以就业为导向、以能力为本位”的原则,突出实用性、适用性和先进性,结构合理,论述准确,内容翔实,注意知识的层次性和技能培养的渐进性;遵循难点分散的原则,合理安排各章的内容,降低学生的学习难度;通过丰富的实例来引导读者学习,旨在培养他们的实践动手能力和创新精神。每章后面均配习题和上机操作。

本书可作为职业院校计算机类专业的教材,也可作为网页设计人员、网站开发和维护人员的参考书。

未经许可,不得以任何方式复制或抄袭本教程之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

移动网页设计:基于 jQuery Mobile / 赵增敏主编. —北京:电子工业出版社, 2017.12

ISBN 978-7-121-32821-3

I. ①移… II. ①赵… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 244035 号

策划编辑:关雅莉

责任编辑:杨 波

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:20 字数:512 千字

版 次:2017 年 12 月第 1 版

印 次:2017 年 12 月第 1 次印刷

定 价:39.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254617, luomn@phei.com.cn。

前言

近年来,随着移动互联网的迅速发展,智能手机和平板电脑等移动设备的应用日益普及,网页设计和网站开发正在向移动端迁移,在传统的网页设计课程中适时地引入移动网页设计已是势在必行。由于移动设备屏幕面积小,而且以触摸操作为主,因此移动版网页与PC桌面版网页有明显的区别,直接使用HTML5、CSS3和JavaScript进行移动网站开发存在一定的难度,通常需要引入合适的开发框架作为移动网页设计的基础。

jQuery Mobile是一套基于jQuery的移动应用界面开发框架,该框架为移动设备在网络传输、页面呈现、用户行为交互等方面都进行了一些特别的优化,可以通过网页形式呈现类似于移动应用的用户界面,旨在创建使智能手机、平板电脑和台式机设备都能访问的响应式移动网站和应用程序;基于jQuery Mobile开发的移动网站还可以发行为本地移动App,直接运行于Android或iOS平台之上。

本书以jQuery Mobile 1.4.5为蓝本,系统地讲述了jQuery Mobile移动网页设计的方法和技巧。全书共9章。第1章介绍移动网站开发所需要的基础知识,内容包括创建HTML5网页,使用jQuery简化编程及使用jQuery Mobile支持移动开发;第2章介绍页面与对话框,内容包括移动网页结构、页面链接、页面预加载和缓存、对话框页面、页面之间传递参数以及页面加载信息;第3章讲述按钮与弹出窗口的应用,先讨论如何创建和设置按钮,然后讨论如何创建和应用弹出窗口;第4章介绍工具栏与导航栏的应用,先讨论如何创建和设置工具栏,然后讨论如何创建和应用导航栏;第5章讲述网格、表格和列表视图的应用,首先介绍如何创建网格布局,然后讨论如何创建响应式表格,最后详细讲解如何创建和应用列表视图;第6章讲述面板、可折叠块和筛选器的应用,首先介绍如何创建和应用面板,然后讨论如何创建和应用可折叠块,最后讲解如何创建和应用筛选器;第7章介绍jQuery Mobile表单的应用,内容包括表单基础知识,创建文本输入框,单选按钮与复选框,选择菜单以及滑块、范围滑块和翻转开关;第8章讨论各种类型的jQuery Mobile事件及其应用,内容包括页面事件、触摸事件、用户操作事件以及虚拟鼠标事件,作为前面各章所讲知识的综合应用;第9章给出一个综合设计实例,内容包括系统功能设计,系统功能实现以及本地安装包制作。

本书实例中用到的一些人名、电话号码、QQ号码以及电子邮件地址均属虚构,如有雷同,实属巧合。

本书由赵增敏担任主编。参加本书编写的还有朱粹丹、赵朱曦、余霞、吴洁、姜红梅、卢捷、王庆建、胡婷婷、连静、刘颖、李娴、王亮、段丽霞、彭辉、王永锋、王静、郭宏等。由于水平所限,书中疏漏和错误之处在所难免,恳请广大读者提出宝贵意见。

为了方便教师教学，本书还配有教学指南、电子教案和习题答案（电子版）。请有此需要的教师登录华信教育网（<http://www.hxedu.com.cn>）免费注册后进行下载。

编 者





第 1 章 移动网站开发基础 001

1.1 创建 HTML5 网页 001

1.1.1 用 HTML 语言创建内容 001

1.1.2 用 CSS 设置样式 005

1.1.3 用 JavaScript 实现功能 007

1.2 使用 jQuery 简化编程 009

1.2.1 在网页中引用 jQuery 010

1.2.2 jQuery 构造函数 011

1.2.3 jQuery 选择器 013

1.2.4 DOM 操作 021

1.2.5 动画效果 025

1.2.6 事件处理 027

1.2.7 Ajax 应用 031

1.3 使用 jQuery Mobile 支持移动设备 033

1.3.1 jQuery Mobile 功能特点 033

1.3.2 jQuery Mobile 的平台兼容性 034

1.3.3 获取 jQuery Mobile 框架 035

1.3.4 配置 jQuery Mobile 开发环境 037

1.3.5 实现 jQuery Mobile 移动开发 037

习题 1 039

上机操作 1 041

第 2 章 页面与对话框 042

2.1 移动网页结构 042

2.1.1 单页面结构 042

2.1.2 多页面结构 044

2.2 页面链接 047

2.2.1 Ajax 导航 047

2.2.2 创建后退链接 050

2.2.3 设置页面切换动画 052

2.3 页面预加载和缓存 054

2.3.1 页面预加载 054

2.3.2 页面缓存 056

2.4 对话框页面 058

2.4.1 打开对话框 058

2.4.2 关闭对话框 061

2.4.3 设置对话框切换动画 063

2.5 页面之间传递参数 065

2.5.1 基于 HTTP GET 方式传递参数 065

2.5.2 基于 Web 存储传递参数 068

2.6 页面加载信息 070

2.6.1 设置页面加载信息 070

2.6.2 设置加载错误信息 072

习题 2 074

上机操作 2 075

第 3 章 按钮与弹出窗口 077

3.1 按钮 077

3.1.1 创建按钮 077

3.1.2 内联按钮 078

3.1.3 微型按钮 081

3.1.4 禁用按钮 082

3.1.5 组合按钮 084

3.1.6 设置按钮样式 087

3.1.7 设置按钮图标 088

3.2 弹出窗口 094

3.2.1 创建和显示弹出窗口 094

3.2.2 关闭弹出窗口 095

3.2.3 定位弹出窗口 096

3.2.4 设置弹出窗口切换动画 098

3.2.5 创建带箭头的弹出窗口 101

3.2.6 设置弹出窗口主题	103
3.2.7 跨页使用弹出窗口	105
习题 3	107
上机操作 3	108
第 4 章 工具栏与导航栏	109
4.1 工具栏	109
4.1.1 页眉工具栏	109
4.1.2 页脚工具栏	112
4.1.3 动态工具栏	114
4.1.4 固定工具栏	115
4.1.5 全屏工具栏	116
4.1.6 外部工具栏	118
4.2 导航栏	121
4.2.1 创建导航栏	121
4.2.2 设置导航栏主题	124
4.2.3 设置导航栏图标	126
4.2.4 创建外部导航栏	129
习题 4	132
上机操作 4	133
第 5 章 网格、表格和列表视图	134
5.1 网格布局	134
5.1.1 创建基本网格	134
5.1.2 创建响应式网格	138
5.2 响应式表格	141
5.2.1 创建回流表格	141
5.2.2 创建列切换表格	144
5.3 列表视图	148
5.3.1 创建只读列表	148
5.3.2 创建链接列表	150
5.3.3 创建插页列表	152
5.3.4 筛选列表内容	154
5.3.5 创建分组列表	156
5.3.6 设置列表图标	158
5.3.7 设置列表文本格式	160
5.3.8 设置列表拆分按钮	163
5.3.9 设置列表主题	165
5.3.10 动态创建列表视图	167
习题 5	170
上机操作 5	171

第 6 章 面板、可折叠块和筛选器	172
6.1 面板	172
6.1.1 创建内部面板	172
6.1.2 创建外部面板	174
6.1.3 设置面板显示方式	177
6.1.4 设置面板关闭方式	179
6.2 可折叠块	181
6.2.1 创建可折叠块	181
6.2.2 设置可折叠块外观	182
6.2.3 设置可折叠块图标	183
6.2.4 创建可折叠表单	185
6.2.5 创建可折叠块系列	186
6.2.6 创建可折叠集合	188
6.3 筛选器	191
6.3.1 创建搜索字段	191
6.3.2 筛选可折叠集合	193
6.3.3 筛选表格	196
6.3.4 筛选控件组	197
6.3.5 筛选 div 元素	200
习题 6	201
上机操作 6	202
第 7 章 jQuery Mobile 表单	204
7.1 表单基础	204
7.1.1 创建表单	204
7.1.2 提交表单	205
7.2 文本输入框与文本区域	207
7.2.1 创建文本输入框	207
7.2.2 设置文本输入类型	209
7.2.3 创建文本区域	212
7.3 单选按钮与复选框	214
7.3.1 创建单选按钮组	214
7.3.2 创建复选框	217
7.4 选择菜单	221
7.4.1 创建本机选择菜单	221
7.4.2 创建选择菜单控件组	225
7.4.3 创建自定义菜单	228
7.4.4 创建多项选择菜单	231
7.5 滑块、范围滑块和翻转开关	234
7.5.1 创建滑块	234

7.5.2 创建范围滑块	237	8.5.1 updatelayout 事件	278
7.5.3 创建翻转开关	241	8.5.2 hashchange 事件	278
习题 7	245	习题 8	278
上机操作 7	246	上机操作 8	279
第 8 章 jQuery Mobile 事件	248	第 9 章 综合设计实例	281
8.1 页面事件	248	9.1 系统功能设计	281
8.1.1 初始化事件	248	9.1.1 系统功能分析	281
8.1.2 页面加载事件	251	9.1.2 数据库设计与实现	282
8.1.3 页面切换事件	255	9.1.3 系统功能模块划分	285
8.1.4 页面改变事件	260	9.2 系统功能实现	286
8.2 触摸事件	263	9.2.1 创建移动 App 项目	286
8.2.1 敲击事件	263	9.2.2 系统初始化	288
8.2.2 滑动事件	265	9.2.3 系统主页实现	289
8.3 用户操作事件	268	9.2.4 信息添加功能实现	292
8.3.1 方向改变事件	268	9.2.5 详情查看功能实现	295
8.3.2 滚屏事件	269	9.2.6 信息修改功能实现	298
8.4 虚拟鼠标事件	271	9.2.7 批量删除功能实现	301
8.4.1 vclick 和 vmousecancel 事件	271	9.3 本地安装包制作	305
8.4.2 vmousedown 和 vmouseup 事件	273	9.3.1 App 参数配置	305
8.4.3 vmouseover 和 vmouseout 事件	274	9.3.2 发行原生安装包	309
8.4.4 vmousemove 事件	276	习题 9	311
8.5 其他事件	278	上机操作 9	312

第1章

移动网站开发基础

移动版网站与传统 PC 版网站的区别主要表现在前端显示上。为了保证移动网站在智能手机、平板电脑等移动设备上能够获得良好的用户体验，通常需要针对这些移动设备专门编写一套 HTML、CSS 以及相关 JavaScript 脚本代码。本章将介绍移动网站开发所需要的一些基础知识和基本技能，主要内容包括创建 HTML5 网页、使用 jQuery 简化编程以及使用 jQuery Mobile 支持移动开发等。

1.1

创建 HTML5 网页

HTML5 是超文本标记语言 HTML 的最新版本，可以用来构建跨设备运行的移动网站。使用 HTML5 创建网页通常需要完成以下三个方面的任务：使用 HTML 语言创建网页的框架结构并在网页中添加各种各样的内容，使用 CSS 样式对网页元素的样式和外观进行设置，并通过编写 JavaScript 脚本实现网页的实用功能。

1.1.1 用 HTML 语言创建内容

无论是移动网站还是传统网站，都是由存储在服务器上的一些 HTML 文档和相关资源组成的。HTML 文档通常称为 Web 页或网页。在网页中可以添加各种各样的内容，如段落、列表、表格、表单、图片、视频以及动画等，这些内容都是使用 HTML 语言来创建的。

1. HTML 元素

网页的内容是由各种各样的 HTML 元素构成的，这些 HTML 元素可以使用标签来定义。多数 HTML 标签的语法格式如下：

```
<标签 属性="值" 属性="值"...>内容</标签>
```

其中，标签是用一对尖括号“<”（小于号）和“>”（大于号）括起来的单词或单词缩写，如<html>...</html>、<head>...</head>、<div>...</div>、<table>...</table>等。<标签>表示开始标签，</标签>表示结束标签。在开始标签与结束标签之间可以放置文本或其他 HTML 元素，这就是 HTML 元素的内容。开始标签、结束标签连同两者之间的内容构成了 HTML 元素。

例如, 要在网页中显示一个一级标题, 可用 `h1` 元素来实现, 代码如下。

```
<h1>jQuery Mobile 移动网站开发</h1>
```

要在网页中添加一个段落, 则要用 `p` 元素来实现, 代码如下:

```
<p>欢迎您加入移动开发! </p>
```

元素的开始标签与结束标签之间一不一定要有内容。没有内容的元素称为**空元素**。对于空元素, 可以只用一个标签表示, 即把开始标签与结束标签合二为一, 并将斜线符号 (`/`) 放到开始标签的末尾。例如, `code` 元素用于表示计算机代码文本, 如果在 `code` 元素的开始标签与结束标签之间没有内容, 则可以写成 `<code>`, 称为**自闭合标签**, 它等价于 `<code></code>`。

有一些 HTML 元素只能使用一个标签表示, 在其中放置任何内容都不符合 HTML 规范。这种没有内容的元素也称为**虚元素**。例如, `hr` 就是虚元素, 它表示主题内容的变化, 在网页中显示一条水平分隔线。虚元素也可以用空元素结构表示, 例如 `<hr>` 标签可以写成 `<hr />`。

2. HTML 属性

使用 HTML 标签定义一个元素时, 通过在元素中添加属性可以设置附加信息。元素的属性只能用在开始标签或单个标签上, 而不能用于结束标签。属性通常以名称/值对的形式出现, 例如 `name="username"`。属性值应该放在引号内。既可以使用双引号, 也可以使用单引号。如果属性值本身就含有双引号, 则必须使用单引号, 例如 `value='Click the "OK" button'`。多个属性之间用空格分隔, 并且不分先后顺序。

例如, 用 `a` 元素在网页中创建一个超链接, 并通过设置 `href` 属性指定该链接指向的目标网址, 代码如下:

```
<a href="http://www.baidu.com">百度一下</a>
```

元素的属性分为全局属性 (标准属性) 和局部属性 (专有属性)。全局属性可用于所有 HTML 元素, 局部属性则为个别元素提供其特有的配置信息。

下面对一些常用的全局属性加以说明。

- **id**: 用来给元素分配一个唯一的标识符。使用该标识符可以将 CSS 样式应用到元素上, 或者在 JavaScript 程序中选择特定的元素。
- **class**: 用来对元素进行归类。使用该属性可以对文档中某一类元素应用一个或多个 CSS 样式, 或者在 JavaScript 程序中选择某一类元素。
- **style**: 用来直接在元素上定义 CSS 样式, 由此定义的样式称为元素内嵌样式。
- **title**: 规定元素的额外信息, 此信息可在工具提示中显示。

有一些 HTML 属性属于布尔属性, 对于这种属性不需要设定一个值, 只需要将属性名添加到元素的开始标签中即可。例如:

```
<button type="submit" disabled>提交</button>
```

在这个例中, 使用 `button` 元素创建了一个按钮, `type` 属性指定按钮的类型, `submit` 表示提交按钮。`disabled` 属性是一个布尔属性, 设置时添加属性名 `disabled` 即可。设置 `disabled` 属性将禁用按钮, 从而阻止用户提供表单数据。

对于布尔属性, 也可以指定一个空字符串 (`""`)、单词 `"true"` 或属性名称作为其值。

在 HTML5 中, 还可以对元素应用自定义属性, 这类属性的名称必须以 `"data-"` 作为前缀。自定义属性 `data-*` 是 HTML5 的新增功能之一。

例如, 要使用 `a` 元素来创建按钮, 则需要设置自定义属性 `data-role`, 代码如下:

```
<a href="#next" data-role="button">转到下一页</a>
```

之所以在这类属性名称之前添加前缀 `data-`, 是为了避免与 HTML 的未来版本中可能增加的属性

名称发生冲突。自定义属性与 CSS 和 JavaScript 结合起来很有用。在使用 jQuery Mobile 框架制作移动网站时，这一类自定义属性得到了广泛应用。

3. HTML 注释

为了增加代码的可读性，可以使用注释标签在 HTML 文档中添加注释，语法格式如下：

```
<!-- 在此输入注释文字 -->
```

注释文字内容会被浏览器忽略，在浏览器中查看网页时是看不到这些内容的。使用注释可以对源代码进行解释，这样做有助于在后期对代码进行编辑和维护。

4. 网页基本结构

HTML 文档具有特定的结构，通常包含一些关键性元素。HTML5 网页的基本结构如下：

```
1: <!doctype html>
2: <html>
3: <head>
4: <meta charset="utf-8">
5: <title>网页标题</title>
6: </head>
7:
8: <body>
9:   在此处添加网页内容...
10: </body>
11: </html>
```

源代码分析

第 1 行：添加了文档类型声明<!doctype>。这是 HTML 文档中的第一个成分。文档类型声明并不是 HTML 标签而是一条指令，它告诉浏览器编写网页所使用的 HTML 规范是什么版本。<!doctype html>指令告诉浏览器编写网页所使用的 HTML 规范是 HTML5。如果希望在 HTML 网页中使用 HTML5 文档类型，就必须在网页的第一行添加这条指令，只有这样浏览器才能了解所预期的文档类型。

第 2 行：添加了 html 元素的开始标签<html>，相应的结束标签是位于第 11 行的</html>。开始标签<html>、结束标签</html>连同两者之间的所有内容构成了 html 元素，该元素是网页的根元素，用于定义整个 HTML 文档，它告诉浏览器这是一个 HTML 文档。HTML 文档是由嵌套的 HTML 元素构成的，根元素 html 的内容是 head 元素和 body 元素。

第 3 行：添加了 head 元素的开始标签<head>，该元素的结束标签</head>位于第 6 行。head 元素用于向浏览器提供有关 HTML 文档的信息，通常称为头部信息。浏览器不会向用户显示这些头部信息。每个 HTML 文档都应该有一个 head 元素。在网页基本结构中，head 元素的内容是另外两个元素，即 meta 元素和 title 元素。除此之外，在文档头部还经常使用 link 元素来引用外部 CSS 样式文件，使用 script 元素来添加 JavaScript 脚本或引用外部 JavaScript 文件，使用 style 元素创建文档内嵌 CSS 样式。

第 4 行：添加了一个 meta 元素，它只有一个标签，采用虚元素形式。meta 元素描述网页的一些元数据，它通过 charset 属性指定文档的默认编码。utf-8 用 1~4 个字节编码 Unicode 字符，可以用于在网页上显示简体中文、繁体中文及其他语言。meta 标签除了用来设置文档编码，还有许多其他用途。

第 5 行：添加了 title 元素，它由开始标签<title>、结束标签</title>和标题文字组成，用于指定网页的标题。当在浏览器中加载网页时，网页标题就显示在当前文档所在选项卡的标题栏中；将当前网页添加到收藏夹时网页标题将显示在收藏夹中；在搜索引擎结果页面中网页标题作为搜索结果的标题出现。

第 8 行：添加了 body 元素的开始标签<body>，该元素的结束标签是位于第 10 行的</body>。body 元素用于定义 HTML 文档的主体。在 HTML5 中，删除了 body 元素的所有局部属性，这些属性即使在 HTML 4.01 中也是不赞成使用的。

第 9 行：由此开始可以使用各种 HTML 标签来添加 body 元素包含的内容，也就是 HTML 文档包含的内容，如文本、超链接、图形、图像、列表、表格、音频、视频以及动画等。

例 1.1 在网页中创建登录表单，并将各个表单控件放置在表格中。源文件为 01-01.html，源代码如下。

```

1: <!doctype html>
2: <html>
3: <head>
4: <meta charset="utf-8">
5: <title>网站登录</title>
6: </head>
7:
8: <body>
9: <h1>网站登录</h1>
10: <form method="post" action="">
11:   <table>
12:     <tr>
13:       <td><label for="username">用户名: </label></td>
14:       <td><input type="text" id="username" required placeholder="输入用户名"></td>
15:     </tr>
16:     <tr>
17:       <td><label for="password">密码: </label></td>
18:       <td><input type="password" id="password" required placeholder="输入密码"></td>
19:     </tr>
20:     <tr>
21:       <td>&nbsp;</td>
22:       <td><input type="submit" value="登录">&nbsp;</td>
23:       <td><input type="reset" value="重置"></td>
24:     </tr>
25:   </table>
26: </form>
27: </body>
28: </html>

```

源代码分析

第 9 行：添加了 h1 元素，用于定义一个 HTML 标题。在网页中可用<h1>~<h6>标签来定义 HTML 标题。<h1>用于定义重要等级最高的标题。<h6>则用于定义重要等级最低的标题。

第 10 行：添加了 form 元素的开始标签<form>，该元素的结束标签</form>位于第 27 行。form 元素用于创建供用户输入的 HTML 表单。在开始标签<form>中设置了表单元素的两个属性，method 属性指定用于发送表单数据的 HTTP 方法（例中为 post），action 属性指定当提交表单时向何处发送表单数据（例中为空字符串）。表单只是定义了一个区域，在该区域中可以包含各种表单控件。表单本身提供没有输入数据的手段，要让用户通过表单输入数据，还必须在表单中添加各种表单控件，例如文本框、按钮等。

第 11 行：添加了 table 元素的开始标签<table>，该元素的结束标签</table>位于第 26 行。table 元素用于定义表 HTML 格。每个表格有一些行（用 tr 元素定义），每一行被分割为若干个单元格（用 td 元素定义）。td 用于定义数据单元格，在数据单元格可以包含文本、图片、列表、段落、表单按钮、表格等。例中的表格包含三行两列。

第 13 行、第 17 行：分别在单元格中添加了一个 label 元素，用于为 input 元素定义标注，提示文本框的用途。通过设置 for 属性规定 label 与哪个表单控件绑定。<label>标签的 for 属性应当

与相关元素的 id 属性相同。该标签为鼠标用户改进了可用性。当用户单击该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

第 14 行：在单元格中添加一个 input 元素并将 type 属性设置为 text，以创建单行文本框。通过设置 required 属性规定在提交表单之前必须在该文本框中填写内容，如果不填写内容，则阻止提交表单；通过设置 placeholder 属性指定输入字段预期值的简短提示信息。

第 17 行：在单元格中添加了一个 input 元素并将 type 属性设置为 password，以创建密码输入框，当用户在此框中输入密码字段时，所输入的字符会被遮蔽。

第 22 行、第 23 行：在单元格中添加两个 input 元素，并将其中一个的 type 属性设置为 submit 以创建提交按钮，将另一个的 type 属性设置为 reset 以创建重置按钮，还通过设置 value 属性指定在这些按钮上显示的标题文字。当单击提交按钮时，表单数据将被发送到 form 元素的 action 属性指定的位置进行处理；当单击重置按钮时，各个表单控件将恢复为初始状态。

网页显示结果如图 1.1 所示。



图 1.1 登录页面

1.1.2 用 CSS 设置样式

使用 HTML 标签定义网页内容后，还必须使用 CSS 来设置 HTML 元素的样式，例如网页文本所用的字体、字号和颜色，以及段落的缩进量、对齐方式和行间距等。使用 CSS 样式可以精确地为 HTML 元素设置格式，并且能够将其应用到网站的任何页面中。如果希望对网站进行全局更新，只需要修改 CSS 样式表，就能够使网站中的所有页面自动地更新。

CSS 样式表由一组 CSS 规则组成，每个 CSS 规则由选择器和属性声明两个部分组成，其中选择器用于标识和选择一个或多个 HTML 元素，属性声明用于设置所选元素的样式。CSS 属性声明包含属性名和属性值两个部分，属性名与属性值与冒号分隔，不同属性声明之间用分号分隔。CSS 样式按所在位置不同可分为元素内嵌样式、文档内嵌样式和外部样式。

1. 元素内嵌样式

元素内嵌样式是指通过元素的全局属性 style 来设置 CSS 属性，使用这种样式时不需要使用选择器，它只能应用于所在元素。例如，下面用 style 属性对 div 元素的样式进行了设置：

```
<div style="height: 120px; width: 300px; background-color: grey;"></div>
```

在上述代码中，通过 style 属性设置了 div 元素的高度 (height)、宽度 (width) 以及背景颜色 (background-color)。

2. 文档内嵌样式

文档内嵌样式是指放在文档内部的 CSS 样式，这种样式通常放在网页头部，它只能应用于当前页面。创建文档内嵌样式时，应使用 style 元素定义一个样式表，并在该样式表设置一组 CSS 样式规则，每个规则中的所有属性声明需要用一对花括号括起来。

例 1.2 在网页头部创建 CSS 样式表。源文件为 01-02.html，源代码如下。

```
1: <!doctype html>
2: <html>
3: <head>
4: <meta charset="utf-8">
5: <title>渐变背景、圆角边框与阴影</title>
```

```

6: <style>
7: .demo {
8:   width: 300px;           /* 设置元素的宽度 */
9:   height: 180px;          /* 设置元素的高度 */
10:  margin: 0 auto;          /* 设置元素的外边距 (使其水平居中) */
11:  padding: 0.5em;          /* 设置元素的内边距 */
12:  line-height: 60px;       /* 设置元素的行高 */
13:  font-size: 32px;         /* 设置字号大小 */
14:  font-weight: bold;        /* 设置字体粗细 */
15:  text-align: center;       /* 设置文本对齐方式 */
16:  text-shadow: 6px 6px 12px grey; /* 设置文本阴影效果 */
17:  border: 3px solid orange; /* 设置元素边框的宽度、线型和颜色 */
18:  border-radius: 16px;      /* 设置边框圆角半径 */
19:  box-shadow: 12px 12px 12px gray; /* 设置边框阴影 */
20:  background-image: linear-gradient(white, red); /* 设置元素背景为线性渐变 */
21: }
22: </style>
23: </head>
24:
25: <body>
26: <div class="demo">
27:   <p>jQuery Mobile<br>移动网站开发</p>
29: </div>
30: </body>
31: </html>

```

源代码分析

第 6~第 20 行: 使用 style 元素创建了一个 CSS 样式表。

第 7~第 19 行: 创建了一个 CSS 规则, 以“.demo”作为选择器, 对网页中 class 属性值为“.demo”的元素的多种 CSS 属性进行了设置。每个属性用途通过 CSS 注释标出。



图 1.2 创建文档内嵌样式

第 25~第 28 行: 添加了一个 div 元素, 并将其 class 属性设置为“demo”。该元素应用了 CSS 样式表中的那个 CSS 规则。

网页显示效果如图 1.2 所示。

3. 外部样式表

元素内嵌样式只能应用于所在元素, 文档内嵌样式只能应用于当前页面中的元素。为了将 CSS 样式应用于同一网站的多个页面中, 必须将 CSS 样式存储在单独的文件中, 这就是 CSS 样式文件, 其文件扩展名为.css。在 CSS 样式文件中, 不再需要添加<style>标签, 可以直接定义 CSS 规则。

例如, 可以创建一个 CSS 样式文件并命名为 mystyle.css, 其内容如下。

```

div {
  height: 120px;           /* 设置高度 */
  width: 200px;            /* 设置宽度 */
  border: thin solid green; /* 设置边框的宽度、线型和颜色 */
  background-color: #7ebdeb; /* 设置背景颜色 */
}

```

这个 CSS 样式文件可以应用于同一网站中的所有页面。如果要在某个网页中引用这个 CSS 样式文件, 则应在网页头部添加以下<link>标签:

```
<link rel="stylesheet" href="mystyle.css">
```


其中, rel 属性规定了当前文档与被链接文档之间的关系,只有 "stylesheet" 值得到了所有浏览器的支持,该属性不能省略; href 属性规定被引用文档的 URL 位置,可以是相对路径,也可以是绝对路径。

在实际应用中,也可以使用 link 元素从内容分发网络 CDN (Content Delivery Network) 上加载所需要的 CSS 样式文件。例如:

```
http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css>
```

内容分发网络 CDN 的基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节,使内容传输的更快、更稳定。

1.1.3 用 JavaScript 实现功能

用 HTML 语言创建网页内容后,除了通过 CSS 设置网页的样式和布局,通常还需要借助 JavaScript 脚本对发生在网页中的各种事件的进行处理,以完成许多常见任务,例如在网页中绘制图形、控制媒体播放、存储和查询数据,对表单数据的有效性进行验证,并根据具体情况适时改变网页的内容和样式,以生成更流畅、更美观的动态效果,最终实现所需要的实用功能。

JavaScript 是一种轻量级的编程语言,是一种解释性脚本语言,其代码不需要进行预编译,插入 HTML 页面后,即可由所有的现代浏览器执行。JavaScript 脚本解释器是浏览器的组成部分,无需专门下载和安装。

1. 在网页中直接添加 JavaScript 脚本

在网页中可以使用 script 元素来直接添加 JavaScript 客户端脚本。script 元素既可以放在网页 head 部分,也可以放在网页的 body 部分,还可以同时放在 head 和 body 部分。在网页中不限制添加 script 元素的数量。

添加 script 元素时,可以将 type 属性设置为 "text/javascript",以规定脚本的 MIME 类型。不过,也可以不设置这个属性,因为在现代浏览器中 JavaScript 就是 HTML5 的默认脚本语言。

例 1.3 在网页中添加 JavaScript 脚本。源文件为 01-03.html,源代码如下。

```
1: <!doctype html>
2: <html>
3: <head>
4: <meta charset="utf-8">
5: <title>控制灯泡的开与关</title>
6: <style>
7: .container {
8:     margin: 0 auto;
9:     padding: 20px;
10:    text-align: center;
11: }
12: #status {
13:    display: inline-block;
14:    margin-left: 3px;
15:    padding: 3px 6px 3px 6px;
16:    background-color: grey;
17:    color: white;
18:    font-style: italic;
19: }
20: </style>
21: <script>
22:    window.onload=function() {           //设置 window 对象的 load 事件处理程序
23:        var bulbo, status;
24:        bulbo=document.getElementById("bulbo"); //获取 image 对象
25:        status=document.getElementById("status"); //获取 span 对象
```

```

26:     bulbo.onclick=function() { //设置图像的 click 事件处理程序
27:         if ( bulbo.src.match("off") ) { //若灯泡当前处于关闭状态
28:             bulbo.src="../images/bulbon.gif"; //更换为点亮灯泡的图像
29:             bulbo.title="单击关闭灯泡"; //修改图像的提示文字
30:             status.style.backgroundColor="red"; //改变状态信息的背景颜色为红色
31:             status.style.color="yellow"; //改变状态信息的文字颜色为黄色
32:             status.innerHTML="点亮了"; //改变状态信息的文字内容
33:         } else { //若灯泡当前处于点亮状态
34:             bulbo.src="../images/bulboff.gif"; //更换为关闭灯泡的图像
35:             bulbo.title="单击点亮灯泡"; //修改图像的提示文字
36:             status.style.backgroundColor="grey"; //改变状态信息的背景颜色为灰色
37:             status.style.color="white"; //改变状态信息的文字颜色为白色
38:             status.innerHTML="熄灭了"; //改变状态信息的文字内容
39:         }
40:     }
41: }
42: </script>
43: </head>
44:
45: <body>
46: <figure class="container">
47:     
48:     <figcaption>灯泡<span id="status">熄灭了</span></figcaption>
49: </figure>
50: </body>
51: </html>

```

源代码分析

第 46~第 48 行: 添加了一个 figure 元素, 用于生成一幅灯泡插图, 并通过 figcaption 元素为该插图设置标题。在 figcaption 元素中包含一个 span 元素, 用于指示灯泡的状态。

第 6~第 20 行: 定义了一个 CSS 样式表, 其中包含两条 CSS 规则。第一条规则以“.container”作为选择器, 用于匹配网页中 class 属性值为“container”的 figure 元素, 对该元素的样式属性进行了设置。第二条规则以“#status”为选择器, 用于选择网页中 id 属性值为“status”的 span 元素, 并对其样式属性进行了设置

第 21~第 42 行: 通过添加 script 元素定义了一段 JavaScript 客户端脚本。

第 22 行: 对 window 对象的 onload 事件属性进行了设置, load 事件会在文档加载完成后立即发生。在这里设置 onload 事件属性值指向一个匿名函数, 通过函数执行的代码包含在一对花括号内 (右花括号位于第 41 行), 当文档加载就绪时会执行这些代码。

第 24 行、第 25 行: 通过调用 document 对象的 getElementById()方法获取网页中 image 元素和 span 元素对应的 DOM 对象。

第 26 行: 对 image 元素对象的 onclick 事件属性进行了设置, 当在网页中单击灯泡图像时会发生 click 事件。在这里, 设置 onclick 事件属性指向一个匿名函数, 在该函数中将根据当前所加载的图像不同而执行不同的操作, 包括修改图像的来源文件和提示信息、更改 span 元素的背景颜色、前景颜色和文字内容等。

第 27 行: 通过调用字符串对象的 match()方法在字符串内检索指定的值。如果找到匹配的文本, 该方法将返回一个数组, 其中存放了与所找到的匹配文本有关的信息, 否则它将返回一个 null 值。

在浏览器打开该网页时, 显示的是灯泡关闭图像, 单击该图像即变成灯泡点亮图像, 同时状态信息也发生变化, 结果如图 1.3 和图 1.4 所示。

2. 引用外部脚本文件

也可以把 JavaScript 脚本保存到外部脚本文件中, 这种文件通常包含被多个网页使用的代码, 例如各种公用的 JavaScript 函数定义等。外部 JavaScript 文件的文件扩展名是.js, 在该文件不能包

含<script>标签。如果需要在网页中使用外部文件，可以在<script>标签设置 src 属性并将该属性值指定为该文件的 URL 路径，此时不要在开始标签<script>与结束标签</script>之间添加任何脚本内容。

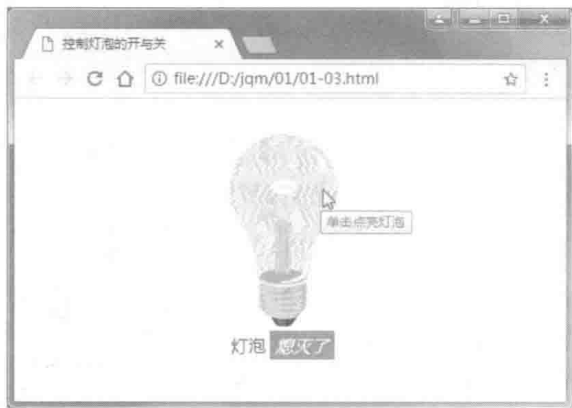


图 1.3 灯泡处于关闭状态



图 1.4 灯泡处于点亮状态

例如，可以创建一个脚本文件 myscript.js 并在其中编写一条语句，源代码如下。

```
document.writeln("Hello, World!");
```

如果希望在网页中加载这个脚本文件，只需要将 script 元素的 src 属性设置为该文件的 URL 路径即可。源代码如下：

```
1: <!doctype html>
2: <html>
3: <head>
4: <meta charset="utf-8">
5: <title>外用外部 JavaScript 文件</title>
6: </head>
7: <body>
8: <script src="js/myscript.js"></script>
9: </body>
10: </html>
```

也可以从内容分发网络 CDN 上加载 JavaScript 脚本文件。例如：

```
<script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
```

加载外部脚本文件时，还可以对 script 标签设置以下两个属性。

（1）async 属性：规定异步执行脚本。async 属性是一个布尔属性。如果设置 async 属性，则脚本相对于页面的其余部分异步地执行，即当页面继续进行解析时脚本将被执行。

（2）defer 属性：规定当页面已完成解析后执行脚本。defer 属性也是一个布尔属性，它规定当页面完成加载后才会执行脚本。

以上两个属性仅适用于外部脚本，即只有在设置 src 属性时才能使用。如果既不使用 async 属性也不使用 defer 属性，则在浏览器继续解析页面之前立即读取并执行脚本。

1.2 使用 jQuery 简化编程

jQuery 是一款轻量级的 JavaScript 框架，其核心理念是“少写、多做”。它提供了简单易用的 API，使得诸如 HTML 文档遍历、DOM 操作、事件处理、动画效果和 Ajax 交互等功能可以在众多