

高等院校信息技术规划教材

Web系统开发与实践

马瑞新 原旭 编著

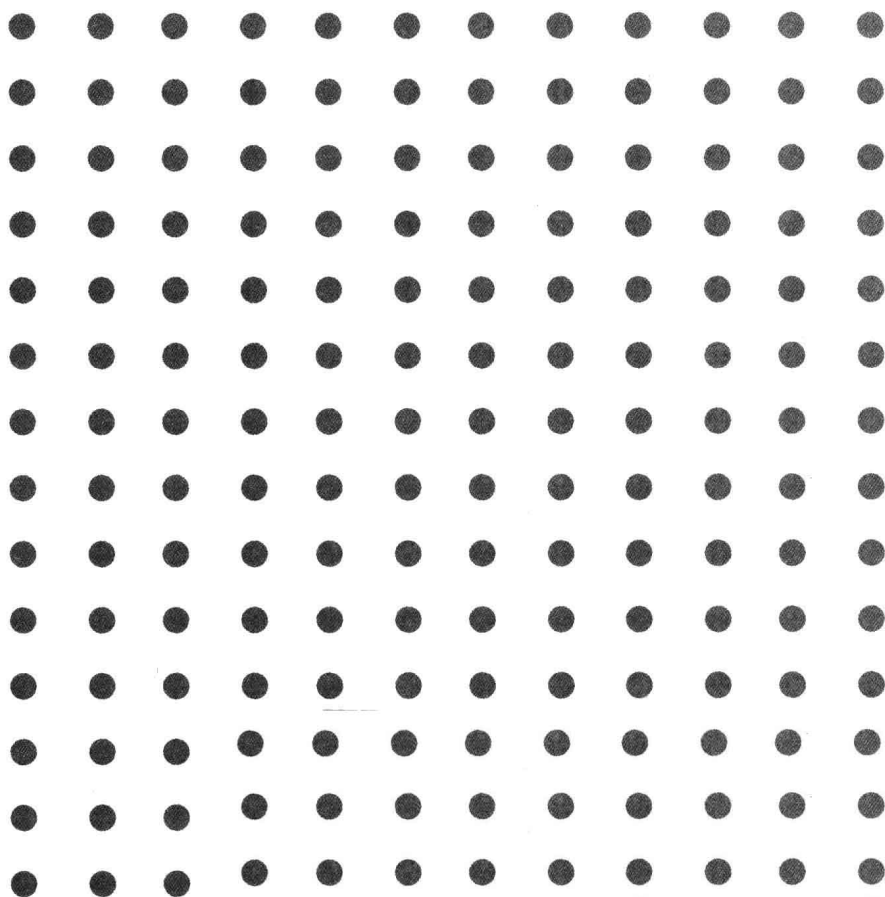


清华大学出版社

息 技 术 规 划 教 材

Web系统开发与实践

马瑞新 原旭 编著



清华大学出版社
北京

内 容 简 介

本书以 Web 开发为背景,力图系统、全面地介绍 Web 应用开发所涉及的内容,由浅入深地展开,在内容和结构安排上力求做到系统性和连贯性。本书分为 3 部分,共 5 章。第一部分包括第 1~3 章,主要介绍 Web 应用与开发的基本概念与特征,以 HTML、CSS、JavaScript 技术为主,介绍静态页面制作,同时辅以案例进行说明。第二部分包括第 4 章,是以 ASP.NET 技术为主,介绍一个网上书城的开发。第三部分包括第 5 章,是以 Java 为主的一个在线通讯录开发。

本书可作为高等院校信息管理与信息系统、电子商务、工商管理及管理理学等专业的教材,也可供相关专业技术人员、工程技术人员和其他人员参考。

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

Web 系统开发与实践/马瑞新,原旭编著. —北京:清华大学出版社,2012.12

高等院校信息技术规划教材

ISBN 978-7-302-30476-0

I. ①W… II. ①马… ②原… III. ①互联网络—程序设计—高等学校—教材 IV. ①TP393

中国版本图书馆 CIP 数据核字(2012)第 249881 号

责任编辑:白立军

封面设计:傅瑞学

责任校对:白 蕾

责任印制:何 芊

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>, 010-62795954

印刷者:三河市君旺印装厂

装订者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.5

字 数:429 千字

版 次:2012 年 12 月第 1 版

印 次:2012 年 12 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

前言

foreword

Web 开发技术是 IT 应用型人才应该具备的关键技能之一,本书以 Web 开发为背景,力图系统、全面地介绍 Web 应用开发所涉及的内容,由浅入深地展开,在内容和结构安排上力求做到系统性和连贯性。本书共 5 章,可分为 3 部分。第一部分包括第 1~3 章,主要介绍 Web 应用与开发的基本概念与特征,以 HTML、CSS、JavaScript 技术为主,介绍静态页面制作,同时辅以案例进行说明。第二部分包括第 4 章,以 ASP.NET 技术为主,介绍一个网上书城的开发。第三部分包括第 5 章,以 Java 技术为主的一个在线通讯录的开发。

本书的特点是实用性强。文中所选取的实例,均是在 Web 开发中所需要解决的实际问题,这些实例不但能够说明问题,而且具有很强的实用性,读者只需将某些代码更改为自己的网页中相应内容,便可方便地让自己的网页有同样精彩的动态效果。

本书在介绍实例的同时,也向读者介绍了程序的设计思想,使读者能够举一反三,运用所学知识设计更多的实用程序。在设计实例的过程中,本书不仅考虑了网页的功能,而且对网页的美观和布局也进行了考虑。

学习计算机程序设计的最好方法是实践,因此建议读者上机编写、运行和调试本书所给的例程。本书中的所有程序都在 Web 环境中调试通过,可以帮助读者通过上机实践来检查自己对书中内容的理解和掌握程度。

学习是一个慢慢体会的过程,同时也是一个逐渐发现的过程,一个人能够学习、积累、总结和思考,这将是幸福的。

本书由马瑞新、原旭编著,参与本书编写的还有刘畅、王永山、崔亚杰。同时也感谢大连理工大学软件学院的全体同学,是他们热情地支持和鼓励让我有信心完成此书的创作,每当我快要放弃时,

总是他们给予我力量,我特将此书作为他们的礼物,希望我们一起在软件开发上更上一层楼。

由于作者水平有限,加之时间仓促,书中的错误之处在所难免,敬请广大专家、读者不吝指正,予以赐教。

马瑞新

2012 年 9 月

目录

Contents

第 1 章 Web 技术基础	1
1.1 HTML 标签	1
1.2 HTML 表单	6
1.3 HTML 框架	17
1.4 CSS 样式表	22
1.5 习题训练	39
第 2 章 JavaScript	42
2.1 基本语法	42
2.2 DOM 编程	52
2.3 表单验证	70
2.4 习题训练	82
第 3 章 招聘网站设计	86
3.1 首页设计	86
3.2 注册页面	97
3.3 登录页面	102
3.4 简历管理页面	106
3.5 职位搜索页面	118
3.6 招聘公司页面	129
3.7 习题训练	142
第 4 章 基于 ASP.NET 的网上书城	143
4.1 系统概述	143
4.1.1 前台购书系统	143
4.1.2 后台管理	144



4.2	数据访问类	151
4.3	用户注册模块	164
4.4	用户登录模块	173
4.5	网站访问计数器模块	183
4.6	图书信息查询模块	185
4.7	图书信息浏览模块	191
4.8	购物车模块	200
4.9	购物结算与订单查询模块	206
4.10	后台管理模块	211
4.11	习题训练	224
第 5 章 基于 Java 的在线通讯录		226
5.1	系统数据库	226
5.2	JavaBean	228
5.3	JDBC	236
5.4	数据库 Bean	244
5.5	Servlet 技术	249
5.6	页面设计	262
5.7	MVC 设计模式	272
5.8	习题训练	274

Web 技术基础

1.1 HTML 标签

在当今社会,网络已成为人们生活的一部分,网络提供的服务主要以网页形式展现出来,HTML 是创建网页的基础语言,如果不了解 HTML,就谈不上网页设计。HTML 的英文全称是 HyperText Marked Language,中文叫做“超文本标记语言”。对 HTML 的编辑使用普通的文本编辑器即可(记事本、Editplus 等)。

HTML 是 Web 用于创建和识别文档的标准语言,这些标记都是通过使用标签来完成的,标签可指定网页在浏览器中的显示方式。本章介绍 HTML 文件的基本结构和相关标签等。

完整的 HTML 文件至少包括<HTML>标签、<HEAD>标签、<TITLE>标签和<BODY>标签,并且这些标签都是成对出现的,开头标签为<>,结束标签为</>,在这两个标签之间添加内容。通过这些标签中的相关属性可以设置页面的背景色、背景图片等。

HTML 文档主要由以下三部分组成。

(1) HTML 部分: HTML 部分是以<HTML>标签开始,以</HTML>标签结束。

```
<HTML>
:
</HTML>
```

<HTML>标签告诉浏览器这两个标签之间的内容是 HTML 文档。

(2) 头部: 头部以<HEAD>标签开始,以</HEAD>标签结束。这部分包含显示在网页标题栏中的标题和其他在网页中不能显示的信息。标题包含在<TITLE>和</TITLE>标签之间。

```
<HEAD>
  <TITLE>...</TITLE>
</HEAD>
```

(3) 主体部分: 主体部分包含在网页中显示的文本、图像和链接。主体部分以<BODY>标签开始,以</BODY>标签结束。



```
<BODY>
:
</BODY>
```

值得注意的是标签不区分大小写,可以使用<html>来代替<HTML>。

HTML 文档的结构如下:

```
<HTML>
<HEAD>
<TITLE>我的第一个网页 </TITLE>
</HEAD>
<BODY >
    Hello World!
</BODY>
</HTML>
```

创建网页也非常简单,只需要按照下面的步骤完成即可。

使用记事本创建网页的步骤:

- (1) 打开记事本;
- (2) 输入 HTML 代码;
- (3) 保存为 *.html 或 *.htm 文件,注意格式问题;
- (4) 打开网页预览效果。

<META>标签出现在网页的标题部分,这是一个特殊的 HTML 标签,提供有关网页的信息。有时访问网页时发现文字是乱码,其实就是没有正确设置<META>标签中 charset 属性。如要将 gb2312 指定为默认字符集类型,则使用以下<META>标签:

```
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>网页标题</TITLE>
</HEAD>
```

其中 charset 用于设置网页的编码语系,简体中文使用 charset=gb2312,繁体中文使用 charset=big5,纯英文网页建议使用 iso-8859-1,或者不设编码也可,网页会根据系统所在国家或页面主体所对应的编码体系来正确设置 charset。

在默认情况下,使用 Web 浏览器浏览网页时,其背景色是白色,如果我们想把背景色换成好看的颜色或图片怎么办? 其实很简单,使用 bgcolor 属性可以改变网页的背景色,使用 background 属性可以把网页背景设为图片。

```
<BODY bgcolor="#FFCCFF" background="back_image.GIF" >
    Hello World!
</BODY>
```

文本是网页不可缺少的元素之一,是网页发布信息所采用的主要形式,为了让网页中的文字看上去编排有序,整齐美观、错落有致,我们就要设置文本的大小、颜色、字体类型及换行等。

文本相关的标签大都放在 BODY 标记内。常用的标签有字体标签、图片标签、超链接标签、列表标签、表格、表单、块级元素(div 和 span)。

1. 字体标签

基本语法: ***

```
<HTML>
  <BODY>
    <FONT face="隶书" size="5" color="bule">HTML 字体标签实例</FONT>
  </BODY>
</HTML>
```

2. 图片标签

```
<HTML>
  <BODY>
    <IMG src=" demopic.jpg" width="200" height="100" border="5">
  </BODY>
</HTML>
```

其中 src 属性表示图片的路径和文件名, width 和 height 为宽和高, border 为边线宽度。

3. 超链接标签

基本语法 **, <A>是 Anchor(锚)的缩写, 从一个页面链接到另一个页面。

```
<HTML>
  <BODY>
    <A href=" http://www.sina.com.cn" >链接到新浪网站</A>
  </BODY>
</HTML>
```

4. 列表标签

有两种方式, 有序列表和无序列表。

```
<HTML>
  <BODY>
    有序列表
    <OL>
      <LI>HTML 编程基础</LI>
      <LI>CSS 基础</LI>
      <LI>javascript 编程基础</LI>
    </OL>
    无序列表
```



```
<UL>
  <LI>HTML 编程基础</LI>
  <LI>CSS 基础</LI>
  <LI>javascript 编程基础</LI>
</UL>
</BODY>
</HTML>
```

5. 表格标签

表格基本标签<TABLE><TR><TD>

```
<HTML>
<BODY>
  <TABLE border="1">
    <TR><TD>11</TD><TD>12</TD></TR>
    <TR><TD>21</TD><TD>22</TD></TR>
    <TR><TD>31</TD><TD>32</TD></TR>
  </TABLE>
</BODY>
</HTML>
```

跨行跨列是利用 rowspan 和 colspan 设置跨行跨列。

```
<HTML>
<BODY>
  <TABLE border="1">
    <TR><TD rowspan="2">跨两行</TD><TD colspan="2">跨两列</TD></TR>
    <TR><TD>1000</TD><TD>1000</TD></TR>
    <TR><TD>1000</TD><TD>3000</TD><TD>4000</TD></TR>
  </TABLE>
</BODY>
</HTML>
```

cellpadding 和 cellspacing 属性的使用,前者是单元格的边距,即字与单元格边框的距离。后者指单元格间距。

```
<HTML>
<BODY>
  <TABLE border="1" cellpadding="10" cellspacing="0">
    <TR><TD>第一行第一列</TD><TD>第一行第二列</TD></TR>
    <TR><TD>1 第二行第一列</TD><TD>第二行第二列</TD></TR>
  </TABLE>
  <BR>
  <TABLE border="1" cellpadding="0" cellspacing="10">
    <TR><TD>第一行第一列</TD><TD>第一行第二列</TD></TR>
```

```

    <TR><TD>1 第二行第一列</TD><TD>第二行第二列</TD></TR>
  </TABLE>
</BODY>
</HTML>

```

表格样式:

可以用过 bordercolor 设置表格边框颜色, bgcolor 设置背景颜色, align 设置表格的对齐方式。

```

<HTML>
<BODY>
  <TABLE border="4" bordercolor="orange">
    <TR bgcolor="orange" align="middle">
      <TD><font color="white"><B>序号</B></FONT></TD>
      <TD><font color="white"><B>姓名</B></FONT></TD>
      <TD><font color="white"><B>高数</B></FONT></TD>
      <TD><font color="white"><B>语文</B></FONT></TD>
      <TD><font color="white"><B>软件工程</B></FONT></TD>
    </TR>
    <TR align="middle">
      <TD>1</TD><TD>杨过</TD><TD>150</TD><TD>150</TD><TD>150</TD></TR>
    <TR align="middle"><TD>2</TD><TD>小龙女
      </TD><TD>150</TD><TD>150</TD><TD>150</TD></TR>
  </TABLE>
</BODY>
</HTML>

```

6. 表单标签

通用格式: <form method="Post" action="do_submit.jsp">***</form>

Post 方式在浏览器的地址栏中不显示提交的信息,这种方式传输数据没有数据量的限制;Get 方法将信息传递到浏览器的地址栏上,最大传输信息量是 2KB。不指明,默认是 Get 方式。

```

<HTML>
<BODY>
  <FORM method="Post" action="do_submit.jsp">
    用户名: <INPUT type="text" name="user_id"><BR>
    密码: <INPUT type="text" name="user_pwd"><BR><BR>
    <INPUT type="Submit" value="提交" name="btn_submit">
    <INPUT type="Reset" value="重写" name="btn_reset">
  </FORM>
</BODY>
</HTML>

```

7. DIV 和 SPAN 标签

```
<HTML>
  <BODY>
    <DIV id="div1" style="background:yellow">这是一个 DIV</DIV>
    <SPAN id="span1" style="background:yellow">这是一个 SPAN </SPAN >
  </BODY>
</HTML>
```

8. 其他常用标签

包括文本框、文本区域、密码框、多选框、单选框和下拉框等。除了文本区域和下拉框,其他的只需要修改 type 属性即可。

```
<HTML>
  <BODY>
    <INPUT type="text" name="text" value="文本框">
    <INPUT type="password" name="text" value=" 密码框">
    <INPUT type="checkbox" name="text" value=" 多选框">
    <INPUT type="radio" name="text" value=" 单选框">
    <TEXTAREA NAME="textareal" ROWS="5" COLS="10"></TEXTAREA>
    <SELECT NAME=""></SELECT>
  </BODY>
</HTML>
```

1.2 HTML 表单

热衷于上网的用户经常会在网上看到一些注册页面、购买商品搜集信息页面、网上调查问卷页面、搜索工具页面等,这些页面都包含有表单,如图 1.1 所示。

网页中的表单用途很广,而且还在不断发展。典型的表单应用如下所示。

- (1) 注册用户。
- (2) 收集信息。
- (3) 反馈信息。
- (4) 为网站提供搜索工具。

创建表单后,就可以在表单中放置控件以接受用户的输入。这些控件通常放在<FORM></FORM>标签之间一起使用,也可以在表单之外用来创建用户界面。在网上访问时,会经常看到一些常用的控件,例如,让用户输入姓名的单行文本框,让用户输入密码的密码框、让用户选择性别单选按钮以及让用户提交信息的提交按钮等。

不同表单控件有不同的用途。如果要求用户输入仅仅是一些文字信息,如“姓名”、“备注”、“留言”等,一般使用单选按钮、复选框和下拉列表框,如图 1.2 所示。“性别”、

图 1.1 包含表单的注册页面

“爱好”、“出生日期”中的月份选择等常采用这些控件, 如果要把填写好的表单信息提交给服务器, 一般使用“提交”按钮, 其他一些不太常用的表单控件就不一一列举了。

图 1.2 常见表单控件

<FORM> 标签用于在网页中创建表单区域, 属于一个容器标签, 表示其他表单标签需要在它的包围中才有效, <input> 便是其中一个。用于设定各种输入资料的方法。<FORM> 元素的属性有两种: action 和 method。



action: 此属性指示服务器上处理表单输出的程序。一般来说,当用户单击表单上的“提交”按钮后,信息发送到 Web 服务器上,由 action 属性所指定的程序处理。语法为: action="URL"。

method: 此属性告诉浏览器如何将数据发送给服务器,它指定向服务器发送数据的方法(用 post 还是 get)。如果值为 get,浏览器将创建一个请求,该请求包含页面 URL、一个问号和表单的值。浏览器会将该请求返回给 URL 中指定的脚本,以进行处理。如果将值指定为 post,表单上的数据会作为一个数据块发送到脚本,而不使用请求字符串,语法为: method=(get|post)。

示例 1.1 演示 post 和 get 方法的区别。

示例 1.1

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>get 和 post 区别演示</TITLE>
</HEAD>

<BODY>
<FORM name="form1" method="post" action="">
<P>名字:
    <INPUT name="name" type="text" class="input" id="fname">
</P>
<P>密码:
    <INPUT name="pass" type="password" class="input" id="pass">
</P>
<P>
    <INPUT type="submit" name="Button" value="提交">
    <INPUT type="reset" name="Reset" value="重填">
</P>
</FORM>
</BODY>
```

在示例 1.1 中,若把 method="post"改为 method="get"就变成了使用 get 方法将表单提交给服务器。这两种方法有什么区别呢?

先看看使用 post 和 get 方法提交表单后浏览器地址栏的变化,如图 1.3 所示,这是采用 post 方法提交的表单,浏览器地址栏前后没有变化。

如图 1.4 所示,这是采用 get 方法提交的表单,浏览器地址栏前后没有变化。

在浏览器地址栏中能看到“get_show.html? name=mike&pass=123”,这其实就是刚才输入的用户名和密码。由此可见,使用 post 方法提交的表单信息更安全,相反,使用 get 方法提交表单信息极不安全,建议大家在使用表单时尽可能使用 post 方法来传送数据。

网页中的表单由许多不同的表单元素组成,这些表单元素各自都有很多属性,下面

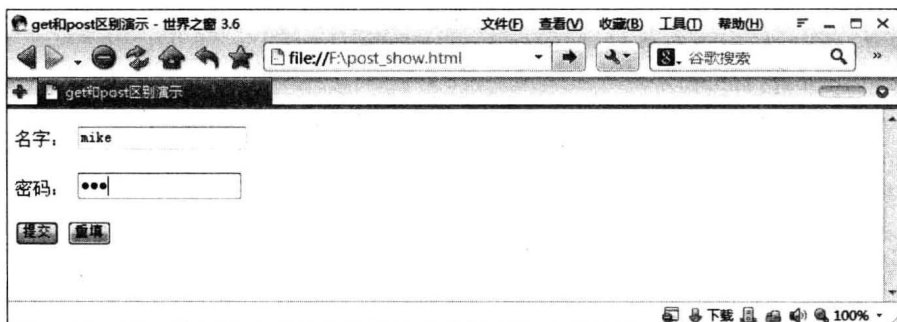


图 1.3 采用 post 提交表单



图 1.4 采用 get 提交表单

对这些表单元素中一些常用的属性做一个统一介绍。

下面给出表单元素的统一格式：

```
<FORM name="form3" method="post" action="">
  <INPUT type="checkbox" name="gen" value="男"
    size="21" maxlength=4 checked="checked">
  :
</FORM>
```

1. 文本框

在表单中最常用的表单输入元素就是文本框(text)，它提供给用户输入单行文本信息，例如用户名的输入框。如要在表单里创建一个文本框，将 type 属性改为 text 就可以了，如示例 1.2 所示。

示例 1.2

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>文本框练习</TITLE>
</HEAD>
```



```
<BODY>  
<FORM name="form1" method="post" action="">  
  <P>名      ;:  
    <INPUT type="text" value="张三" size="20" name="fname">  
  </P>  
  <P>姓      ;:  
    <INPUT name="lname" type="text">  
  </P>  
  <P>登录名:  
    <INPUT name="sname" type="text" size="20">  
  </P>  
</FORM>  
</BODY>  
</HTML>
```

2. 密码框

只需将文本框的 type 属性设为 password 就可以创建一个密码框。密码框的各属性和文本框一样,唯一不同的就是密码框输入的全部字符都以 * 显示,如示例 1.3 所示。

示例 1.3

```
<HTML>  
<HEAD>  
<META http-equiv="Content-Type" content="text/html; charset=gb2312">  
<TITLE>密码框</TITLE>  
</HEAD>  
  
<BODY>  
<FORM name="form2" method="post" action="">  
  <P>用户名:  
    <INPUT name="name" type="text" size="21">  
  </P>  
  <P>密      码:  
    <INPUT name="pass" value="123456" type="password" size="22">  
  </P>  
</FORM>  
</BODY>  
</HTML>
```

3. 单选按钮

将文本框表单元素 `type` 属性设为 `radio` 就可以创建一个单选按钮。单选按钮控件用于一组相互排斥的值,组中每个单选按钮控件应具有相同的名称,用户一次只能选择一个选项。只能从组中选定的单选按钮才会在提交的数据中生成 `name/value` 对,单选