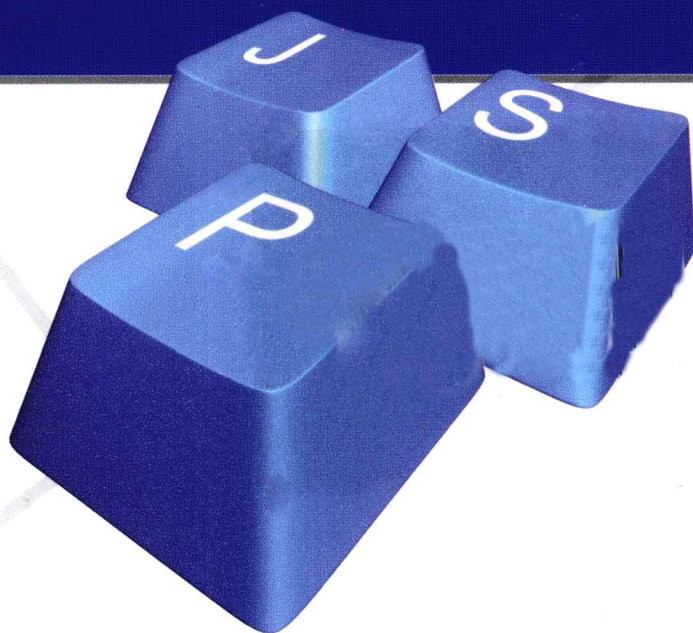


21 世纪面向工程应用型
计算机人才培养规划教材

颜志军 主 编
王玲 胡增顺 副主编

JSP 与 Servlet 程序设计实践教程



清华大学出版社

21 世纪面向工程应用型计算机人才培养规划教材

JSP 与 Servlet 程序设计实践教程

颜志军 主 编
王 玲 胡增顺 副主编

清华大学出版社
北 京

内 容 简 介

本书介绍了基于 Java 技术的动态网页开发技术 JSP 与 Servlet 的相关内容,主要包括 JSP 及 Servlet 程序运行环境的搭建,JSP 的指令、动作、隐含对象,JavaBean,在不同范围内共享数据,Servlet, Cookie,会话跟踪,MVC,JSP 与 Servlet 中的异常处理,自定义标签,国际化与本地化及 JSP 表达式语言。

本书以先易后难的顺序介绍了 JSP 及 Servlet 开发技术的主要技术点,非常适合高校计算机相关专业作为学习 JSP 及 Servlet 开发技术的教材,也可以作为培训机构的培训教材。同时,对于正在使用 JSP 及 Servlet 作为编程语言的开发人员,本书也有一定的参考价值。

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

JSP 与 Servlet 程序设计实践教程/颜志军主编. —北京:清华大学出版社,2012.3

(21 世纪面向工程应用型计算机人才培养规划教材)

ISBN 978-7-302-26865-9

I. ①J… II. ①颜… III. ①JAVA 语言—程序设计 ②JAVA 语言—主页制作—程序设计

IV. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2011)第 191568 号

责任编辑:梁 颖 李 晔

封面设计:杨 兮

责任校对:李建庄

责任印制:李红英

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:北京密云胶印厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19 字 数:477 千字

版 次:2012 年 3 月第 1 版 印 次:2012 年 3 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

产品编号:041813-01

Java 作为面向对象编程语言的杰出代表正被应用在越来越多的领域,从掌上设备到大型机,从嵌入式应用到企业级应用,Java 的身影随处可见。Java 作为开源的编程语言和开发工具,使得开发者无需担心版权和许可费用的问题,这也使得越来越多的项目倾向于使用 Java 作为开发语言。更为重要的是,Java 作为跨平台的语言,其应用可以方便地在不同操作系统平台之间进行移植,这使得那些将来可能更换运行平台的应用会更多地考虑将 Java 作为其开发语言。

在 Web 开发领域,使用 Java 作为脚本语言及底层逻辑支撑的 JSP 技术及 Servlet 技术也继承了 Java 的上述优点,同时它也拥有诸如 Tomcat、JBoss 这样的开源服务器及一些商业服务器支撑其开发与部署,基于不同的设计架构,开发者也可以从 Spring、Struts、Hibernate 等众多的开源框架中获得支持。因而,当前从个人站点到企业级应用,JSP 及 Servlet 都有越来越广泛的应用,特别是在企业级应用中,JSP 及 Servlet 更是有着不可替代的作用。

本书从 JSP 及 Servlet 的开发环境搭建开始讲解 JSP 及 Servlet 的开发技术,从简单的 JSP 页面开发开始,由浅入深逐步介绍 JavaBean、Servlet、自定义标签、国际化与本地化及 JSP 表达式语言等内容。在整个过程中以图解为重要手段,以理论分析为导向,带领读者慢慢步入 JSP 及 Servlet 开发的殿堂,掌握 JSP 及 Servlet 的开发技巧。

本书的内容结构

JSP 与 Servlet 是采用 Java 技术进行 Web 开发的最基础的技术,也是掌握 Java EE 技术的起点。本书以先易后难的顺序依次介绍了 JSP 及 Servlet 的相关技术,从内容上说,本书覆盖了 JSP 及 Servlet 的主要内容,从简单的 JSP 编程到 MVC 架构、从在页面中固化语言字符串到国际化与本地化编程,本书均有涉及。

本书主要分为三大部分:

第一部分包含第 1~8 章的内容,其主要介绍了 JSP 开发技术,其中包括在 JSP 页面中使用 Java 代码、JSP 指令、JSP 动作、JSP 中的隐含对象、Java Bean 在 JSP 中的应用以及如何在 Web 应用中共享数据。通过学习该部分的内容,读者可以完成 JSP 页面的编程,并可以利用 JSP 页面完成一些简单的 Web 应用开发。

第二部分包含第 9~13 章的内容,主要讲述了 Servlet 开发技术以及 MVC 架构,其主要内容包括 Servlet 基础、Servlet 处理客户请求、Cookie、会话跟踪以及 MVC。这一部分内容结合 JSP 开发技术就可以完成较为复杂的 Web 应用开发,并且通过本部分内容的学习,读者将更加深入地了解在进行 Web 开发时如何设计恰当、适用并利于维护与扩展的整体结构。

第三部分包括第 14~17 章的内容,主要介绍了在 JSP 及 Servlet 技术领域较为高级的技术,其主要内容包括异常处理、自定义 JSP 标签、国际化与本地化编程以及 JSP 表达式语言。这一部分内容在实际开发中除了异常处理被经常用到以外,其他部分如果采用第三方框架进行开发往往用到的时候不多,但是学习及理解了这些技术,将对理解那些第三方框架的实现及使用有极大的帮助,它将有助于提升程序整体设计的能力,并对程序质量的提升会有较大帮助。

全书由颜志军统稿,其中第 14 和 15 章由胡增顺编写,其余章节由颜志军和王玲编写。

本书的读者对象

本书以步步图解的方式对 JSP 及 Servlet 开发环境搭建、JSP 及 Servlet 程序执行过程等进行注解说明,非常适合于学习 JSP 及 Servlet 的初学者学习。如果您是以下类型的读者,此书会适合您:

1. 初步掌握了 Java 技术,但对 JSP 及 Servlet 所知甚少或者一无所知,但希望学习 JSP 及 Servlet 开发技术的读者。

2. 对 JSP 及 Servlet 有部分了解,但并不清楚其实质、原理的读者。

3. 对 JSP 及 Servlet 比较熟悉,但希望了解更多技术细节的读者。

同时,本书也适合希望使用 Java 从事 Web 开发、培训等相关工作的朋友。

致谢

本书能够编写完成,首先要感谢我的妻子,没有她的支持与鼓励,本书可能不会最终完成;也要感谢我 6 岁的女儿,正是她令我心静如水,能够潜心写作;最后,要感谢清华大学出版社的编辑梁颖先生,梁先生的建议令本书增色不少,同时也正是梁先生的信任使得本书能够以作者期望的结构、形式出现在读者面前。

读者在使用本书的过程之中,如果有任何的疑问,欢迎来信探讨。同时,由于作者学识所限,书中表述不清甚至错误也在所难免,欢迎读者来信批评指正。作者的电子邮箱是 ansreader@126.com。

颜志军

2011 年 12 月

第 1 章 网页开发技术概述	1
1.1 HTML 简介	1
1.1.1 HTML 的基本结构	1
1.1.2 头部分	2
1.1.3 体部分	3
1.2 动态网页技术	10
1.3 JSP 动态网页技术优势	10
1.4 小结	11
1.5 练习	12
第 2 章 JSP 及 Servlet 程序运行环境搭建	13
2.1 JDK 安装及环境变量配置	13
2.2 Tomcat 安装及配置	16
2.3 通过 JSP 简单示例网页验证 Tomcat 环境配置	18
2.4 小结	19
2.5 练习	20
第 3 章 在 JSP 中调用 Java 代码	21
3.1 基本动态技术元素	21
3.2 Java 类	25
3.3 JSP 页面的底层工作机制	26
3.4 小结	30
3.5 练习	30
第 4 章 JSP 指令	31
4.1 page 指令	31
4.2 include 指令	34
4.3 小结	36
4.4 练习	36
第 5 章 JSP 动作	38
5.1 jsp:include 动作	38



5.2	jsp:forward 动作	41
5.3	jsp:plugin 动作	44
5.4	小结	46
5.5	练习	46
第 6 章 JSP 隐含对象		47
6.1	request	47
6.1.1	取得网页请求参数	47
6.1.2	取得请求头部信息	53
6.1.3	取得客户和主机信息	56
6.2	response	58
6.2.1	设置 Content-Type	58
6.2.2	发送错误响应至客户端	59
6.2.3	重定向	61
6.3	session	62
6.4	application	63
6.5	out	65
6.6	config	68
6.7	小结	69
6.8	练习	69
第 7 章 Java Bean 在 JSP 中的应用		70
7.1	Bean 的基本概念	70
7.2	Bean 的应用	72
7.2.1	简单工具 Bean 示例	73
7.2.2	在 Bean 中实现数据库连接池	75
7.3	Bean 的属性及应用	80
7.3.1	值 Bean 的特性	80
7.3.2	值 Bean 的应用	81
7.4	小结	86
7.5	练习	86
第 8 章 共享数据		87
8.1	在页面间传递数据	87
8.1.1	通过请求在页面间传递数据	87
8.1.2	通过 Java Bean 在页面间传递数据	89
8.2	会话共享数据	91
8.2.1	会话简介	91
8.2.2	在会话对象中保存数据	92

8.2.3 在会话中共享 Java Bean	94
8.3 用户间共享数据	96
8.3.1 在应用中共享数据	96
8.3.2 在应用中共享 Java Bean	97
8.4 小结	98
8.5 练习	98
第 9 章 Servlet 基础	99
9.1 Servlet 与 JSP	99
9.1.1 Servlet 的功用	99
9.1.2 Servlet 与 JSP 的比较及选择	100
9.2 使用 Servlet 生成纯文本	101
9.2.1 编写 Servlet 代码	101
9.2.2 在服务器中部署 Servlet	102
9.2.3 Servlet 工作流程简析	104
9.3 使用 Servlet 生成 HTML	105
9.4 Servlet 生命周期	106
9.4.1 Servlet 高效率的原因	106
9.4.2 Servlet 生命周期探究	106
9.4.3 特殊的 Servlet——过滤器	111
9.4.4 Servlet 应用实例	112
9.5 小结	116
9.6 练习	116
第 10 章 Servlet 处理客户请求	117
10.1 Servlet 读取客户请求参数	117
10.1.1 Servlet 遍历请求对象中的参数	117
10.1.2 Servlet 读取表单参数	119
10.2 根据请求参数自动填充表单 Bean	121
10.2.1 Java Bean 及 JSP 填充表单 Bean 回顾	121
10.2.2 在 Servlet 中实现值 Bean 的填充	121
10.2.3 参数缺失时重新显示表单	124
10.3 Servlet 处理客户请求实例	128
10.4 小结	134
10.5 练习	134
第 11 章 Cookie	135
11.1 Cookie 简介	135
11.1.1 Cookie 的作用	135

11.1.2	Cookie 存在的问题	136
11.1.3	删除浏览器保存的 Cookie 及禁用 Cookie	137
11.2	Cookie 的生成与读取	141
11.2.1	服务端使用 Cookie 的过程	141
11.2.2	生成与读取 Cookie	142
11.3	Cookie 的属性	144
11.3.1	Cookie 的 name 和 value 属性	144
11.3.2	Cookie 的 maxAge 属性	145
11.3.3	Cookie 的其他属性	148
11.4	Cookie 常用实例	150
11.4.1	记录用户的偏好	150
11.4.2	保存登录账号与密码	152
11.5	小结	156
11.6	练习	156
第 12 章	会话跟踪	157
12.1	Web 应用中的会话	157
12.1.1	会话存在的必要性	157
12.1.2	会话跟踪的实现方式	158
12.2	利用会话跟踪保持访问信息	160
12.2.1	在会话中保存与读取数据	160
12.2.2	会话属性与会话控制	162
12.3	会话配置及会话使用注意事项	165
12.3.1	在 Tomcat 中配置会话	165
12.3.2	会话使用注意事项	165
12.4	会话跟踪实例	166
12.4.1	利用会话协调页面行为	166
12.4.2	简单购物车	169
12.5	小结	176
12.6	练习	176
第 13 章	模型-视图-控制构架	177
13.1	MVC 简介	177
13.1.1	Web 应用程序结构	177
13.1.2	MVC 框架特征	178
13.2	实现 MVC 框架的一般步骤	180
13.2.1	定义表示数据的 Java Bean	180
13.2.2	使用 Servlet 处理请求	181
13.2.3	填充 Java Bean	182

13.2.4	将 Java Bean 存储到请求、会话或者 Servlet 上下文	183
13.2.5	转发请求	184
13.2.6	JSP 页面从 Java Bean 中提取数据	185
13.2.7	MVC 框架时序流程	186
13.3	MVC 框架实例	187
13.4	小结	194
13.5	练习	195
第 14 章	异常处理	196
14.1	利用 Java 的异常处理机制处理异常	196
14.1.1	Java 异常处理机制回顾	196
14.1.2	在 Servlet 和 Java Bean 中进行异常处理	198
14.2	JSP 错误页面	200
14.2.1	在 JSP 页面中使用 Java 异常处理机制	200
14.2.2	使用 JSP 错误页面显示错误信息	202
14.3	在 Tomcat 中配置错误页面	204
14.3.1	配置标准错误页面	204
14.3.2	配置特定异常错误页面	206
14.4	异常处理实例	213
14.5	小结	217
14.6	练习	217
第 15 章	自定义 JSP 标签	218
15.1	简单自定义 JSP 标签实例	218
15.1.1	实现自定义 JSP 标签 Java 代码	218
15.1.2	配置标签库	219
15.1.3	使用自定义 JSP 标签	220
15.2	为自定义 JSP 标签添加属性	221
15.2.1	自定义标签通过属性增加灵活性	221
15.2.2	定义自定义标签的属性	221
15.2.3	自定义标签处理程序获取属性并使用	222
15.3	处理自定义 JSP 标签的标签体	223
15.3.1	标签体的概念及需求	223
15.3.2	定义包含标签体的自定义标签	224
15.3.3	自定义标签处理程序处理标签体	225
15.4	自定义标签总结	226
15.4.1	标签定义总结	226
15.4.2	标签处理程序总结	228
15.5	自定义标签实例	229

15.6	小结	232
15.7	练习	232
第 16 章	国际化与本地化编程	233
16.1	国际化编程	233
16.1.1	Java 对国际化的支持	233
16.1.2	国际化 Web 应用	234
16.2	本地化编程	236
16.2.1	为 Web 应用准备本地化字符串	236
16.2.2	在 Servlet 中使用本地化文本	238
16.2.3	在 JSP 页面中使用本地化文本	239
16.3	国际化与本地化编程实例	241
16.4	小结	245
16.5	练习	245
第 17 章	JSP 表达式语言	246
17.1	表达式语言简介	246
17.1.1	表达式语言的简单示例	246
17.1.2	表达式语言的功能	247
17.2	配置表达式语言及标准脚本元素的有效性	247
17.2.1	禁止表达式语言执行	247
17.2.2	禁止标准脚本元素的使用	250
17.3	表达式语言访问变量	251
17.3.1	表达式语言访问 Java Bean 的属性	251
17.3.2	表达式语言访问不同作用域内的变量	255
17.3.3	表达式语言访问集合中的变量	257
17.3.4	表达式语言访问隐含对象	259
17.4	表达式语言的运算符	260
17.4.1	算术运算符	260
17.4.2	关系运算符	261
17.4.3	逻辑运算符	261
17.4.4	运算符综合例程	261
17.5	小结	262
17.6	练习	263
附录	课后习题参考答案	264
参考文献		292

网页开发技术概述

计算机技术的出现引发了诸多领域的变革,而 Internet 的出现又可称为计算机领域内的变革。Internet 能够深入当前这个世界的每一个角落,又与网页开发技术密不可分。本书将从本章开始由浅入深地介绍网页开发技术。

1.1 HTML 简介

HTML(Hypertext Mark-up Language,超文本标记语言)使用文本标记的形式来说明需要展示内容的显示特性,而浏览器能够解释这些标记并以标记所指定的形式展示相应的内容。

1993 年 6 月 HTML 发布第 1 版,而到 2008 年 1 月时 HTML5 已经发布,在此期间,XHTML(可扩展的超文本标记语言,通常它比 HTML 具有更严格的书写标准,具备更好的跨平台能力,它也属于 HTML 家族)也形成标准并且发布。而无论采用静态网页技术还是动态网页技术,最终被浏览器解释的都是 HTML 代码。全面掌握 HTML 需要一本独立的教材才能完成,本节简要介绍 HTML 的相关情况,掌握基本的 HTML 技术是学习 JSP 及 Servlet 技术的前提条件。

1.1.1 HTML 的基本结构

HTML 分为头部与正文两部分。头部主要说明当前网页的字符集、关键字、网页描述等信息,这些信息不会被显示在网页上,但是它控制着网页整体的一些特性,而正文部分则是网页中可以展现出来供人们在浏览器中阅读的部分。

通常一个 HTML 网页的框架结构如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

此行说明当前HTML文档所使用的HTML版本,但是对于主流浏览器,省略此部分并不会影响对HTML的解析

标记<head>与</head>之间是HTML网页的头部

标记<body>与</body>之间是HTML网页的正文部分

注意：在标记语言中，标记通常是成对出现的，其开始标记放置于“<...>”之中，而结束标记放置于“</...>”之中。如果一个标记不需要在两处分别标明开始与结束，则可以将标记置于“<.../>”之中。

编写完 HTML 网页文件后在存储为文件时，可以选择 .htm 或 .html 作为其后缀。

在编写 HTML 网页时，早期人们习惯将需要展示在浏览器中的内容与展示形式一起写在 HTML 网页文件之中，但是近些年来，将需要展示的内容与展示形式进行分离的做法越来越流行。也就是说，现在通常将需要展示的内容写在 HTML 文件的正文部分，而其展示形式则被写入一个称为“样式表”的文件（.css 文件）之中。如果样式表中包含的内容不多，或者该样式仅供一个特定的 HTML 文件所使用，也可以将这些样式写入 HTML 文件的头部分。

1.1.2 头部分

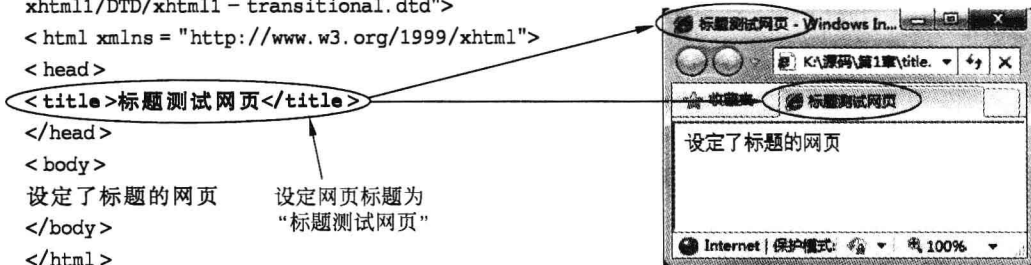
头部分通常包含 title 与 meta 标记，title 标记用于设定当前网页的标题，而 meta 标记则可以通过属性完成不同的功能。

1. title 标记

title 标记的作用是设定当前网页显示的标题，该标题会被显示于浏览器的窗口标题栏，支持多标签窗口的浏览器也会将该标题显示于标签标题栏。

一个使用 title 标记的样例 HTML 文件如下：

```
<!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>
<title>标题测试网页</title>
</head>
<body>
  设定了标题的网页
</body>
</html>
```



设定了标题的网页

设定网页标题为“标题测试网页”

通常将 HTML 网页文件的标题设定为当前网页展示内容的主题，一般搜索引擎在判断一个网页的内容时也会特别注意 title 标记之中的内容，因此如果期望网页在搜索引擎中有较好的排名，应当仔细设计 title 标记之中的内容。

注意：搜索引擎在对网页按关键字进行排名时使用了较复杂的算法，但是不同的搜索引擎有不同的算法，同一搜索引擎也会时常更改其算法。而通常 title 标记都是这些算法考虑的重要因素之一。

title 标记并非 HTML 的必须组成部分，如果一个网页省略 title 标记，通常浏览器会将该网页的全路径作为该网页标题。例如，在 Firefox 中标签页标题栏会显示该网页的路径与文件名；而在 IE8 中无论标签页的标题栏还是 IE8 本身的标题栏都将显示该网页的全路径。

2. meta 标记

meta 标记用于表明一组元数据，元数据并不是当前网页文档的内容，而是说明了一些与当前网页文档相关的信息，比如作者信息、网页关键字等。

meta 标记的属性包含有 name、content、scheme 与 http-equiv 及很少用到的 lang。通

常 name 属性与 content 属性会配对使用, name 属性说明当前要表达的信息(参数), 而 content 属性则说明信息的内容(参数值)。

例如, 以下 meta 标记分别表明了网页的关键字、描述及作者信息:

```
<meta name="keywords" content="JSP,Servlet">
<meta name="description" content="本站介绍了 JSP 及 Servlet 的基本开发技术" />
<meta name="author" content="颜志军,ansreader@126.com" />
```

而 http-equiv 属性与 content 属性配对使用时, 则可以设定一些对浏览器有用的信息。例如, 以下 meta 标记表明了当前网页使用的字符集是 gb2312。

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

注意: 浏览器以 meta 标记中指定的网页字符集显示当前网页, 如果字符集设定错误, 通常网页将显示为空白页或乱码。

1.1.3 体部分

HTML 网页的体部分(正文部分)是网页要展示的部分, 也是静态网页技术中要掌握的重点内容。本小节将重点介绍其中的一些常用标记。

1. 标题标记(h)

HTML 提供了标题标记 h 用以表明一段文字是一个标题或者是一段被强调的文字。HTML 提供的标题标记一共有六级, 即 h1~h6, h1 是级别最高的标题, 其字体最大; h6 则是级别最低的标题, 其字体最小。

以下代码演示了 6 级标题的用法及其在 IE 中显示的默认效果:

```
<!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=gb2312" />
<title>标题标记示例</title>
</head>
<body>
<h1>一级标题(h1)</h1>
<h2>二级标题(h2)</h2>
<h3>三级标题(h3)</h3>
<h4>四级标题(h4)</h4>
<h5>五级标题(h5)</h5>
<h6>六级标题(h6)</h6>
</body>
</html>
```



注意: 通常 h1~h6 的字体按从大到小的顺序排列, 但是也可以通过层叠样式表(CSS)改变其风格。层叠样式表是定义的一组格式设置规则, 通过层叠样式表可以将网页的内容与表现形式分离, 并且可以达到精简网页, 加快浏览器加载网页的目的。

2. 段落标记及换行标记(p 及 br)

在 HTML 语言中, 可以用标记 p 来表明一个段落。P 的开始标记<p>表明段落的开始

位置,而 p 的结束标记</p>则表明段落的结束位置。在 HTML 中两个段落之间通常会有相对较大的行距。

如果希望在一个段落之中换行,则需要使用到换行标记 br,br 的开始与结束位置必须在一个标记之中,即 br 通常只能以
的形式出现。换行后的两行文字通常行距较小。

一个包含段落标记 p 及换行标记 br 的 HTML 样例 HTML 文件如下:

```
<!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=gb2312" />
<title>见与不见</title>
</head>
<body>
<h3>见与不见</h3>
<p>你见,或者不见我<br/>
我就在那里<br/>
不悲不喜</p>
<p>你念,或者不念我<br/>
情就在那里<br/>
不来不去</p>
<p>你爱,或者不爱我<br/>
爱就在那里<br/>
不增不减</p>
<p>你跟,或者不跟我<br/>
我的手就在你手里<br/>
不舍不弃</p>
<p>来我的怀里<br/>
或者<br/>
让我住进你的心里<br/>
默然 相爱<br/>
寂静 欢喜</p>
</body>
</html>
```

显示时文字将换行,与源代码中的换行没有关系,其效果是由
产生的

<p>与</p>之间是一个完整的段落,段落前后有较大的行距

3. 超链接标记(a)

超链接是 HTML 中的核心技术,网页之间的跳转绝大部分都是通过超链接完成的。超链接的标记符号是 a,超链接标记 a 的属性 href 可以指定超链接的目标页面,而在开始标记<a>和结束标记之间的文字则是在网页上显示的超链接文字,用鼠标单击它网页就可以跳转到 href 所指定的页面。超链接标记 a 的另一个常用属性是 target,属性 target 可以指定单击超链接后新的页面的显示位置,其允许的值如表 1-1 所示。

表 1-1 target 属性值及其含义

target 属性值	含 义
_blank	超链接目标页面将在新窗口中打开
_parent	超链接目标页面将在父窗口中打开
_self	超链接目标页面将在当前窗口中打开,这也是 target 属性的默认值
_top	超链接目标页面在顶层窗口中打开

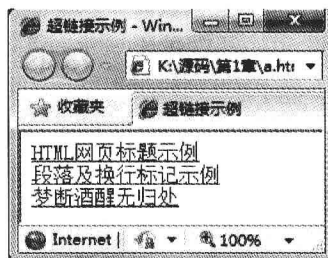
注意：父窗口与顶层窗口的含义需要在了解框架的概念后才能理解，但框架在动态网页技术中有更好的替代方案，所以框架不被推荐使用，因此本书将不会涉及框架的内容。

以下网页文件包含了三个超链接：

```
<!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=gb2312" />
<title>超链接示例</title>
</head>
<body>
<a href="title.html">HTML 网页标题示例</a><br/>
<a href="p.html" target="_blank">段落及换行标记示例</a><br/>
<a href="http://www.yanzhijun.net" target="_blank">梦断酒醒无归处</a>
</body>
</html>
```

通常属性href的值是一个网页的相对路径地址

属性href的值也可以是一个绝对路径，甚至其他站点上的一个路径



注意：网站中的文件路径分隔符是“/”，它与 Windows 下的路径分隔符“\”不同，在使用相对路径和绝对路径时都要注意这一点。在 href 属性中使用相对路径时也可以利用“..”来表示上一级目录(父目录)。

4. 显示图片标记(img)

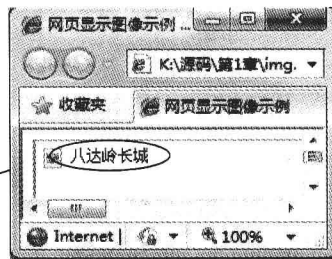
在浏览器中展示的网页不仅可以显示不同字形字号的文字，而且可以同时包含有图片、声音甚至视频等多媒体元素。而这其中图片是最为常见的多媒体元素。在 HTML 代码中可以利用标记 img 为网页加入图片元素。

标记 img 最重要的属性是 src，它指定了要在网页中显示的图片文件的存储位置，该图片既可以是当前网站中的图片文件，也可以是当前站点之外保存于其他站点之中的图片文件。默认情况下图片将以图片本身的尺寸在浏览器中展示，而如果图片显示的尺寸需要改变，或者试图改变图片的长宽比，则可以通过 img 标记的属性 width 与 height 分别指定图片在网页中显示的宽度与高度。

以下是仅包含有一个图片的示例网页代码：

```
<!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=gb2312" />
<title>网页显示图像示例</title>
</head>
<body>

</body>
</html>
```



注意：属性 alt 在大多数情况下并无太大用处，但是当浏览器禁止图像显示，或者网速



5. 表格

表格通常用于展示一系列的数据,它由若干行和若干列单元格组成。早期的 HTML 编程中也经常利用表格对页面进行布局控制(当前依然有大量的开发者在利用表格进行布局控制,但是现在更鼓励通过 div 与 css 进行布局控制)。表格最基本的标记是 table、tr 与 td,它们分别代表表格、表格中的一行和一行中的一列。表格还有一个名为 th 的标记,用于表示表格中一列的表头。

表格相关的标记还拥有较多的属性,用以控制表格的表现形式和风格。其中较常用的有控制表格边框宽度的属性 border、用于跨越多行或多列表格单元格的属性 rowspan 和 colspan 等。

以下是一个简单表格的示例代码：

<!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 = gb2312" />

<title>基本表格示例</title>

</head>

<body>

<table border = "1">

<tr>

<td>孙权</td>

<td>鲁肃</td>

<td>周瑜</td>

</tr>

<tr>

<td>刘备</td>

<td>诸葛亮</td>

<td>赵子龙</td>

</tr>

<tr>

<td>曹操</td>

<td>司马懿</td>

<td> </td>

</tr>

</table>

</body>

</html>

table的border属性说明表格线的宽度，如果不指定或者将其值设定为0，则将不显示表格线

从开始标记<tr>到结束标记</tr>是表格的一行

从开始标记<td>到结束标记</td>是表格一行中的一列

" "是HTML中所谓的实体引用的一种，它表示在显示网页时显示一个空格

注意：如果希望在 HTML 网页中显示空格，需要使用符号“ ”对需要显示空格的位置进行替换。如果需要显示多个空格，则需要使用多个“ ”。而在网页源文件中直接使用连续的多个空格时，浏览器将仅显示一个空格。

在实际应用表格时,经常需要将部分相邻的单元格进行合并。单元格列合并可以通过 标记的属性 colspan 来实现,而行合并则可以通过 标记的属性 rowspan 实现。属性 colspan 与 rowspan 的值则表明要合并的列或行的数量。 | |