

普通高等教育“十三五”规划教材
Web应用&移动应用开发系列规划教材

Java EE YINGYONG
KAIFA JIAOCHENG

Java EE应用开发教程

● 编著 吴志祥 张 智 曹大有
焦家林 赵小丽



华中科技大学出版社
<http://www.hustp.com>

普通高等教育“十三五”规划教材
Web应用&移动应用开发系列规划教材

Java EE YINGYONG
KAIFA JIAOCHENG

Java EE应用开发教程

编 著 吴志祥 张 智 曹大有
焦家林 赵小丽
参 编 柯 鹏 鲁屹华 王 静
张新华 董剑波 肖 念



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

内 容 提 要

本书系统地介绍了 Java EE 应用开发的基础知识和实际应用,共分 9 章,包括 Web 应用开发基础知识、JSP 技术、Servlet 与 MVC 开发模式、表现层框架 Struts 2、Hibernate 和 MyBatis 等 ORM 框架、Spring 框架与 SSH 整合、Spring MVC 框架、企业级 Java Bean 开发和项目管理工具 Maven 的使用等。

本书以实用为出发点,其内容从简单到复杂,循序渐进,结构合理,逻辑性强。每个知识点都有精心设计的典型例子说明其用法,每章配有标准化的练习题及其答案、实验指导。与本书配套的教学网站,包括教学大纲、实验大纲、各种软件的下载链接、课件和案例源代码下载、在线测试等。

本书可以作为高等院校计算机专业和非计算机专业学生的 Web 开发教材,也可以作为 Web 编程爱好者的参考书。

图书在版编目(CIP)数据

Java EE 应用开发教程/吴志祥等编著. —武汉:华中科技大学出版社,2016.8

普通高等教育“十三五”规划教材

ISBN 978-7-5680-1495-3

I. ①J… II. ①吴… III. ①JAVA 语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2015)第 321878 号

Java EE 应用开发教程

Jave EE Yingyong Kaifa Jiaocheng

吴志祥 张 智 曹大有 焦家林 赵小丽 编著

策划编辑:康 序

责任编辑:史永霞

封面设计:抱 子

责任监印:朱 玢

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321913

录 排:华中科技大学惠友文印中心

印 刷:武汉市籍缘印刷厂

开 本:787mm×1092mm 1/16

印 张:18.25

字 数:455 千字

版 次:2016 年 8 月第 1 版第 1 次印刷

定 价:42.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

目前,市场上关于 Java EE 的相关书籍比较多,但真正从零基础开始、内容简明而又系统的教材并不多见。为此,笔者组织一线相关教师编写了这本符合高校教学需要的 Java EE 教材。

本书系统地介绍了 Java EE 应用开发的基础知识和实际应用,共分 9 章,包括 Web 应用开发基础知识、JSP 技术、Servlet 与 MVC 开发模式、表现层框架 Struts 2、Hibernate 和 MyBatis 等 ORM 框架、Spring 框架与 SSH 整合、Spring MVC 框架、企业级 Java Bean 开发和项目管理工具 Maven 的使用等。其内容从简单到复杂,循序渐进,结构合理,逻辑性强。

本书以实用为出发点,每个知识点都有精心设计的典型例子说明其用法,每章后配有习题及实验。与本书配套的教学网站,包括了教学大纲、实验大纲、各种软件的下载链接、课件和案例源代码下载、在线测试等,极大地方便了教与学。

本书写作特色鲜明,一是教材结构合理,作者对教材目录设置进行了深思熟虑的推敲,在正文中指出了相关章节知识点之间的联系;二是知识点介绍简明,例子生动并紧扣理论,很多例子是作者精心设计的;三是在教材中通过大量的截图,清晰地反映了用户库、jar 包、软件包、类(或接口)四个软件层次;四是通过综合案例的设计与分析,让学生综合使用 Java EE 应用开发的各个知识点;五是有配套的上机实验网站,包括实验目的、实验内容、在线测试(含答案和评分)和素材的提供等。

本书可以作为高等院校计算机专业和相关专业学生学习“Java EE 架构”等课程的教材,也可以作为 Web 开发者的参考书。

Java EE 开发比其他 Web 开发门槛高,具体表现在以下几个方面。

(1) 对 Java 编程知识有较高要求,如 Java 集合框架、泛型、可变参数、单例模式与工厂模式等,在 Java EE 开发中均会涉及。

(2) Java Web 项目可以使用不同的框架方式,它们依赖大量的 jar 包,其版本的不同可能带来问题。例如,使用 Spring MVC 3.2 完成的项目,换成 Spring MVC 4.1 后,可能会由于 JSON 包的版本没有相应提升而出现 Ajax 请求失效的现象。

(3) 各种框架在提高开发效率的同时,也屏蔽了许多技术细节,如基于对象查询的 Hibernate 框架、实现系统松耦合、易于维护的 Spring 框架。

(4) Java EE 开发所涉及的软件和技术特别多,如集成开发环境、数据库、构建企业级应用的多种服务器软件、CSS 样式技术、JS 技术、Ajax 技术和管理项目 jar 的 Maven 技术等。

(5) 需要掌握单元测试和动态调试模式的使用,特别是后者,以便在程序运行中出现异常时能跟踪调试。

本书由吴志祥和张智老师整体构思,并与曹大有、焦家林和赵小丽等老师共同编著,具体分工如下:曹大有老师完成了第一、二章的编写,赵小丽老师完成了第三、九章的编

写，吴志祥和张智老师共同完成了第四、五、六章的编写，焦家林老师完成了第七、八章的编写。

本书可以作为高等院校计算机专业和非计算机专业学生的 Web 开发教材，也可以作为 Web 编程爱好者的入门参考书。

获取本书配套的课件、案例源代码等教学资料，可访问 <http://www.wustwzx.com>。

需要特别感谢的是孙陈同学，他全程参与了教材案例的设计与测试。

由于编者水平有限，书上错漏之处在所难免，在此真诚欢迎读者多提宝贵意见，读者可通过访问作者的教学网站 <http://www.wustwzx.com> 与作者 QQ 联系，以便再版时更正。

编者

2016 年 8 月于武汉

目 录

第 1 章 Web 应用开发基础.....	1
1.1 网站与网页基础	1
1.1.1 Web 应用体系与 B/S 模式	1
1.1.2 常用 HTML 标记及其使用	2
1.1.3 CSS 样式与 Div 布局	3
1.1.4 客户端脚本 JavaScript、jQuery 及 Ajax	7
1.2 Java 与 Java EE 概述	10
1.2.1 Java 与 JDK	10
1.2.2 Java EE/Web 及其开发模式	11
1.3 搭建 Java Web 应用的开发环境	14
1.3.1 使用绿色版的 Web 服务器 Tomcat 7	14
1.3.2 下载、安装和配置 MyEclipse 2013	15
1.3.3 MyEclipse 若干快捷操作	20
1.3.4 创建、部署和运行一个简单的 Web 项目	20
1.3.5 Java Web 项目结构分析	23
1.3.6 Java Web 项目中中文乱码产生原因及解决方案	23
1.4 MySQL 数据库及其服务器	24
1.4.1 数据库概述与其 MySQL 安装	24
1.4.2 MySQL 前端工具 SQLyog	25
1.4.3 在 Java 项目中以 JDBC 方式访问 MySQL 数据库	26
1.4.4 封装 MySQL 数据库访问类	28
1.5 Java 单元测试与动态调试	30
1.5.1 单元测试 JUnit 4	30
1.5.2 动态调试模式 Debug	31
习题 1	32
实验 1 Web 应用开发基础	33
第 2 章 使用纯 JSP 技术开发 Web 项目	35
2.1 JSP 页面概述	35
2.1.1 JSP 页面里的 page 指令	36
2.1.2 JSP 脚本元素：声明、表达式和脚本程序	36
2.1.3 文件包含指令 include	37
2.1.4 引入标签库指令 taglib	37

2.1.5	JSP 动作标签	38
2.2	JSP 内置对象与 Cookie 信息	41
2.2.1	向客户端输出信息对象 out.....	41
2.2.2	响应对象 response.....	41
2.2.3	请求对象 request.....	42
2.2.4	会话对象 session.....	43
2.2.5	应用的共享对象 application.....	46
2.2.6	页面上下文对象 pageContext	48
2.2.7	Cookie 信息的建立与使用*	49
2.3	表达式语言 EL 与 JSP 标准标签库 JSTL	51
2.3.1	表达式语言 EL.....	51
2.3.2	JSP 标准标签库 JSTL.....	52
2.4	纯 JSP 技术实现的会员管理项目 MemManal	53
2.4.1	项目总体设计及功能.....	53
2.4.2	项目若干技术要点.....	54
2.4.3	Web 项目中 JSP 页面的动态调试方法	58
	习题 2	59
	实验 2 使用纯 JSP 技术开发 Java Web 项目	61
第 3 章	使用 MVC 模式开发 Web 项目	63
3.1	JavaBean 与 MV 开发模式.....	63
3.1.1	JavaBean 规范与定义	63
3.1.2	与 JavaBean 相关的 JSP 动作标签	64
3.1.3	MV 开发模式.....	65
3.1.4	使用 MV 模式开发的会员管理系统 MemMana2.....	70
3.2	Servlet 组件	72
3.2.1	Servlet 定义及其工作原理	72
3.2.2	Servlet 协作与相关类和接口	73
3.2.3	基于 HTTP 请求的 Servlet 开发	75
3.3	Servlet 基本应用	77
3.3.1	使用 Servlet 处理表单	77
3.3.2	Servlet 作为 MVC 开发模式中的控制器	78
3.3.3	使用 Servlet 实现文件下载*	79
3.3.4	使用 FileUpload 实现文件上传*.....	82
3.4	基于 MVC 模式开发的会员管理项目 MemMana3	88
3.4.1	项目总体设计及功能.....	88
3.4.2	项目若干技术要点.....	88
3.4.3	MVC 项目里程序的分层设计(DAO 模式)	94
3.5	Servlet 监听器与过滤器*	98

3.5.1	Servlet 监听器与过滤器概述	98
3.5.2	使用接口 HttpSessionListener 统计网站在线人数	101
3.5.3	使用接口 Filter 进行身份认证	102
3.5.4	使用接口 Filter 统一网站字符编码	104
习题 3		107
实验 3	使用 MVC 模式开发 Web 项目	109
第 4 章	Web 表现层框架 Struts 2	111
4.1	Struts 2 框架及其基本使用	111
4.1.1	Struts 2 框架实现原理	111
4.1.2	建立 Struts 2 用户库	112
4.1.3	Struts 2 框架的主要接口与类	114
4.1.4	Struts 2 框架配置	115
4.1.5	控制器里数据的自动接收与转发	117
4.2	使用 Struts 标签显示转发数据	124
4.2.1	Struts 标签库概述	124
4.2.2	UI 标签	125
4.2.3	数据标签 set 和 property	126
4.2.4	控制标签 if/elseif/else	126
4.2.5	循环标签 iterator	127
4.2.6	标签 bean 与 param	127
4.2.7	标签 action	128
4.2.8	Ajax 标签 datetimepicker 和 tree	128
4.3	Struts 2 拦截器	129
4.3.1	Struts 拦截器的工作原理	129
4.3.2	自定义拦截器及其配置	130
4.3.3	拦截器应用示例	130
4.4	Struts 输入校验	133
4.4.1	客户端验证与服务器端验证	133
4.4.2	使用 Struts 内置校验	133
4.5	基于 Struts 2 框架开发的会员管理项目 MemMana4	137
4.5.1	项目总体设计	137
4.5.2	使用 Ajax 技术处理管理员登录	138
4.5.3	Struts 文件上传	141
习题 4		147
实验 4	在 Web 项目里使用 Struts 2 框架	148
第 5 章	对象关系映射工具 ORM	149
5.1	对象关系映射 ORM 与对象持久化	149
5.2	Hibernate 框架及其基本使用	150

5.2.1	创建 Hibernate 用户库	152
5.2.2	Hibernate 主要接口与类	153
5.2.3	创建映射文件	155
5.2.4	编写 Hibernate 配置文件	155
5.2.5	在 Java 项目中使用 Hibernate 框架的一个简明示例	156
5.3	在 Java Web 项目中使用 Hibernate 框架	159
5.3.1	创建 Hibernate 工具类	159
5.3.2	封装分页类 Pager	159
5.3.3	封装使用 Hibernate 实现的数据库访问类 MyDb	162
5.3.4	基于 Hibernate 框架开发的会员管理项目 MemMana4_h	166
5.4	Java 对象持久化 API——JPA	171
5.4.1	JPA 是一种 ORM 产品规范	171
5.4.2	JPA 的主要接口与类	172
5.4.3	JPA 使用基于注解的模型类	173
5.4.4	JPA 配置文件 persistence.xml	174
5.4.5	JPA 规范+Hibernate 框架实现的数据库访问类设计	174
5.4.6	使用 JPA 开发的会员管理项目 MemMana4_jpa	177
5.5	持久化框架 MyBatis	179
5.5.1	MyBatis 概述及主要 API	179
5.5.2	使用 MyBatis 的主要步骤	180
5.5.3	使用 MyBatis 开发的会员管理项目 MemMana4_mybatis	182
习题 5		186
实验 5	持久化框架的使用	187
第 6 章	Spring 框架与 SSH 整合	189
6.1	Spring 简介	189
6.1.1	软件设计的单例模式与简单工厂模式	189
6.1.2	控制反转 IoC	190
6.1.3	面向切面 AOP	190
6.2	Spring 框架的基本使用	191
6.2.1	创建 Spring 用户库	191
6.2.2	Spring 框架的主要类与接口	191
6.2.3	Spring 配置文件	192
6.2.4	使用 Spring 配置文件的两种方式	193
6.2.5	测试 Spring 依赖注入的 Hello 程序	193
6.3	使用 Spring 整合的 Web 项目	196
6.3.1	Spring 整合 Struts 2	196
6.3.2	Spring 整合 Hibernate	200
6.3.3	SSH 整合	200

6.4 使用 SSH 整合的会员管理项目 MemMana6_ssh.....	202
6.4.1 项目总体设计	202
6.4.2 主要功能实现	206
习题 6	209
实验 6 Spring 框架与 SSH 整合	210
第 7 章 Spring MVC 框架	211
7.1 Spring MVC 及其环境搭建	211
7.1.1 Spring MVC 概述	211
7.1.2 创建 Spring MVC 3.2 用户库	212
7.1.3 Spring MVC 项目配置	212
7.1.4 Spring MVC 框架配置文件	214
7.2 Spring MVC 框架工作原理	216
7.2.1 Spring MVC API	216
7.2.2 Spring MVC 控制器及方法注解	217
7.2.3 Spring MVC 工作原理	217
7.3 Spring MVC 文件上传与 Ajax	218
7.3.1 Spring MVC 文件上传	218
7.3.2 Spring MVC 处理 Ajax 请求	220
7.4 基于 Spring MVC 的会员管理项目 MemMana7_h	222
7.4.1 项目整体设计	222
7.4.2 项目详细设计	226
习题 7	235
实验 7 Spring MVC 框架的使用	236
第 8 章 企业级 Java Bean 开发	237
8.1 EJB 与分布式应用	237
8.1.1 EJB 概述	237
8.1.2 分布式多层应用架构	238
8.1.3 EJB 相关类	239
8.2 JNDI 与对象序列化	239
8.2.1 Java 命名与目录接口 JNDI	239
8.2.2 对象序列化	240
8.3 创建 EJB 服务器端	241
8.3.1 服务器软件 JBoss 下载与配置	241
8.3.2 EJB 中的三种 Bean 及其状态设置	242
8.3.3 设置远程/本地服务接口	244
8.3.4 创建 EJB 服务器端项目、配置数据源	244
8.3.5 部署 EJB 服务器端项目	247
8.4 创建 EJB 客户端	247

8.4.1 创建 EJB 客户端的一般步骤..... 247

8.4.2 基于 EJB 访问但不含数据库访问的 Java 示例项目 248

8.5 使用 EJB 开发的会员管理系统..... 251

8.5.1 项目总体设计 251

8.5.2 项目若干技术要点与详细设计..... 253

习题 8 262

实验 8 使用 EJB 实现企业级分布式应用 263

第 9 章 使用 Maven 管理 Java/Web 项目 265

9.1 Maven 概述 265

9.1.1 项目对象模型 POM..... 265

9.1.2 本地仓库、远程仓库与中央仓库..... 267

9.2 Maven 项目开发基础 268

9.2.1 Maven 3 开发环境搭建 268

9.2.2 在 MyEclipse 中新建项目时应用 Maven 支持 269

9.2.3 在 MyEclipse 中新建 Maven 项目 271

9.3 Maven 项目单元测试、发布和导入..... 272

9.3.1 Maven 单元测试 272

9.3.2 Maven Web 项目发布 273

9.3.3 Maven 项目导入 273

习题 9 275

实验 9 使用 Maven 管理 Java/Web 项目 276

习题答案 277

参考文献 282



Web 应用开发基础

计算机的应用经历了从桌面型(指安装在本机上运行的桌面软件,即单机版本)到多用户型(指一台主机带若干终端,即多用户版本)再到 Web 型(指采用 B/S 体系的网站系统),最终延伸到以手机客户端为代表的移动平台系统。Web 应用使人们超越了时间、地理位置的限制,可以方便地处理各种各样的信息。

作为 Web 应用开发的基础,本章主要介绍了 B/S 体系的含义、搭建 Java Web 应用开发环境和使用 JDBC 方式访问 MySQL 数据库;此外,还介绍了 Java 单元测试和动态调试方法。学习要点如下:

- 理解 Web 应用与传统的桌面应用方式的不同;
- 掌握 Java Web 服务器的运行环境;
- 掌握 MyEclipse 开发环境的使用;
- 掌握使用 JDBC 访问 MySQL 数据库的方法;
- 掌握 Java 单元测试和动态调试的使用方法。

1.1 网站与网页基础

1.1.1 Web 应用体系与 B/S 模式

在 Internet 网站中,存放着许多服务器,最重要的服务器是 Web 服务器,客户端通过浏览器等软件来访问 Web 服务器里的网站。

访问网站,最终是对网站里网页的访问。通过访问网页,人们能够查询所需要的信息,也能提交信息并将其保存在数据库服务器里。

网页分为静态网页与动态网页两种。静态网页采用 HTML 的标签语言编写,动态网页除了包含静态的 HTML 代码外,还包含了只能在服务器端解析的服务器代码。动态网页是与静态网页相对应的,通常以.aspx、.jsp、.php 等作为扩展名,而静态网页通常以.html 作为扩展名。

注意:

(1) 动态网页与网页上的各种动画、滚动字幕等视觉上的动态效果没有直接关系,动态网页是采用动态网站技术生成的网页,它可以是纯代码。

(2) 动态网页需要使用某种运行于服务器端的脚本语言编写。脚本分为客户端脚本与服务器端脚本两大类,第 1.1.4 小节将介绍 JavaScript 客户端脚本,第 2 章的 JSP 页面里包含 JSP 服务器脚本。

包含动态网页的网站称为动态网站,其主要特征是服务器能实现与客户端的交互、数据库存储等。

在 B/S 中,客户端使用浏览器等应用程序与 Web 服务器进行通信,并使用超文本传送 HTTP(hypertext transfer protocol)。

网络协议是分层的。其中,HTTP 是建立在 TCP 之上的一种应用层协议,Web 应用中使用 HTTP 作为应用层协议,用以封装 HTTP 文本信息,然后使用传输层协议 TCP/IP 将客户端与 Web 服务器之间的通信信息发到网络上。

注意:在网络应用中,除了 B/S 模式,还有 C/S 模式。C/S 模式的一个典型例子是学校内部的刷校园卡消费系统,刷卡终端与服务器连接的形式采用硬件终端的形式。

1.1.2 常用 HTML 标记及其使用

HTML(hypertext markup language),即超文本标记语言,用于描述 Web 页面的显示格式。在 HTML 中,所有的标记符都是用一对尖括号括起来的,绝大部分标记符是成对出现的,包括开始标记符和结束标记符。开始标记符和相应的结束标记符定义了该标记符作用的范围。结束标记符与开始标记符的区别是结束标记符在<号之后有一个斜杠。例如,定义一个向上滚动的新闻的 HTML 代码为:

```
<marquee width="300" height="280" direction="Up">滚动新闻文本</marquee>
```

除了<marquee>标记外,常用的 HTML 标记如下。

- 超链接标记<a>:用于设计超链接。
- 区隔标记:用于修饰特定的文本。
- 区块标记<div>:具有 float、padding 和 margin 等 CSS 样式属性,这些 CSS 样式属性是不具备的。
- 图像标记:用于引入图像。
- 段落标记<p>:可以对一个段落应用 CSS 样式。
- 换行标记
:起换行作用,单标记名后的斜杠表示自闭。
- 列表标记或:需要配合标记使用。
- 表格标记<table>、<tr>、<td>和<th>:常用于数据显示。
- 面内框架标记<iFrame>:定义页内框架。
- 表单标记<form>:需要内嵌若干<input>标记。

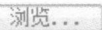
注意:在客户端,页面呈现的过程就是浏览器程序解释 HTML 标记的过程。

表单常用来制作客户端的信息录入界面或登录界面。当用户单击“提交”按钮后,浏览器地址栏将出现一个新的 HTTP 请求,跳转至表单处理页面,接收用户提交的信息并做相应的处理。一个表单定义的示例代码如下:

```
<form name="表单名称" method="post" action="表单处理程序">
..... <!--定义接收用户数据输入的表单元素-->
```

```
<input type="Submit" value="提交">
</form>
```

如果不指定表单的 action 属性值, 则默认由本页面自处理。表单自处理的 JSP 页面, 参见第 2.4.2 小节项目 MemMana1 里的会员登录页面 mLogin.jsp 等。

对于文件上传表单(使用了标记<input type="file" name="wjy"/>, 必须对表单使用属性 enctype="multipart/form-data"。页面浏览时, 会出现选择文件的按钮 , 单击它后出现选择文件对话框。文件上传表单, 参见项目 MemMana4 的后台管理功能。

在表单内, 还可以用命令按钮来响应客户端的单击事件, 其定义方法如下:

```
<input type="button" value=? onClick="客户端脚本方法">
```

注意:

- (1) 表单的 method 属性值一般指定为“post”, 它也是默认值。
- (2) 提交按钮/重置按钮, 只能作为表单里的最后元素。
- (3) 提交按钮通常是表单必需的, 而重置按钮则不然。
- (4) 定义表单元素时, 一般要使用 name 属性, 因为客户端脚本和服务器脚本是按元素名称来获取提交值的。
- (5) 在网站开发实务中, 对表单提交的数据进行有效性验证的方式有两种, 一种方式是定义表单的 onSubmit 事件来实现客户端脚本进行验证(参见案例项目 MemMana1 之会员注册页面 mRegister.jsp), 另一种是在服务器程序中验证。显然, 客户端验证可以减轻 Web 服务器的压力, 值得推荐。

页面框架是指页面里的一块区域, 使用 Div 布局页面时, 可以将某个 Div 定义为页内框架, 其方法是使用成对的 HTML 标记<iFrame>及</iFrame>, 定义格式如下:

```
<div><iFrame src="预载页面" name="框架名" width="" height=""></iFrame></div>
```

其中, src、name、width 和 height 是 iFrame 标记的四个常用属性, 但 src 不是必需的。

页内框架应用于超链接中, 将链接页面的内容输出到指定的页内框架中, 而不是打开一个新窗口, 引用方法如下:

```
<a href="目标页面" target="页内框架名">
```

注意:

- (1) 使用页内框架, 避免频繁打开新窗口, 使浏览过程更加连贯, 从而改善用户体验。
- (2) 使用页内框架的示例, 参见项目 MemManal 的主页 index.jsp。

1.1.3 CSS 样式与 Div 布局

1. CSS 样式技术

CSS 是 1996 年底产生的新技术, 是 cascading style sheet 的缩写, 译名为层叠样式表。CSS 是一组样式, 它并不属于 HTML, 把 CSS 样式应用到不同的 HTML 标记中, 可扩展 HTML 功能, 如调整字间距、行间距、取消超链接的下划线效果、多种链接效果等, 这是

原来的 HTML 标记无法实现的效果。

使用 CSS 技术,除了可以在单独网页中应用一致的格式外,对于大网站的格式设置和维护更具有重要意义。将 CSS 样式定义到样式表文件中,然后在多个网页中同时应用该样式表中的样式,就能确保多个网页具有一致的格式,并且能够随时更新(只需更新样式表文件),从而大大降低网站的开发和维护工作量。

由于 CSS 样式的引入,HTML 新增了<style>和两个标记,对所有产生页面实体元素的 HTML 标记都可以使用属性 style、class 或 id 来应用 CSS 样式。

常用的 CSS 选择器的特性如下。

- 类选择器:在<style>标记内定义时,样式名前缀为“.”,由用户决定哪些对 HTML 标记使用 class 属性来应用该样式。
- ID 选择器:在<style>标记内定义时,样式名前缀为“#”,对 HTML 标记使用 id 属性来应用该样式,且要求应用本 ID 样式的页面元素是唯一的。
- 标签选择器:在<style>标记内,以 HTML 标记作为样式名(无前缀),用来重新定义 HTML 标记的外观(自动应用于相应的 HTML 标记)。
- 伪类选择器:对超链接的不同状态的样式的定义,包括 a:hover(鼠标位于超链接上时)等。

注意:

(1) 当不涉及 JavaScript 脚本(含 jQuery)时, ID 样式与类样式可以互换。ID 选择器的唯一性是指应用 ID 样式的页面元素应当是唯一的。

(2) 对一个页面元素同时应用多种样式时,其选择器名称之间使用空格隔开。

(3) 上面的伪类选择器是复合内容选择器(也称组合选择器)的一种使用形式。

专业的网页设计软件 Dreamweaver(以下简称 DW),提供了样式面板,可以进行 CSS 样式的可视化操作。编辑网页时,使用样式面板可以创建 CSS 样式。在指定 CSS 样式属性(字体大小、颜色和行高)、单击“确定”按钮后,将在页面头部生成成对标记<style>和</style>,其代码如下:

```
<style type="text/css">
.zw {
    font-size: 12px; /*字体大小, 像素为单位*/
    color: #F00; /*颜色为红色*/
    line-height:20 /*文字的行高*/
}
</style>
```

多个选择器之间使用空格分隔,表示需要根据文档的上下文关系(也称父子关系)来确定 HTML 标记应用或者避免的 CSS 样式,即通过 CSS 样式来实现准确定位。例如:

```
.faces .face1 { /*face1 也称后代选择器*/
/*定义类选择器face1的CSS样式属性*/
```

```

}
.menu {
    /*定义类选择器menu*/
}
.menu ul{
    /*定义作为后代的标签选择器ul的CSS样式属性*/
}
.menu ul li{
    /*定义辈分更低的后代的标签选择器li的CSS样式属性*/
}
.menu ul li a{
    /*定义辈分更低的后代的标签选择器a的CSS样式属性*/
}

```

每个 HTML 标记所生成的页面元素，都有其默认的外观。例如，HTML 标记<a>所产生的超链接，在默认情况下，存在下划线。在实际的网站开发时，通常通过重新定义 HTML 标记<a>样式，可以取消默认的下划线，例如：

```

<style type="text/css">
    a {
        text-decoration: none; /*取值none时无下划线，取值为underline时有下划线*/
        font-size: 18px; /*设置链接文字的大小*/
    }
</style>

```

与伪类选择器对应的样式称为伪类样式。例如除了 a:hover 外，还有 a:active(超链接被选中时)、a:visited(超链接被访问时)和 a:link(没有被访问时)都是伪类样式。

内联样式是通过 style 属性把 CSS 样式属性键值对引入到定义对象的 HTML 标记，例如：

```
<span style="font-size: 24px; color: red;">文字</span>
```

CSS 滤镜是 CSS 样式的扩展，它能将特定效果应用于文本容器、图片或其他对象。CSS 滤镜通常作用于 HTML 控件元素，如 img、td 和 div 等。

在 CSS 样式中，通过关键字 filter 引入滤镜。例如，对于空间文字，应用 shadow 滤镜可以实现文字的阴影效果，其 CSS 样式的属性如下：

```
filter:shadow(color=cv,direction=dv)
```

其中：滤镜参数 color 表示阴影的颜色，cv 值可使用代表颜色的英文单词，如 red、blue、green 等，也可以使用色彩代码；参数 direction 表示阴影的方向，dv 取值为 0~360。

注意：不同的浏览器对滤镜的支持是有区别的。例如，Shadow 滤镜只有 IE 浏览器支持，而其他浏览器不支持。

外部样式是指样式的定义在一个单独的文件里，该样式文件以.css 作为扩展名。建立外部样式文件后，可以在网站的每个页面里引用它，用于统一网站风格。

在 DW 中，引用外部样式的一个快捷方法是从站点文件面板拖曳 CSS 样式文件至页面代码窗口里的头部(</title>)，此时会在页面代码窗口里增加如下代码：

```
<link rel="stylesheet" type="text/css" href="带路径的样式文件名.css">
```

2. CSS+Div 布局

在使用 Div 布局的页面里，通常情况下，页面里最外面的那个大 Div，一般需要设置成水平居中，其方法是应用如下的 CSS 样式属性：

```
margin:0 auto; /*margin-right与margin-left属性值为auto*/
```

当 Div 嵌套时，同一级别的多个 Div，其默认位置关系是上下关系，要改变成左右关系，只需要对同一级别的多个 Div 设置如下的 CSS 样式属性：

```
float:left; /*并排多个Div*/
```

Div 常用的 CSS 样式属性如表 1.1.1 所示。

表 1.1.1 Div 常用的 CSS 样式属性

CSS 属性名	功 能 描 述
position	定位属性，常用取值为 absolute、relative，默认值为 static
left 和 top	定义左上角点，适用于 absolute 和 relative 两种定位方式，相对父 Div
right 和 bottom	定义右下角点，适用于 absolute 和 relative 两种定位方式，相对父 Div
width 和 height	定义 Div 的宽度和高度，以像素为单位
text-align	定义 Div 里面内容的对齐方式
border	定义 Div 的边框，以像素为单位
background	定义 Div 背景图片
float	浮动，取值 left 或 right，常用于实现 Div 的并排
margin	外填充，用于设置 Div 之间的间距，可按“上右下左”的顺序分别设置
padding	内填充，用于设置 Div 与其内部元素的间距，也可分别设置
overflow	取值为 hidden 时，隐藏超出 Div 尺寸的内容，不破坏整体布局
z-index	定义层叠加的顺序，取值整数，值越大，就越靠上

Div 除了通过 style 属性应用内联 CSS 样式外，还可以通过 class 属性应用类样式或通过 id 属性应用 ID 样式。