

JSP从零开始学

(视频教学版)

刘 鑫 编著



- 本书由浅入深，由理论到实践，方便JSP入门人员自学并逐步完善自己的知识结构
- JSP开发环境搭建、JSP语法、JSP内置对象、Servlet技术、EL标签、会话管理、Java Bean、JSTL标签库、自定义标签、JDBC使用、XML使用、资源国际化等基础知识和示例，都在这里了
- 综合运用JSP技术开发一个完整的网上购物系统



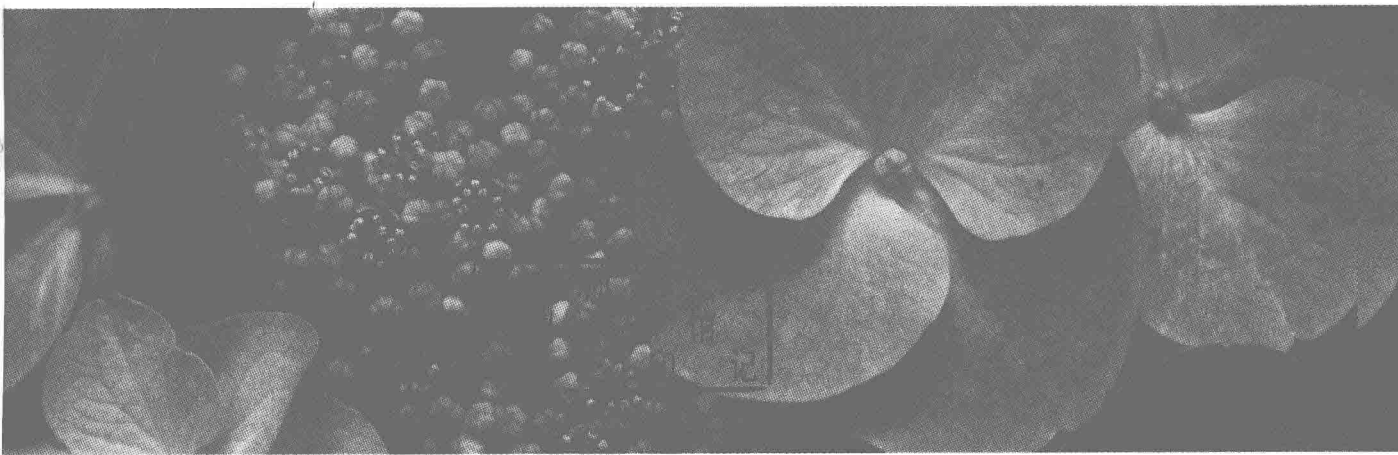
本书示例源代码、课件、教学视频下载

清华大学出版社

JSP从零开始学

(视频教学版)

刘 鑫 编著



清华大学出版社
北 京

内 容 简 介

JSP 依靠强大的 Java 基础,成为世界上最流行的 Web 开发利器。本书通过大量的实例,循序渐进地为读者介绍了有关 JSP 开发所涉及的各类知识,所有版本一律采用最新版本,是目前市场上学习 JSP 技术的首选。

本书分为 14 章,首先介绍网页开发的基础原理,然后搭建 JSP 开发环境,再介绍 JSP 基础、基本语法、内置对象、Servlet、JavaBean、MySQL、XML 文件、资源国际化等,最后通过一个完整的在线购物网站案例,回顾前面所学的 JSP 技术。

本书利用实例贯穿所有的语法,具有很强的操作性,适合 JSP 初学者、Web 开发者和所有前端人员学习。

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 从零开始学:视频教学版 / 刘鑫编著. - 北京:清华大学出版社,2016
ISBN 978-7-302-42188-7

I. ①J… II. ①刘… III. ①JAVA 语言—网页制作工具 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2015)第 272718 号

责任编辑:夏非彼

封面设计:王 翔

责任校对:闫秀华

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

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:190mm×260mm

印 张:27.5

字 数:704 千字

版 次:2016 年 1 月第 1 版

印 次:2016 年 1 月第 1 次印刷

印 数:1~3000

定 价:69.00 元

产品编号:065613-01

前言

互联网时代，Web 无所不在，开发更好的网页和 Web 应用成为很多企业的追求。鉴于使用 Java 开发的用户最多，JSP 也称为最热门的 Web 开发技术。JSP 实现了动态页面与静态页面的分离，脱离了硬件平台的束缚，而且还可以编译后运行，这大大提高了它的执行效率。在网络信息量巨大的今天，使用 JSP 技术提高效率是网站的首选。鉴于学习 JSP 的人数众多，我们特地编写这样一本 JSP 入门书籍，内容安排由浅入深，由理论到实践，适合 JSP 初学者逐步学习和完善自己的知识结构。

为什么选择 JSP

(1) JSP 语言的可扩充性。JSP 技术依靠众所周知的 Java 技术为载体，可以很轻松的将应用平台进行扩展，类似于插板上的插头，只要插孔够用，可以很容易的扩展出很多的电器。

(2) JSP 服务器的高稳定性。JSP 服务器的运行环境主要是 Linux 服务器平台，这种平台在大型应用服务中采用甚多，国内著名的电商平台：淘宝、凡客、京东以及各种 OA 办公自动化平台都采用 Linux 服务器作为应用环境。

(3) 并发性能。JSP 在用户访问量的负载程度上具有很大的优势，在多数用户同时访问的情况下不会出现崩溃、服务器瘫痪的危险。

(4) 安全性。由于 JSP 语言最初便定位于 OA 办公自动化和电子商务平台等大型应用平台，因此对程序本身的安全性做出了极高的要求。

(5) JSP 数据量的吞吐及负载能力。基于 JSP 技术操作的数据库连接在信息访问量上要更加快速、庞大，效率更高、数据的稳定性也更强。

(6) JSP 的高安全性。JSP 技术主要依赖于 Java 语言，所以在信息的安全性上要更加优越，程序的开发更加缜密，大大降低了网站黑客被入侵的概率。

(7) JSP 网站访问的高速性。JSP 技术制作的网站因其代码量更加优化、冗余度低，所以访问时会更加高效，速度会更快。

本书的内容安排

本书共分为 14 章，主要章节规划如下。

第 1 章介绍网页最基本的知识，包括什么是动态、静态网页，网页浏览的原理，网络传输的协议，流行的网页开发语言，JSP 网页的执行顺序等。

第 2 章详细介绍要使用 JSP 必须配置的开发环境，包括 JDK 的安装与配置、Tomcat 的安装与配置，MyEclipse 的安装与配置。最后还介绍如何用 MyEclipse 来开发第一个 JSP 程序。

第 3 章介绍 JSP 最最基础的语法知识，包括 JSP 的注释、声明、表达式、指令、动作等。

第4章介绍JSP的内置对象 request、response、session、application、out、page、config，这些对象实现了网页请求的一些最基本的应用。

第5章是Servlet技术的应用，这是JSP实现更多功能的关键，本章着重介绍了Servlet的生命周期、编写和部署、应用程序事件、监听器、过滤器和异步处理等等。

第6章是EL标签的应用，标签可以让JSP更高效的工作，本章重点介绍了如何使用EL标签和如何禁用EL标签。

第7章是网页的请求、响应与会话管理，这在整个网页浏览周期中是最最关键的地方，一个网页从请求到响应的过程中，我们如何保存数据、传输数据都在本章体现。

第8章介绍了如何使用Java Bean读取数据库，网页信息量这么大，数据的存储和读取非常关键，本章重点是数据库的操作。

第9章介绍了JSTL标签库，这是JSP常用的一个标签库，包括了5大标签种类，本章通过实例的方式详细介绍了这些标签的使用。

第10章教会我们实现自定义标签，如果官方提供的标签不够用，我们还可以使用自己制作的标签。

第11章是使用JDBC连接数据库，JDBC使用非常广泛，也是JSP操作数据库最常用的方式。

第12章介绍了XML文件格式，如今遍地都是XML文件，这类文件如何编写，如何使用，都是本章介绍的重点。

第13章介绍了资源国际化，这是保证页面中文和英文都能更清晰表达内容的关键所在，使用资源国际化，我们可以开发面向全球的网页。

第14章通过一个完整的网上购物系统，详细介绍了使用JSP开发Web网站的整个流程，包括后台类的编写、页面的设计和公共页面的处理。

本书读者与作者

- Java Web 开发初学者
- JSP 初学者
- 从事JSP编程的Web开发人员
- 高校、大中专院校的师生

本书第1~10章由湖南铁道职业技术学院的刘鑫编写，其他参与编写的还有李阳、张学军、陈士领、陈丽、殷龙、张鑫、赵海波、张兴瑜、毛聪、王琳、陈宇、生晖、张喆、王健，排名不分先后。

代码、课件与教学视频下载

本书配套代码、课件与教学视频下载地址（注意数字和字母大小写）如下：

<http://pan.baidu.com/s/1dDJksGp> 密码：3y17

如果下载有问题，请电子邮件联系 booksaga@163.com，邮件主题为“JSP 从零开始学”。

编 者
2015年12月

目 录

第 1 章 网页制作与浏览原理	1
1.1 我们所理解的网页	1
1.1.1 静态网页	2
1.1.2 动态网页	2
1.1.3 浏览器和服务器对应的 B/S 模式	2
1.2 常见的动态网页开发语言	3
1.2.1 JSP	3
1.2.2 PHP	4
1.3 网页的浏览原理	4
1.3.1 Web 是什么	4
1.3.2 HTTP 超文本传输协议是什么	5
1.4 用 JSP 进行网页开发	6
1.4.1 JSP 与其他语言相比的优势	6
1.4.2 JSP 网页的执行顺序	7
1.4.3 实例：第一个 Hello JSP 网页	8
1.5 上机实践	9
第 2 章 搭建 JSP 开发环境	10
2.1 安装并配置 Java 环境	10
2.1.1 下载并安装 JDK	10
2.1.2 配置环境变量	11
2.1.3 测试 Java 配置结果	12
2.2 安装并配置 Tomcat 服务器	13
2.2.1 下载并安装 Tomcat 服务器	13
2.2.2 Tomcat 的文件结构	14
2.2.3 Tomcat 的工作原理	15
2.3 使用 MyEclipse 开发工具	16
2.3.1 下载并安装 MyEclipse	16
2.3.2 在 MyEclipse 中配置 J2EE 环境	18
2.3.3 在 MyEclipse 中配置 Tomcat	19
2.3.4 MyEclipse 使用技巧	20

2.3.5 其他 IDE	25
2.4 实例: 使用 MyEclipse 开发一个完整的 Java Web 网页	25
2.5 第一次运行 JSP 文件的两个常见问题	30
2.6 在 MyEclipse 中导入原来的项目	31
2.7 上机实践	33
第 3 章 JSP 的基础语法	34
3.1 JSP 的注释和声明	34
3.1.1 JSP 中的注释	34
3.1.2 JSP 中的声明	36
3.2 JSP 表达式	38
3.3 JSP 指令	40
3.3.1 与页面属性相关的 page 指令	40
3.3.2 引入文件的 include 指令	41
3.3.3 与标签相关的 taglib 指令	42
3.4 JSP 动作	45
3.4.1 <jsp:include>动作	45
3.4.2 <jsp:forward>动作	48
3.4.3 <jsp:param>动作	49
3.5 上机实践	53
第 4 章 JSP 的内置对象	54
4.1 request 对象	54
4.1.1 request 对象的常用方法	54
4.1.2 使用 request 对象接收请求参数	55
4.1.3 请求中文乱码的处理	57
4.1.4 获取请求的头部信息	58
4.1.5 获取主机和客户机信息	60
4.2 response 对象	62
4.2.1 response 对象的常用方法	62
4.2.2 设置头信息	62
4.2.3 设置页面重定向	65
4.3 session 对象	66
4.3.1 获取 session ID	67
4.3.2 登录用户信息的保存	71
4.4 application 对象	76
4.4.1 application 对象的常用方法	76
4.4.2 获取指定页面的路径	77
4.4.3 设计一个网站计数器	78
4.5 out 对象	79

4.5.1	out 对象的常用方法	79
4.5.2	out 对象的使用示例	79
4.6	page 对象	81
4.6.1	page 对象的常用方法	82
4.6.2	page 对象的使用示例	82
4.7	config 对象	83
4.7.1	config 对象的常用方法	83
4.7.2	config 对象的使用示例	83
4.8	上机实践	85
第 5 章	Servlet 技术的应用	86
5.1	Servlet 是什么	86
5.2	Servlet 的技术特点	87
5.3	Servlet 的生命周期	88
5.4	编写和部署 Servlet	92
5.4.1	编写 Servlet 类	92
5.4.2	部署 Servlet 类	96
5.5	Servlet 与 JSP 的比较	97
5.6	Servlet 进阶 API	98
5.6.1	Servlet、ServletConfig 与 GenericServlet	99
5.6.2	使用 ServletConfig	101
5.6.3	使用 ServletContext	104
5.7	应用程序事件、监听器	107
5.7.1	ServletContext 事件、监听器	107
5.7.2	HttpSession 事件监听器	111
5.7.3	HttpServletRequest 事件、监听器	116
5.8	过滤器	119
5.8.1	过滤器的概念	119
5.8.2	实现与设置过滤器	120
5.8.3	请求封装器	122
5.8.4	响应封装器	125
5.9	异步处理	135
5.9.1	AsyncContext 简介	135
5.9.2	模拟服务器推送	138
5.10	上机实践	143
第 6 章	EL 标签的应用	144
6.1	认识 EL 标签	144
6.1.1	EL 标签的语法	144
6.1.2	EL 标签的功能	145

6.1.3	EL 标签的操作符	149
6.2	EL 标签的隐含变量	151
6.2.1	隐含变量 pageScope、requestScope、sessionScope、applicationScope	151
6.2.2	隐含变量 param、paramValues	151
6.2.3	其他变量	153
6.3	禁用 EL 标签	154
6.3.1	在整个 Web 应用中禁用	154
6.3.2	在单个页面中禁用	155
6.3.3	在页面中禁用个别表达式	155
6.4	上机实践	155
第 7 章	网页的请求、响应与会话管理	156
7.1	从容器到 HttpServlet	156
7.1.1	Web 容器做了什么	156
7.1.2	doXXX()方法有什么用	158
7.2	HttpServletRequest 对象的应用	159
7.2.1	使用 getReader()、getInputStream()读取 Body 内容	159
7.2.2	使用 getPart()、getParts()取得上传文件	164
7.2.3	使用 RequestDispatcher 调派请求	168
7.3	HttpServletResponse 对象的应用	174
7.3.1	使用 getWriter()输出字符	174
7.3.2	使用 getOutputStream()输出二进制字符	177
7.3.3	使用 sendRedirect()、sendError()方法	179
7.4	会话管理基本原理	182
7.4.1	使用隐藏域	182
7.4.2	使用 Cookie	182
7.4.3	使用 URL 重写	183
7.5	HttpSession 会话管理的应用	184
7.5.1	使用 HttpSession 管理会话	184
7.5.2	HttpSession 管理会话的原理	187
7.5.3	HttpSession 与 URL 重写	187
7.5.4	HttpSession 中禁用 Cookie	188
7.5.5	HttpSession 的生命周期	188
7.5.6	HttpSession 的有效期限	189
7.6	实例: 用 Servlet 实现网站的注册和登录	190
7.6.1	实现网站注册功能	190
7.6.2	实现网站登录功能	196
7.7	实例: 使用 HttpSession 实现猜字游戏	199
7.8	上机实践	201

第 8 章 使用 Java Bean 读取数据库	202
8.1 MySQL 数据库入门	202
8.1.1 MySQL 的安装和配置	202
8.1.2 启动 MySQL 服务	211
8.1.3 登录 MySQL 数据库	213
8.2 MySQL 数据库的基本操作	215
8.2.1 创建数据库	215
8.2.2 删除数据库	217
8.2.3 创建数据库表	217
8.2.4 修改数据库表	218
8.2.5 修改数据库表字段名	219
8.2.6 删除数据库表	220
8.3 MySQL 数据库的数据管理	220
8.3.1 插入数据	221
8.3.2 修改数据	221
8.3.3 删除数据	222
8.4 Java Bean 的使用	223
8.4.1 认识 Java Bean	223
8.4.2 在 JSP 中使用 Bean	224
8.4.3 访问 Bean 属性	226
8.4.4 Bean 的作用域	233
8.5 实例: 利用 Java Bean 实现用户登录验证	239
8.6 DAO 设计模式	244
8.6.1 DAO 设计模式简介	244
8.6.2 DAO 命名规则	245
8.6.3 DAO 开发	245
8.6.4 JSP 调用 DAO	252
8.7 上机实践	255
第 9 章 JSTL 标签库	257
9.1 JSTL 标签概述	257
9.1.1 JSTL 的来历	257
9.1.2 一个标签实例带你入门	258
9.2 JSTL 的 core 标签库	259
9.2.1 <c:set> 标签、<c:out> 标签	259
9.2.2 <c:if> 标签	259
9.2.3 <c:choose>、<c:when>、<c:otherwise> 标签	261
9.2.4 <c:set> 标签	262
9.2.5 <c:forEach> 标签	262
9.2.6 <c:forTokens> 标签	264

9.2.7	<c:remove>标签	264
9.2.8	<c:catch>标签	265
9.2.9	<c:import>标签与<c:param>标签	265
9.2.10	<c:redirect>标签	265
9.2.11	<c:url>标签	266
9.3	JSTL 的 fmt 标签库	266
9.3.1	<fmt:requestEncoding>设置编码	267
9.3.2	<fmt:setLocale>显示所有地区的数据格式	267
9.3.3	<fmt:bundle>、<fmt:message>、<fmt:param>资源国际化	267
9.3.4	<fmt:setBundle>标签	269
9.3.5	<fmt:formatNumber>显示不同地区的各种数据格式	270
9.3.6	<fmt:parseNumber>解析数字	270
9.3.7	<fmt:formatDate>格式化日期	271
9.3.8	<fmt:parseDate>解析日期	272
9.3.9	<fmt:setTimeZone>标签和<fmt:timeZone>标签	272
9.4	JSTL 的 fn 方法库	273
9.4.1	fn:contains()函数与 fn:containsIgnoreCase()函数	273
9.4.2	fn:startsWith()函数与 fn:endsWith()函数	273
9.4.3	fn:escapeXml()实现 HTML 编码	273
9.4.4	fn:indexOf()函数与 fn:length()函数	274
9.4.5	fn:split()函数与 fn:join()函数	275
9.5	JSTL 的 SQL 标签库	275
9.5.1	<sql:setDataSource>标签	276
9.5.2	<sql:query>标签	276
9.5.3	<sql:update>标签	277
9.5.4	<sql:dateParam>标签与<sql:param>标签	277
9.5.5	<sql:transaction>标签事务管理	280
9.6	JSTL 的 XML 标签库	280
9.6.1	<x:parse>获取新浪 RSS 新闻	281
9.6.2	<x:out>输出指定元素	282
9.6.3	<x:forEach>遍历新浪 RSS 新闻	282
9.6.4	<x:if>标签	282
9.6.5	<x:choose>、<x:when>、<x:otherwise>标签	283
9.6.6	<x:set>标签	283
9.6.7	<x:transform>转化 XML 为 HTML	283
9.7	上机实践	284
第 10 章	实现自定义标签	285
10.1	编写自定义标签	285
10.1.1	版权标签	285

10.1.2	tid 标签库描述文件	287
10.1.3	TagSupport 类简介	289
10.1.4	带参数的自定义标签	290
10.1.5	带标签体的自定义标签	293
10.1.6	多次执行的循环标签	297
10.1.7	带动态属性的自定义标签	299
10.2	嵌套的自定义标签	300
10.2.1	实例: 表格标签	300
10.2.2	嵌套标签的配置	302
10.2.3	嵌套标签的运行效果	303
10.3	JSP 2.x 标签	305
10.4	上机实践	307
第 11 章	使用 JDBC 连接数据库	308
11.1	JDBC 简介	308
11.1.1	查询实例: 列出人员信息	308
11.1.2	各种数据库的连接	311
11.2	MySQL 的乱码解决	312
11.2.1	MySQL 的乱码解决	312
11.2.2	从控制台修改编码	313
11.2.3	从配置文件修改编码	314
11.2.4	利用图形界面工具修改	314
11.2.5	URL 中指定编码方式	315
11.3	JDBC 基本操作: CRUD	315
11.3.1	查询数据库	315
11.3.2	插入人员信息	316
11.3.3	注册数据库驱动	321
11.3.4	获取自动插入的 ID	321
11.3.5	删除人员信息	322
11.3.6	修改人员信息	323
11.3.7	使用 PreparedStatement	329
11.3.8	Statement 与 PreparedStatement 批处理 SQL	331
11.4	处理结果集	332
11.4.1	查询多个结果集	332
11.4.2	可以滚动的结果集	333
11.4.3	带条件的查询	333
11.4.4	ResultSetMetaData 元数据	338
11.4.5	直接显示中文列名	340
11.5	上机实践	340

第 12 章 XML 文件格式.....	341
12.1 初识 XML.....	341
12.1.1 什么是 XML	341
12.1.2 XML 的用途	342
12.1.3 XML 的技术架构	343
12.1.4 XML 开发工具.....	343
12.2 XML 基本语法.....	344
12.2.1 XML 文档的基本结构.....	344
12.2.2 标记必须闭合	345
12.2.3 必须合理地嵌套	345
12.2.4 XML 元素.....	345
12.2.5 XML 属性.....	346
12.2.6 只有一个根元素	346
12.2.7 大小写敏感	347
12.2.8 空白被保留	347
12.2.9 注释的写法	347
12.2.10 转义字符的使用	347
12.2.11 CDATA 的使用	348
12.3 JDK 中的 XML API	348
12.4 最常见的 XML 解析模型	349
12.4.1 DOM 解析.....	349
12.4.2 SAX 解析.....	352
12.4.3 DOM4j 解析	355
12.5 XML 与 Java 类映射 JAXB	357
12.5.1 什么是 XML 与 Java 类映射.....	357
12.5.2 JAXB 的工作原理	358
12.5.3 Java 对象转化成 XML.....	359
12.5.4 XML 转化为 Java 对象.....	360
12.5.5 更为复杂的映射	362
12.6 上机实践.....	365
第 13 章 资源国际化	367
13.1 资源国际化简介	367
13.1.1 国际化编程 I18N.....	367
13.1.2 本地化编程 L10N.....	367
13.2 资源国际化编程	368
13.2.1 资源国际化示例	368
13.2.2 资源文件编码.....	369
13.2.3 显示所有 Locale 代码	370
13.2.4 带参数的资源.....	372

13.2.5 ResourceBundle 类	373
13.2.6 Servlet 的资源国际化	375
13.2.7 显示所有 Locale 的数字格式	377
13.2.8 显示全球时间	378
13.3 上机实践	380
第 14 章 简易的网上购物系统	381
14.1 系统需求分析	381
14.2 系统总体架构	382
14.3 数据库设计	383
14.3.1 E-R 图	383
14.3.2 数据物理模型	383
14.4 系统详细设计	384
14.4.1 系统包的介绍	385
14.4.2 系统的关键技术	385
14.4.3 过滤器	392
14.5 系统首页与公共页面	393
14.6 用户登录模块	395
14.7 用户管理模块	397
14.7.1 用户注册	397
14.7.2 修改用户信息	400
14.7.3 查看用户信息	403
14.7.4 修改用户密码	404
14.8 购物车模块	406
14.8.1 添加购物车	406
14.8.2 删除购物车	410
14.8.3 查看购物车	410
14.8.4 修改购物车	412
14.8.5 删除购物车所有商品	413
14.8.6 购物车中的页面	414
14.9 商品模块	416
14.9.1 查看商品列表	416
14.9.2 查看单个商品	420
14.10 支付模块	421
14.10.1 支付商品	421
14.10.2 查看已支付商品	422
14.10.3 查看已支付商品页面	423
14.10.4 支付中的页面	423
14.11 实战总结	426

第 1 章

◀ 网页制作与浏览原理 ▶

为人们打开外部世界的是互联网，而人们能看得见的是存在于互联网中的无数个网页，这些网页虽然变幻多端，但无论怎么变化，它必须遵循一定的 Web 浏览规则，如本章介绍的 HTTP 协议。本章的目的是让读者了解什么是网页，网页的发展，网页的浏览原理，以及网页开发中使用的脚本语言。

1.1 我们所理解的网页

网页是读者上网浏览时看到的页面，其通过浏览器，呈现在电脑上。用户在浏览器的地址栏，输入一个网站地址，如 www.baidu.com，则打开一个页面，这个页面就是我们常说的“网页”。网站是一个具有多个网页的站点，如 www.baidu.com 就是一个网站的地址，用户通过这个地址来访问网站，网站包含多个相关的网页。图 1.1 是打开的百度网站的一个网页，也是百度网站的主页。

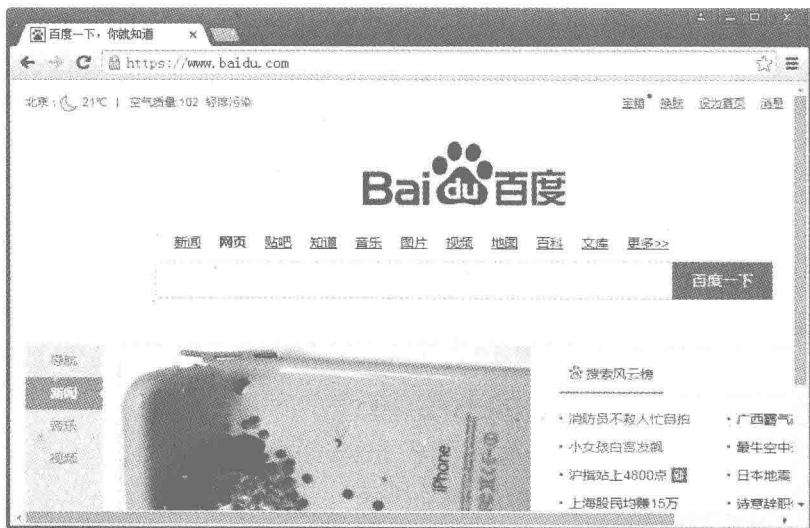


图 1.1 百度网站的主页

网页分为静态网页和动态网页，本节将具体介绍这两种网页。

1.1.1 静态网页

静态页面是指网络上内容和外观总是保持不变的页面。这些页面的文件名后缀通常为.htm或者.html。这些网页的制作最为简单，由 HTML（超文本标记语言）实现，适合表现相对固定的内容，如网站的联系方式、公司简介等等。下面通过编写一个名为 hello.html 的静态页面，帮助读者理解静态页面的工作方式。

新建一个文本文件，手动输入如下代码，然后保存为 hello.html。

```
01 <html>
02 <!--标题-->
03 <head><title>Welcome!</title></head>
04 <!--页面主体-->
05 <body>
06     Hello, 静态网页!
07 </body>
08 </html>
```

提示

由于某些系统设置为不显示文件扩展名，所以有时文件的名称显示为“hello.html”，但其实并不是.html类型的文件，而是“hello.html.txt”文件，所以需要设置系统显示扩展名。

对于该页面，用户可以双击打开，不管用户何时何地以怎样的方式访问，页面的内容都不会再改变，这就是静态页面，也就是说，页面的内容是固定的。

1.1.2 动态网页

静态网页有很多问题，最明显的就是无法与服务器进行交互，用户无法从服务器获取信息并自动更新，也无法将用户的信息提交到服务器。这就出现了动态网页，动态网页一般由两部分组成：静态页面和动态操作。通过 HTML 来显示页面，然后通过动态操作完成信息的更新。

动态网页的实现依靠浏览器端和服务器端的互动。服务器端可以实时处理浏览器端的请求，然后将响应的内容传给浏览器（这些内容可能来自数据库，每次请求的内容都不同）。这样，动态页面就显示在浏览器中了。

由于本书主要就是讲解动态网页，所以这里不再举例，读者可以看本章最后的 JSP 示例。

1.1.3 浏览器和服务器对应的 B/S 模式

B/S 模式（Browser/Server）也就是通过浏览器来访问服务器。用户可以在互联网的任何一个角落，甚至可以是无线终端（PDA 等）。B/S 模式所用的业务逻辑及数据支持都是在服务器上，当用户通过页面提出请求时，服务器及时响应，并把运行后的数据及时送回。B/S 模型图如图 1-2 所示。

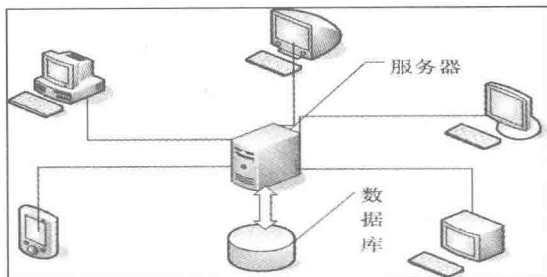


图 1.2 B/S 模型图

B/S 模型的优势:

- (1) 易于维护。基于 B/S 模型的系统当需要升级或维护时，只需修改服务器程序即可。
- (2) 易于实现。B/S 模型的表现层可以用制作网页的 HTML 来实现，浏览器和网页设计技术已经相当成熟。而且用 Java 技术开发的 Web 系统可以安装在任意一种服务器系统平台上，也就是常说的具有跨平台性。
- (3) 使用方便。无论客户在什么地方，只要服务器正在运行，客户就能通过网络进行连接，实现对数据的访问和操作。

B/S 模型的不利因素：增加了服务器的压力。把业务实现都放在服务器上，当有大量的用户访问时，势必会给服务器带来很大的负担，更有可能造成系统崩溃，所以一定要做好系统数据的备份。

1.2 常见的动态网页开发语言

目前最常见的动态网页开发语言有 JSP 和 PHP，关于谁是最好的语言一直争论不休，笔者建议还是根据项目的安排或性能来做决定，每种语言都有自己独特的优势。

1.2.1 JSP

JSP 技术是由 SUN 公司（现被 Oracle 收购）提出，多家公司参与的，于 1999 年推出的一款建设动态网页的方法。它基于 Java Servlet 技术来开发动态的、高性能的 Web 应用程序。JSP 的网页实际上是在 HTML 文件中加入 Java 代码片段和 JSP 特殊的标记构成的。

因为 JSP 是 Java 的成员，所以 JSP 具有平台无关性即实现跨平台功能，实现了用户界面和程序代码的解耦合，使得业务逻辑和代码的耦合度更低，开发人员可以在不更改 JSP 程序下修改用户的界面。

JSP 页面实质也是个 HTML 页面，只不过它包含了用于产生动态网页内容的 Java 代码，这些 Java 代码可以是 Java Bean、SQL 语句、RMI（远程方法调用）对象等。例如：一个 JSP 页面包含了用于产生静态网页的 HTML 代码，同时也包含了连接数据库的 JDBC 代码，那么