



高等学校计算机专业
“十三五”规划教材

Java Web 程序开发与分析

柴慧敏 编著 ■

COMPUTER

COMPUTER

COMPUTER

COMPUTER

COMPUTER



西安电子科技大学出版社
<http://www.xduph.com>

高等学校计算机专业“十三五”规划教材

Java Web 程序开发与分析

柴慧敏 编著

西安电子科技大学出版社

内 容 简 介

本书由浅入深、循序渐进地分析和讲解了 Java Web 开发的主要技术, 内容包括开发环境的搭建和配置、HTML5、JavaScript 编程技术、JSP 基本编程、JavaBean 应用技术、Servlet 技术基础、JDBC 接口与数据应用以及 Struts2 开发框架等。在每一部分内容中, 都给出了多个应用实例, 以分析 Java Web 开发技术的具体应用过程, 使得读者能够透彻理解。

本书内容全面、重点突出, 凡具有计算机程序设计基础的读者均可阅读。

本书可作为计算机专业本科生的教材或教学参考书, 也可作为 Web 工程开发技术人员的参考书。

图书在版编目(CIP)数据

Java Web 程序开发与分析/柴慧敏编著. —西安: 西安电子科技大学出版社, 2015.11

高等学校计算机专业“十三五”规划教材

ISBN 978-7-5606-3873-7

I. ① J… II. ① 柴… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2015)第 256215 号

策 划 戚文艳

责任编辑 阎 彬 董柏娴

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2015 年 11 月第 1 版 2015 年 11 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 20

字 数 475 千字

印 数 1~3000 册

定 价 36.00 元

ISBN 978-7-5606-3873-7/TP

XDUP4165001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

前 言

随着互联网技术的应用和发展，Web 系统开发的重要性日趋明显，越来越多的应用都是基于 Web 发展起来的，大到企业门户网站，小到网页游戏。Java Web 开发在整个 Web 开发领域一直占据着首要位置，许多的 Web 应用都将 Java Web 开发技术作为首选。目前，越来越多的企业开始注重 Java Web 应用人才的吸收和培养。

本书在编写过程中，注重基本概念和技术的介绍，并结合应用实例较深入地分析主要技术的本质和特点，充分体现了 Java Web 开发技术的应用与理论相结合的特点，使得读者能够准确、系统地掌握基本概念和核心技术。

全书共分 9 章。第 1 章为 Java Web 程序概述，主要介绍了 Web 程序的体系结构和运行机制；第 2 章介绍了 Java Web 开发和运行环境；第 3 章在 HTML4 的基础上介绍了 HTML5 的新标签；第 4 章介绍了 JavaScript 编程技术，并讨论了文档对象模型 DOM 和主要接口的使用方法；第 5 章为 JSP 基本编程，主要介绍了 JSP 的运行机制、JSP 的基本语法和 JSP 主要内置对象的使用；第 6 章讨论了 JavaBean 应用技术，主要介绍了 JavaBean 的使用过程，并深入分析了 JavaBean 的生命周期问题；第 7 章为 Servlet 技术基础，主要分析了 Servlet 实现的相关接口、Servlet 过滤器和 Servlet 监听器的实现与应用；第 8 章介绍了 JDBC 接口与数据库应用，包括 JDBC 连接池和 JDBC 如何存取二进制数据等内容；第 9 章为 Struts 开发框架，主要分析了 MVC 开发模式、Struts 开发框架的配置和应用。

本书的编写得到了西安电子科技大学教材基金资助。

本书在编写过程中得到了西安电子科技大学计算机学院领导和老师的支持和帮助，2013 级硕士研究生谭格帆为本书的出版做了大量工作，2012 级硕士研究生胡韵和周坤也做了一些辅助工作，在此对他们表示诚挚的感谢！本书内容参阅了大量的研究资料，这里谨向书中已列出和未列出的所有文献资料的作者表示敬意。

由于作者水平有限，书中难免存在一些缺点，殷切希望广大读者批评指正。

作 者

2015 年 4 月

目 录

第1章 Java Web 程序概述	1
1.1 Web 程序运行机制	1
1.2 网站的静态与动态	5
1.2.1 静态网页与应用	5
1.2.2 用户与浏览器的动态交互	7
1.2.3 开发动态网站的主流技术	8
1.3 Java Web 程序开发技术与体系	10
1.3.1 B/S 结构编程体系	10
1.3.2 Java EE 架构技术	11
1.3.3 Java 开发 Web 程序的知识体系	14
本章小结	15
第2章 Java Web 开发和运行环境	16
2.1 开发工具的选择	16
2.1.1 浏览器	16
2.1.2 Web 服务器	16
2.1.3 数据库管理系统	19
2.2 Eclipse 集成开发环境配置	20
本章小结	22
第3章 HTML5	23
3.1 HTML 语言简介	23
3.1.1 什么是 HTML	23
3.1.2 HTML 版本发展	24
3.1.3 HTML5 特点	24
3.2 HTML 基本语法与常用标签	25
3.2.1 HTML 文档结构	25
3.2.2 格式化文本标签	27
3.2.3 超链接和锚	30
3.2.4 区域布局与定位	32
3.2.5 创建表格与格式设置	36
3.2.6 创建表单格式	38
3.3 嵌入音频和视频	41
3.3.1 HTML 多媒体基础知识	41
3.3.2 网页中嵌入视频	43

3.3.3 网页中集成音频	43
3.4 层叠样式表(CSS).....	44
3.4.1 理解层叠样式表	44
3.4.2 创建样式表规则	45
3.4.3 使用样式表改变页面格式.....	51
本章小结	53
 第 4 章 JavaScript 编程技术.....	54
4.1 JavaScript 简介.....	54
4.2 JavaScript 基本编程.....	56
4.2.1 基本语句	57
4.2.2 使用变量、字符串和数组.....	60
4.2.3 条件和循环控制流程	65
4.2.4 使用函数	67
4.2.5 面向对象编程	73
4.2.6 事件处理程序	81
4.3 文档对象模型(DOM)	86
4.3.1 DOM 简介	86
4.3.2 理解 DOM 模型结构	86
4.3.3 使用 window 对象.....	89
4.3.4 获取和处理 Web 文档	91
4.3.5 DOM 节点	92
4.3.6 创建可定位元素	95
本章小结	98
 第 5 章 JSP 基本编程.....	99
5.1 JSP 概述	99
5.1.1 什么是 JSP	99
5.1.2 JSP 运行机制	100
5.2 JSP 基本语法	102
5.2.1 JSP 文档结构	102
5.2.2 JSP 脚本元素	106
5.2.3 JSP 指令元素	110
5.2.4 JSP 动作元素	117
5.2.5 JSTL 标准标签库.....	127
5.2.6 自定义标签库	132
5.3 JSP 主要内置对象	136
5.3.1 request 请求对象.....	138
5.3.2 response 响应对象	141

5.3.3 session 会话对象.....	143
5.3.4 application 应用服务对象	147
5.3.5 out 对象	149
5.3.6 其他内置对象	150
本章小结	151
第 6 章 JavaBean 应用技术	152
6.1 什么是 JavaBean.....	152
6.2 JavaBean 的特征与创建	153
6.3 使用 JavaBean.....	154
6.3.1 调用 JavaBean.....	154
6.3.2 访问 JavaBean 属性	155
6.3.3 设置 JavaBean 属性	156
6.4 JavaBean 的生命周期	159
6.5 使用 JavaBean 处理表单数据	164
本章小结	167
第 7 章 Servlet 技术基础	168
7.1 Servlet 简介	168
7.2 Servlet 的创建与部署	170
7.3 Servlet 实现相关接口	177
7.3.1 GenericServlet	177
7.3.2 HttpServlet.....	181
7.4 Servlet 请求和响应相关接口	184
7.4.1 HttpServletRequest 接口	185
7.4.2 HttpServletResponse 接口	197
7.5 其 他 接 口	202
7.5.1 ServletConfig 配置接口	202
7.5.2 HttpSession 会话接口	205
7.5.3 ServletContext 上下文接口	213
7.6 Servlet 过滤器	214
7.6.1 过滤器简介	214
7.6.2 过滤器相关接口	216
7.6.3 过滤器类的实现与部署	222
7.7 Servlet 监听器	224
7.7.1 监听器简介	224
7.7.2 监听器相关接口	224
7.7.3 监听器的应用实现	228
本章小结	231

第 8 章 JDBC 接口与数据库应用.....	232
8.1 JDBC 概述.....	232
8.1.1 什么是 JDBC.....	232
8.1.2 JDBC 体系结构.....	232
8.1.3 驱动程序类型	233
8.2 JDBC 连接数据库.....	235
8.2.1 使用 DriverManager 加载驱动.....	235
8.2.2 使用 DataSource 加载驱动	242
8.3 JDBC 操作数据库接口.....	244
8.3.1 Statement 接口	245
8.3.2 PreparedStatement 接口	250
8.3.3 CallableStatement 接口	255
8.3.4 ResultSet 结果集接口	258
8.4 JDBC 连接池.....	261
8.5 使用 JDBC 存取二进制数据.....	268
8.5.1 图像文件存取	268
8.5.2 声音和视频文件存取	271
本章小结	273
第 9 章 Struts 开发框架	274
9.1 Java Web 程序开发模式	274
9.1.1 非 MVC 模式 (Model1 模式)	275
9.1.2 MVC 模式(Model2 模式)	275
9.2 Struts 开发框架简介	277
9.2.1 Struts 框架基本概念	278
9.2.2 Struts2 基本配置	285
9.2.3 Action 的实现.....	292
9.3 Struts2 类型转换	296
9.3.1 类型转换简介	296
9.3.2 基本数据类型转换	298
9.3.3 数组和集合类型转换	301
9.3.4 自定义类型转换	308
本章小结	311
参考文献	312

第 1 章 Java Web 程序概述

1.1 Web 程序运行机制

从 1991 年万维网 WWW(World Wide Web)出现至今,已经二十多年了。在这二十多年间,世界因为网络发生了翻天覆地的变化,从搜索引擎、社交网络、电子商务,到大数据和云计算,网络已经成为人们生活中最重要的基础设施了。

万维网和互联网(Internet)并不是同一个概念。互联网诞生于 20 世纪 60 年代,那时使用网络的人群基本都是专家和技术人员,他们在 UNIX 的命令行下,通过各种不同的协议进行着数据的共享和信息的传递。而万维网通过 HTTP(Hypertext Transfer Protocol,超文本传输协议),在浏览器中显示由文本和各种资源构成的页面,页面由 URL(统一资源定位符)唯一标识。这种可视化的形式使得网络的使用变得非常便捷,即使是不了解计算机网络知识的人们也可以方便地使用网络了,所以现在当人们使用网络时,大部分情况下都是在使用万维网中的 Web 服务。

Web 技术的发展至今已经有十几年的历史了,它的出现大大改变了人们的生活方式。通过 Web 应用系统,用户可以很方便地进行信息浏览、聊天、游戏、网上购物等。

在开发 Web 程序之前,首先需要清楚 Web 程序的运行机制,其运行机制如图 1.1 所示。

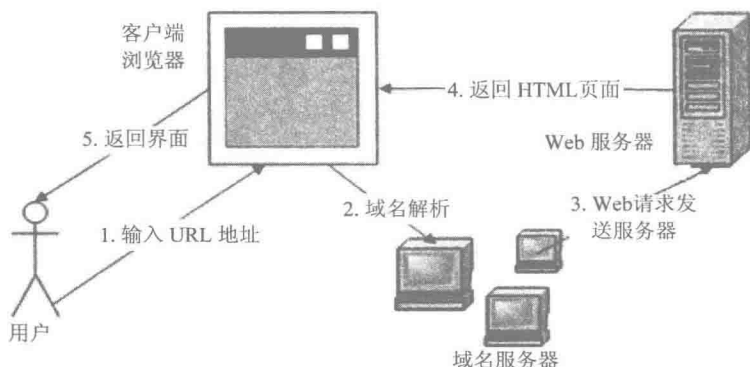


图 1.1 Web 程序运行机制示意图

从图 1.1 中可以分析 Web 程序的运行机制和过程,主要包括以下几点:

- (1) 用户在客户端浏览器中输入 URL 地址,通过浏览器发出 Web 请求;
- (2) 域名服务器对 URL 地址进行解析,并和 Web 服务器建立连接;

(3) Web 请求数据被浏览器封装成 HTTP 报文，并传送到 Web 服务器；

(4) Web 服务器获取 Web 请求后，根据请求信息，执行相关服务器代码(可能会访问后台服务器，如数据库服务器)，并将执行结果以 HTML 页面的方式返回给客户端；

(5) 客户端将 HTML 页面代码在浏览器中解释执行。

在 Web 程序运行中，浏览器承载着向服务器发送请求的功能，在接收到服务器返回的数据后，进行解析并显示。浏览器程序可以运行在个人电脑、智能手机等终端设备上，不需要借助其他软件，即可通过网络与服务器通信。浏览器软件较多，常见的有 Google Chrome、IE 和 Firefox 等。

Web 服务器接收并解析来自浏览器的请求，做出处理后返回响应信息。Web 服务器在运行中监听服务端口，如 HTTP 协议 80 端口，客户端向该端口发送的 HTTP 请求会被服务器解析。若是一个 HTML 文件地址，服务器会将这个文件返回；如果需要运行一个程序，则运行程序并返回运行结果。常见的 Web 服务器软件有 Apache、Microsoft IIS、Nginx 等。

在 Web 程序的运行过程中，涉及一些网络协议或标准，主要包括 HTTP 协议、URL、FTP 等。

1. HTTP

HTTP 是一种无状态的网络协议，默认端口为 80。无状态指的是每一次连接完成后，服务器无法识别上次进行连接的是哪个客户端的什么程序。HTTP 协议传递的报文包括请求报文和响应报文。报文的格式由三部分组成：进行描述的起始行、包含属性的首部块、可选的包含数据的主体。

1) HTTP 请求报文

HTTP 请求报文的格式如下：

```
HTTP 方法 请求的 URL HTTP 版本
若干请求首部
空行
请求体
```

一个典型的 HTTP 请求，如：

```
GET /index.jsp?param=void HTTP/1.1
Host : www.baidu.com
Accept : text/*
```

在这个请求中，起始行是一个请求行，这个请求行表示出该请求是以 GET 方式发出请求的，请求的文件是 Web 文档路径下的 index.jsp 文件，同时传递的还有一个名为“param”，值为“void”的参数，而请求协议的版本是 HTTP/1.1。接下来，后面为名为 Host 和 Accept 的请求首部。Host 首部指出接收请求的服务端地址和端口信息。Accept 首部指出客户端可以接收的 MIME 文件格式，这里使用了通配符“*”，表示发出请求的浏览器可以接收任何文档文件。在空行之后并没有请求体，因为请求体本身是可选的。

HTTP 的第一个版本是 0.9，之后是 HTTP 1.0。目前，HTTP 1.0 已被 HTTP 1.1 取代。HTTP 1.1 与之前的版本相比，将原本的默认连接方式由非持久性连接(短连接)更改为持久

性连接(长连接)。非持久性连接指在一次 HTTP 事件之后,随着 HTTP 响应的结束,浏览器和 Web 服务器之间的连接断开,所以 HTTP 每次只产生一次请求和响应。持久性连接则是在一次 HTTP 响应结束后,服务器和浏览器的连接仍然保持打开状态,直到客户端和服务端决定将其关闭为止。因此默认情况下,客户端可以多次向服务器发出请求。而当需要关闭该 HTTP 请求时,某一方需要向另一方发送 Connection: close 首部(可以是请求报文,也可以是响应报文)。

在 HTTP 1.1 之前,也可以使用 Connection: Keep-Alive 保持连接,如图 1.2 所示。因为服务器能同时提供的连接是有限的,所以对不需要持续的连接,应及时关闭。

HTTP 协议中定义了若干种请求方法,表 1.1 列出了几种较常用的方法。

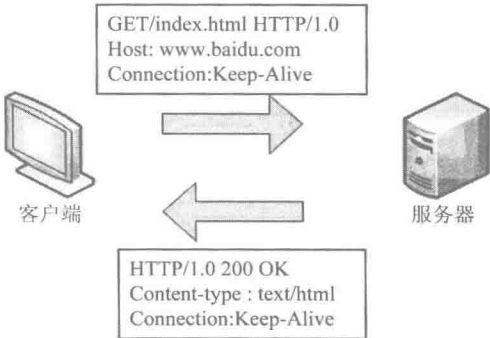


图 1.2 HTTP 协议报文

表 1.1 HTTP 常用请求方法

方 法	功 能
GET	返回指定文档的内容
POST	利用附带的数据执行指定的文档
PUT	利用附带的数据替换指定的文档
DELETE	删除指定文档
HEAD	返回指定文档的头信息

HTTP 中最常用的是 GET 方法和 POST 方法。GET 方法将参数形成字符串,然后附加在 URL 之后传输,参数串将作为 URL 的一部分出现在 HTTP 请求的第一行。其中,参数字符串以“?”开头,参数以“key=value”的“键值对”的方式显示,各“键值对”之间以“&”字符连接。大部分浏览器对 URL 的长度有限制,GET 方式虽然简单可行但当参数过多时,可能超出浏览器能发送的最大 URL。另外,某些不正常的输入行为,比如输入标记对“<”或“>”,可能会引发某些应用的漏洞,或者成为恶意站点脚本攻击的工具。

POST 方法先将 URL 直接发出,消息头中不再含有参数,参数形成的字符串将被放在 HTTP 消息体中发送。URL 中不再显式出现参数,所以参数传递没有长度限制。若发送 HTML 表单数据,或者出于一定的安全性考虑,应当使用这种方式。

比较 GET 和 POST 方法,除了由于参数附加在 URL 中导致的 URL 长度和安全性问题外,浏览器对 GET 方法会有更多的“记忆性”。例如,在浏览器中加入书签时,有时需要根据 GET 方法附加在地址上的参数,来获取同一地址下的不同文章页面;有些浏览器会使用缓存存储的一些页面,这时用 GET 方法访问历史页面时得到的不是服务器上的页面,而是浏览器缓存的页面。

2) HTTP 响应报文

HTTP 响应报文的格式如下:

HTTP 版本 状态码 状态描述

若干响应首部

空行

响应体

一个典型的 HTTP 响应, 如:

HTTP/1.1 200 OK

Content-type : text/plain

Content-length : 777

...

<html> ...

响应报文的起始行称为响应行, 响应行表示服务器响应客户端请求的状态。其中, 状态码之后为状态码的描述, 使得状态码能容易被理解, 在 HTTP 规范中对它并没有严格的规定。例如, 该例中状态码是 200, 描述为 OK, 表示成功响应了。而响应首部 Content-type 表示发送的响应主体的文件类型, 另一个响应首部 Content-length 则表示该文件的准确大小。该例中虽然省略了完整的响应体, 但可以明确的是, 响应体是一个 html 页面。

不同的状态码表示不同的状态, 状态码的第一个数字一般表明大致的状态, 如表 1.2 所示, 2 开头的表示成功, 3 开头的表示已经被重定向, 4 开头的表示客户端请求有误, 而经常出现的 404 请求, 表示无法找到请求文件。

表 1.2 HTTP 协议响应状态码

状态码范围	出错分类
100~101	信息提示
200~206	成功
300~305	重定向
400~415	客户端错误
500~505	服务端错误

HTTP 协议报文中常用的首部分为通用首部、请求首部、响应首部和实体首部。前面已经介绍的 Host 和 Accept 是请求首部, 只能用在请求报文中; Connection 是通用首部, 可以用在请求和响应报文中; Content-type 和 Content-length 是实体首部(请求体(请求报文)和响应体(响应报文)统称实体), 有些实体首部可通用, 有些不能通用。常见的首部还有:

Data: 通用首部, 提供创建报文的日期和时间标志。

Pragma: 通用首部, 是随报文传送指示的一种方式, 可以用于缓存等。

Referer: 请求首部, 指示发出当前请求的 URL。

User-Agent: 请求首部, 表示发出请求的程序, 通常是浏览器的详细信息。

If-Modified-Since: 请求首部, 指示服务器, 若文档在某时间之后未修改, 则不用响应。

Authorization: 请求首部, 为服务器提供对客户端身份做验证的数据。

Cookie: 请求首部, Cookie 数据。

Age: 响应首部, 响应的持续时间。

Set-Cookie: 响应首部, 设置 Cookie。

WWW-Authenticate: 响应首部, 服务器对客户端的质询数据。

Location: 实体首部, 向客户端指出重定向位置。

Expires: 实体首部, 表示实体不再有效, 需要源端重新发送。

Last-Modified: 实体首部, 表示实体最后一次修改的时间。

2. URL

URL(Uniform Resource Locator, 统一资源定位符)表示了 Internet 上的一个资源的引用和地址。Web 应用程序是使用 URL 来定位访问 Internet 上的资源的。

URL 由两部分组成, 通过 “://” 分隔:

(1) 协议标识, 访问资源所用的协议, 如 HTTP、FTP 等。

(2) 资源名称, 表示要访问资源的地址, 包括: 主机名、文件名、端口号、引用。

如果通过 HTTP 协议访问服务器上的文件, 则目标地址的格式应该是: `http://域名或 IP 地址[:端口号]/文档路径`。

这里的文档路径是服务器允许客户端访问的文件位置, 是一个相对地址。服务器对外提供服务的地址是该文档路径的根目录, 服务器不会允许客户端访问该根目录以外的位置。当客户端提交请求希望访问某个文档路径时, 服务器将会对该文档路径对本地的文档进行映射, 映射成本地文档的绝对路径。

按照一般惯例, 直接访问某个文件夹时, 访问的将是名为 `index.jsp` 的页面。可以用下面的例子进行说明: 设有一个部署于 Linux 系统 80 端口下的 Tomcat 局域网服务器, IP 地址是 192.168.0.10, 它安装于 “/home/user/apache-tomcat-8.01” 下, 对外提供服务的文件夹是 `webapps`。若访问 192.168.0.10 下的 `dashboard` 资源, 则应输入 “192.168.0.10/dashboard”, 经过地址映射, 实际访问的是该计算机上的 “/home/user/apache-tomcat-8.01/webapps/dashboard/index.jsp” 文件。其中, 默认的 HTTP 端口可以不用填写。

HTTP 协议可以传输不同的文件格式, MIME(多用途网际邮件扩充协议)指明文件格式, 方便客户端处理。这个协议本身用于通过电子邮件传输不同格式的文件, 应用在 Web 领域中, 使得浏览器可以根据服务器发送的响应首部(Content-type 中指定的 MIME 信息)处理不同格式的文件, 浏览器也可以通过 Accept 对服务器指明, 自己可以接收什么样的文件。

MIME 的格式是: 类型/子类型, 例如, `text/html` 和 `image/gif` 是常见的 MIME 类型, 前者表示类型是 `text`, 子类型是 `html`, 后者表示类型是 `image`, 子类型是 `gif`。

1.2 网站的静态与动态

1.2.1 静态网页与应用

Web 网页可分为静态网页和动态网页, 最开始多采用静态页面, 随着应用需求的深入和 Web 技术的不断更新, 目前, 动态网页已基本取代了静态网页。

静态页面是指返回页面的方式是“静态”的。如果页面在服务器端的状态不会随着请

求变化而变化，这样的页面就是静态页面。但如果请求同一个页面时的参数不同，返回的结果随之改变，则称为动态页面。这种改变可能是由于查询或修改了数据库，也可能是改变了服务器内的数据。简而言之，动态页面是可以和服务器交互的，而静态页面则不具备这样的功能。

静态页面大部分是一个 HTML 文件，也可以是其他标记语言文件，如下列的 HTML 静态页面：

```
<html>
<head>
<title>我的第一个网页</title>
</head>
<body>
  <b>欢迎光临本站!</b>
</body>
</html>
```

该页面的运行结果如图 1.3 所示，其在任何时候的运行结果都不会发生改变，如要改变只能修改页面文件 HTML 代码。



图 1.3 HTML 静态页面运行结果

随着 Web 技术的发展，简单的静态页面已经无法满足需求，一种更高级的网站开发技术正慢慢在互联网中“蔓延”，那就是动态网页开发。这里所指的动态与网页中动态的图片没有任何关系，而是指能够根据用户的请求，动态的生成静态页面的一种网页开发技术。动态页面可以是 JSP、PHP、ASP 页面，或者是程序动态产生的页面，它们在服务器端执行，并将执行结果以 HTML 页面的方式返回给客户端。

例如，我们在某网站中购买书籍时，可以在“搜索商品”栏中输入要查找的书籍：Java 程序设计，则会返回如图 1.4 所示的结果页面。随着输入书籍的不同，即用户请求的不同，则返回结果也不相同，这种页面称为动态页面。

静态页面的功能简单，反应快速，适合作为不需要经常变动显示的页面，比如项目的介绍，信息的发布页面。除了 HTML 之外，XML 和 Makedown 等标记语言也可以经服务器引擎解析后，转译成 HTML 作为静态页面应用。而动态页面的功能将会更加强大，并且

在一个 Web 系统中, 页面主要由动态页面构成。



图 1.4 动态页面“搜索书籍”执行结果

1.2.2 用户与浏览器的动态交互

1. 前端技术

纯粹的 HTML 页面作为 Web 用户界面, 相比于直接和操作系统打交道的 GUI 界面, 不仅页面的显示单一, 功能也是非常弱的。

在客户端浏览器中, 为了增强界面的显示效果, 使用层叠样式表(CSS)分离 HTML 元素和页面的表现形式, 可以精确地设计网页的视觉效果与风格。为了方便用户对页面元素的操纵, 增强交互功能, 可以使用客户端脚本编程语言, 如 JavaScript 脚本, 响应页面上的动作, 与用户进行交互。这两种技术强化了 Web 页面的表现能力和交互能力。需要注意的是, 仅包含客户端脚本代码的页面仍然属于静态页面, 因为客户端脚本代码是由浏览器来解释执行的, 不与 Web 服务器交互。

对于一个 Web 系统而言, 表示层也称为前端。使用 HTML、CSS 和 JavaScript 等脚本语言设计复杂页面的技术称为前端技术, 而产生动态页面、设计 Web 服务器后台业务、查询数据库的相关技术称为后端开发技术。目前, 前端所涵盖的技术在不断发展更新, 如 HTML5、CSS3、Ajax 等。

2. JavaScript 和浏览器

JavaScript 是目前最通用的浏览器脚本语言, 可以嵌入到 HTML 代码中, 并能通过文档对象模型接口访问浏览器页面, 其解释器内置于几乎所有的主流浏览器中。因此, 通过 JavaScript 脚本语言可以动态地创建和修改 HTML 文档, 增强用户和浏览器的交互能力, 提高了网页的易用性和使用效率, 例如, 它可以让页面呈现一些特殊的效果, 对用户的表

单数据进行验证，也可以响应用户对页面的点击、移动等多种操作。

随着 HTML5 标准的推出，JavaScript 的功能越来越强大，除了访问浏览器页面，也可以调用许多系统功能，例如，操作本地文件、图片、摄像头等，从而增加了 Web 用户体验的方式。

在 1995 年互联网兴起时，JavaScript 据说是仅花费 10 天就开发出来的一门语言，诸多设计漏洞被人诟病。2005 年，Google 地图大量使用基于 JavaScript 的 Ajax 技术，不仅标志着 Web2.0 时代的到来，也将提供浏览器与用户交互功能的 JavaScript 推上新的高度。

1.2.3 开发动态网站的主流技术

开发一个动态网站的技术有多种。在实现一个桌面应用程序时，需要依赖于所运行的操作系统，因此最好的选择是对应的操作系统相匹配，但作为 Web 应用，运行在 Web 服务器上的程序可以运行在更高层次的应用平台上，可以不受操作系统的限制，如某种语言的虚拟机或者某种解释器之上。因此，可以乐观地说，开发一个 Web 程序有多种选择，如何选择受开发者喜好的影响。

从实际应用的角度来考虑，使用者除了使用自己熟悉的开发语言外，还应该结合项目的特点来选择语言，另一方面，语言本身富含的哲学思想也会影响开发项目中反射出的文化与核心价值。下面先简要介绍几种常用的 Web 程序开发语言。

1. 经典的网络编程语言

1) PHP

PHP(HyperText Preprocessor)是非常流行的动态网页的开发技术，它在 1994 年由 Apache 团队的 Rasmus Lerdorf 专为 Web 应用程序而开发。PHP 是一种动态类型的服务器端脚本语言，可以直接嵌入 HTML 文档中，为解释执行方式。它能够有效快速地开发网页，调试方便，而且性能较好。PHP 语言的哲学中强调完成(Done is better than perfect)，对数据库访问技术有很好的支持，并且易于入门，适合开发个人或小型 Web 项目。

2) C#/.NET

C# 是微软公司于 2000 年随着 .NET Framework 发布，由研究员 Anders Hejlsberg 主持设计的面向对象的新型语言，其最初的目的是想在 Web 领域与 Java 语言中进行竞争。

在许多优秀程序语言的基础之上，C# 吸取了 Java 的单继承、接口和垃圾回收机制，也吸收了 C++ 的指针、运算符重载和预处理方法，还添加了委托和属性等特性，是一门功能很强大的语言。.NET 平台可以使用包括 C# 在内的多种语言，C#/.NET 通过 ASP.NET (Active Server Pages)框架构建动态页面，它有着 Visual Studio 这个功能强大的 IDE 开发工具和丰富的参考文档。但 .NET Framework 作为微软的产品，尽管已转向一些非 Windows 平台，但其使用还是非常有限的，只能运行在微软操作系统下，这也是 .NET 的一个缺点。

3) Java Web

Java Web 作为 Java EE 架构的一个组成部分，部署多样，体系成熟，在业界得到了广泛应用。Java 的跨平台性非常适合异构网络的分布式开发，易于部署、管理和维护。Java 是一门静态类型的语言，具有良好的性能和健壮性，虽然开发周期较长，但是语言功能很全面。

在 Java Web 应用程序中,通过称为 Servlet 的服务器端程序处理客户端请求和响应。Servlet 的执行由 Servlet 容器进行管理,Servlet 容器有时称为 Servlet 引擎。JSP(Java Server Page)脚本技术可以将 Java 代码嵌入 HTML 文档中,从而动态生成 Web 页面。由于 JSP 页面最终会被编译成 Servlet,因此可以将 JSP 技术视作 Servlet 的一种脚本编程方式,方便易学。另外,JSF(Java Server Faces)框架可以建立应用程序的事件驱动的用户界面模型。JavaBean 是一个被封装的可重用组件,它可以用来支持基于组件的 Web 程序设计。

2. 发展中的 Web 语言

大部分情况下,普通的个人网站的开发并不需要规模庞大的 Web 程序,而经典的编程语言功能健全,但设计复杂,较长的开发周期并不太适合这类问题的需要。因此,对于敏捷开发的需求使得一些小巧的编程语言逐渐受到用户的认可,并在 Web 领域也能够占有一席之地。

1) Python

Python 是荷兰人 Guido van Rossum 于 1991 年发布的一种解释型、面向对象、动态类型的高级程序设计语言。用 Python 编写的程序非常简洁,结构自然优美,清晰易懂,具有高度可读性(比如其强制缩进的风格)。Python 语言的学习简单而且容易入门,适合非计算机专业学生编程语言,也是目前最受欢迎的编程语言之一。

在 NASA、Yahoo、Google 等公司组织中都使用 Python 编写了一些模块的组件,并应用于其系统中。Python 拥有非常丰富的类库,功能较为强大,可以应用在网络分析、文本处理、科学计算、图形图像等众多领域。Python 程序面向于解决实际问题,问题的解决方案直接且唯一。代码长度相对于许多其他流行的编程语言要短小精悍,实用性很强。由于它是一种解释型的脚本语言,在 Web 程序开发中使用会比较方便,但并发能力不太高。

2) Ruby/Ruby on Rails

Ruby 是 1995 年由日本人松本行弘(Matz)发布的一门面向对象的脚本语言,也是目前世界上唯一一个由亚洲人创造的主流编程语言。Ruby 是作者为了改进 Perl 和 Python 而设计的,因此继承了两者的许多特点,比如正则表达式和优雅简洁的代码。

Ruby 运行于解释器上,因此也属于跨平台的语言,它支持多线程,还可以直接调用操作系统的系统调用。Ruby 常常被用来创建针对相对窄的应用范围的编程语言—领域特定语言 DSL(Domain Specific Language)。

Rails 是基于 Ruby 的 Web 应用开发框架,由丹麦人 David Heinemeier Hansson 于 2004 年 7 月公开发布。Rails 应用程序采用的是“模式—视图—控制器”体系结构,即 MVC 结构,所以构建一个 Rails 应用程序需要设计和构建 MVC 的三个组成部分。同时它也是一种新的软件开发范式——敏捷开发(Agile Development)的代表,敏捷开发注重人与人之间的交互,包括开发人员之间的沟通和开发人员与用户之间的沟通,而且强调测试非常重要(Testing is critical)等。Rails 的两个重要原则是,“自己不要重复”DRY(Do not Repeat Yourself)和“约定优于配置”(convention over configuration)。

DRY 是指避免代码重复,每个信息元素只在系统中出现一次,这样可以最小化系统所需内存。而约定优于配置则表示不使用详尽复杂的 XML 配置文件制定 Web 程序的结构。在 Rails 中,应用程序的结构由 MVC 体系规定,不同的部分之间的连接通过约定建立,不