

21世纪高等教育计算机规划教材

COMPUTER

PHP 动态网站 程序设计

PHP Website Development

唐四薪 肖望喜 唐琼 主编

循序渐进掌握 PHP Web 开发基本原理

对 PHP 的传统内容去粗取精

重点讲解 PHP 访问数据库



21世纪高等教育计算机规划教材

COMPUTER

PHP 动态网站 程序设计

PHP Website Development

唐四新 肖望喜 唐琼 主编



人民邮电出版社

北京

图书在版编目 (C I P) 数据

PHP动态网站程序设计 / 唐四薪, 肖望喜, 唐琼主编

— 北京: 人民邮电出版社, 2014. 8

21世纪高等教育计算机规划教材

ISBN 978-7-115-35656-7

I. ①P… II. ①唐… ②肖… ③唐… III. ①PHP语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第127216号

内 容 提 要

PHP 是动态网站开发领域的流行技术。本书系统地介绍了 PHP 的核心知识,并在叙述有关原理时安排了大量的相关实例。本书分为 8 章,内容包括动态网站制作概述、HTML、PHP 语言基础、函数和面向对象编程、Web 交互编程、MySQL 数据库、PHP 访问数据库、PHP 文件访问技术。附录中安排了 PHP 的相关实验。

本书适合作为高等院校各专业“Web 编程技术”或“动态网页设计”等课程的教材,也可作为 Web 编程的培训类教材,还可供网站开发人员参考使用。

-
- ◆ 主 编 唐四薪 肖望喜 唐 琼
责任编辑 刘 博
责任印制 彭志环 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.25 2014 年 8 月第 1 版
字数: 400 千字 2014 年 8 月北京第 1 次印刷

定价: 36.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

PHP 目前仍然是开发生态网站最理想的语言。相比其他 Web 编程语言, PHP 具有简单易学、功能强大、成本低廉、安全性较高和运行环境易于配置等优点, 是初学者学习 Web 应用程序设计的理想入门语言, 且能够用来制作企业级的 Web 应用程序及动态网站。

近几年来, PHP 在国内外的应用发展非常迅速, 许多大型的电子商务网站(如淘宝网等)都采用 PHP 作为网站开发的语言; 同时, 通过对众多软件企业的调查发现, 各种企业对于 PHP 开发人才的需求缺口很大。

为了能编写一本适合于高校教学的 PHP 教材, 也为了能方便读者自学, 本书在写作时注重解决以下一些问题。

(1) 对于讲解 PHP 的运行环境来说, 本书主要介绍安装 AppServ 集成运行环境, 而没有单独介绍 PHP 运行环境中几种软件的安装方法, 因为单独安装和配置各种软件, 对初学者来说比较难, 也没有必要去学习。

(2) 本书在体系结构上仿照一些经典的 ASP 教材进行编写, 如果读者具有 ASP 编程基础, 就能够很快通过体会 PHP 和 ASP 的异同, 来领会 PHP 编程的思路。如果读者不具有任何 Web 编程经验, 本书也能让读者循序渐进地掌握 PHP Web 开发的基本原理。

(3) 对 PHP 访问数据库进行了重点讲解。分别介绍了 MySQL 函数、MySQLI 函数和 PDO 方法访问数据库, 并在介绍完每种方法的原理后, 都安排了一节的实例与习题。

(4) 对 PHP 的传统内容去粗取精。Web 应用程序的功能主要就是查询、添加、删除和修改记录, 因此本书对这些功能的实现进行了重点叙述, 在普通的 PHP 程序、生成静态网页的 PHP 程序中分别实现了查询、添加、删除和修改等基本功能模块。

(5) 在传统 PHP 教材的基础上, 增加了新的流行内容, 如分别在数据库端和 Web 服务器端实现分页程序, 用 PHP 生成静态 HTML 文件的新闻系统, PHP 生成 XML 或 RSS 文件。

本书的内容包括 PHP 网站程序设计的各个方面, 如果要将本书的内容讲授完毕, 大约需要 54 学时的课时, 其中可安排 18 学时的实验课时。目录中, 小节名后注有“*”号的内容供学有余力的学生自学。

本书为使用本书作为教材的教师提供教学用多媒体课件、实例源文件和习题参考答案, 可登录本书配套网络教学平台 (<http://ec.hynu.cn>) 免费下载, 也可和作者联系 (tangsix@163.com)。另外, 该网络教学平台还提供大量网页制作的操作视频和相关软件供免费下载。

本书由唐四薪、肖望喜、唐琼担任主编，唐四薪编写了第 3~8 章的内容，肖望喜编写了第 1 章的内容，唐琼编写了第 2 章的部分内容。参加编写的还有屈瑜君、喻缘、康江林、袁建君、刘艳波、唐亮、黄大足、尹军、邹赛、邢容、邓明亮和唐金娟等。

由于编者水平和教学经验有限，书中错误和不妥之处在所难免，欢迎广大读者和同行批评指正。

编者
2014 年 4 月

目 录

第 1 章 动态网站制作概述 1

1.1 动态网站的原理 1

1.1.1 动态网站的起源 1

1.1.2 动态网站的组成与运行 2

1.1.3 动态网站开发语言 3

1.1.4 动态网站的有关概念* 5

1.2 网页的类型和工作原理 7

1.2.1 静态网页和动态网页 7

1.2.2 为什么需要动态网页 9

1.2.3 PHP 动态网页的工作原理 9

1.3 安装 PHP 的运行环境 10

1.3.1 AppServ 的安装 11

1.3.2 运行第一个 PHP 程序 14

1.3.3 Apache 的配置 16

1.3.4 在 IIS 中集成 PHP 运行环境* 19

1.4 使用 Dreamweaver 开发 PHP 程序 20

1.4.1 新建动态站点 21

1.4.2 编写并运行 PHP 程序 23

1.5 Web 服务器软件 23

习题 24

第 2 章 HTML 26

2.1 HTML 概述 26

2.1.1 HTML 文档的结构 26

2.1.2 Dreamweaver 的开发界面 27

2.1.3 使用 DW 新建 HTML 文件 28

2.1.4 HTML 标记 28

2.1.5 HTML 元素及其分类 29

2.2 文本、图像和超链接标记 30

2.2.1 文本格式标记 30

2.2.2 列表标记 32

2.2.3 图像标记 33

2.2.4 超链接标记<a> 33

2.3 表格标记 35

2.3.1 <table>标记及其属性 35

2.3.2 <tr>、<td>、<th>标记的属性 37

2.3.3 单元格的合并属性 39

2.4 表单标记 39

2.4.1 <form>标记及其属性 40

2.4.2 <input/>标记 42

2.4.3 <select>和<option>标记 44

2.4.4 多行文本域标记<textarea> 45

2.4.5 表单数据的传递过程 45

2.5 CSS 和 JavaScript 的嵌入 46

2.5.1 在 HTML 中引入 CSS 47

2.5.2 在 HTML 中嵌入 JavaScript 48

习题 49

第 3 章 PHP 语言基础 51

3.1 PHP 语法入门 51

3.1.1 PHP 代码的基本格式 51

3.1.2 简单 PHP 程序示例 52

3.2 常量、变量和运算符 54

3.2.1 PHP 的常量和变量 54

3.2.2 变量的作用域和生存期 55

3.2.3 可变变量和引用赋值 57

3.2.4 运算符和表达式 58

3.3 数据类型和类型转换 60

3.3.1 PHP 的数据类型 60

3.3.2 字符串数据类型 60

3.3.3 数据类型的转换	62	5.1.5 使用\$_SERVER[]获取环境 变量信息	108
3.4 PHP 的语句	64	5.2 发送数据给浏览器	109
3.4.1 条件控制语句	64	5.2.1 使用 echo 方法输出信息	109
3.4.2 循环控制语句	65	5.2.2 使用 header()函数重定向网页	109
3.4.3 文件包含语句	68	5.2.3 操作缓冲区	111
3.5 数组	69	5.3 使用\$_SESSION 设置和获取 Session	112
3.5.1 数组的创建	69	5.3.1 存储和读取 Session 信息	113
3.5.2 访问数组元素或数组	70	5.3.2 Session 的创建过程和有效期	114
3.5.3 多维数组	71	5.3.3 利用 Session 限制未登录 用户的访问	116
3.5.4 操作数组的内置函数	72	5.3.4 删除和销毁 Session	117
习题	76	5.4 使用\$_COOKIE 读取 Cookie	118
第 4 章 函数和面向对象编程	80	5.4.1 创建和修改 Cookie	118
4.1 PHP 的内置函数	80	5.4.2 读取 Cookie	119
4.1.1 字符串处理函数	80	5.4.3 Cookie 数组	120
4.1.2 日期和时间函数	82	5.4.4 删除 Cookie	120
4.1.3 检验函数	84	5.4.5 Cookie 程序设计举例	120
4.1.4 数学函数	86	5.4.6 Cookie 和 Session 的比较	123
4.2 自定义函数及调用	87	5.5 使用\$_FILES 获取上传文件信息	124
4.2.1 函数的定义	87	5.5.1 添加上传文件的表单	124
4.2.2 函数的调用	87	5.5.2 使用\$_FILES 获取上传文件 信息	124
4.2.3 变量函数和匿名函数	90	5.5.3 保存上传文件到指定目录	125
4.2.4 传值赋值和传地址赋值	90	5.5.4 同时上传多个文件	126
4.3 面向对象编程	91	习题	128
4.3.1 类和对象	91		
4.3.2 类的继承和多态	94		
习题	96		
第 5 章 Web 交互编程	99	第 6 章 MySQL 数据库	131
5.1 接收浏览器数据	99	6.1 数据库的基本知识	131
5.1.1 使用\$_POST[]获取表单数据	99	6.2 MySQL 数据库的使用	132
5.1.2 使用\$_GET[]获取表单数据	104	6.2.1 使用 phpMyAdmin 管理 数据库	132
5.1.3 使用\$_GET[]获取 URL 字符 串信息	105	6.2.2 使用 phpMyAdmin 导出 导入数据	136
5.1.4 发送 HTTP 请求的基本方法	107		

6.2.3 使用 Navicat 管理数据库·····	136	7.5.1 为网站引用后台程序和数据库·····	174
6.3 SQL 语言简介·····	138	7.5.2 在首页显示数据表中的新闻·····	176
6.3.1 Select 语句·····	139	7.5.3 制作动态图片轮显效果·····	178
6.3.2 添加、删除、更新记录的语句·····	141	7.5.4 制作显示新闻详细页面·····	180
6.3.3 SQL 字符串中含有变量的书写方法·····	143	7.5.5 制作栏目首页·····	182
习题·····	144	7.5.6 FCKeditor 的使用·····	183
第 7 章 PHP 访问数据库·····	145	7.6 数据库接口层 PDO·····	186
7.1 访问 MySQL 数据库·····	145	7.6.1 PDO 的安装·····	187
7.1.1 连接 MySQL 数据库·····	146	7.6.2 创建 PDO 对象连接数据库·····	187
7.1.2 创建结果集并输出记录·····	147	7.6.3 使用 query()方法执行查询·····	189
7.1.3 使用 mysql_query()增、删、改记录·····	151	7.6.4 使用 fetchAll()方法返回所有行·····	190
7.2 添加、删除、修改记录的综合实例·····	152	7.6.5 使用 exec()方法执行增、删、改命令·····	190
7.2.1 管理记录主页面的设计·····	152	7.6.6 使用 prepare()方法执行预处理语句·····	190
7.2.2 添加记录的实现·····	153	7.7 用 PDO 制作博客网站·····	192
7.2.3 删除记录的实现·····	155	7.7.1 数据库的设计·····	192
7.2.4 同时删除多条记录的实现·····	156	7.7.2 首页的制作·····	193
7.2.5 修改记录的实现·····	157	7.7.3 留言模块的制作·····	197
7.2.6 查询记录的实现·····	159	7.7.4 博客后台登录的实现·····	199
7.3 分页显示数据·····	160	习题·····	200
7.3.1 分页程序的基本实现·····	161	第 8 章 PHP 文件访问技术·····	202
7.3.2 对查询结果进行分页·····	164	8.1 文件访问函数·····	202
7.3.3 将分页程序写成函数·····	166	8.1.1 打开和关闭文件·····	202
7.3.4 可设置每页显示记录数的分页程序·····	168	8.1.2 读取文件·····	203
7.4 mysqli 扩展函数的使用·····	169	8.1.3 移动文件指针·····	205
7.4.1 连接 mysql 数据库·····	169	8.1.4 文本文件的写入和追加·····	206
7.4.2 执行 SQL 语句创建结果集·····	170	8.1.5 读写文件的应用——制作计数器·····	207
7.4.3 从结果集中获取数据·····	171	8.2 文件及目录的基本操作·····	209
7.4.4 同时执行多条 SQL 语句·····	172	8.2.1 复制、移动和删除文件·····	209
7.5 新闻网站综合实例·····	173	8.2.2 获取文件属性·····	210

8.2.3 目录的基本操作	211	A.2 实验 2: PHP 语言基础	230
8.2.4 统计目录和磁盘大小	214	A.3 实验 3: 函数的定义和调用	231
8.3 制作生成静态页面的新闻系统	214	A.4 实验 4: 面向对象程序设计	232
8.3.1 数据库设计和制作模板页	215	A.5 实验 5: 获取表单及 URL 参数中的 数据	232
8.3.2 新闻添加页面和程序的制作	216	A.6 实验 6: Session 和 Cookie 的 使用	233
8.3.3 新闻后台管理页面的制作	218	A.7 实验 7: MySQL 数据库的管理	233
8.3.4 新闻修改页面的制作	219	A.8 实验 8: 在 PHP 中访问 MySQL 数据库	234
8.3.5 新闻删除页面的制作	221	A.9 实验 9: 分页程序的设计	235
8.3.6 首页和列表页的静态化	222	A.10 实验 10: 使用 mysqli 函数访问 数据库	235
8.4 利用文件访问函数生成 RSS 文件	225	A.11 实验 11: 使用 PDO 访问数据库	236
8.4.1 XML 语法概述	225		
8.4.2 RSS 概述	226		
8.4.3 用 PHP 输出 RSS 文档	227		
习题	228		

附录 实验

A.1 实验 1: 搭建 PHP 运行和开发 环境	230
------------------------------------	-----

第 1 章

动态网站制作概述

随着互联网技术的应用和普及，各行各业制作网站的热情高涨。目前的网站一般都是动态网站，简单地说，动态网站是一种使用 HTTP（超文本传输协议）作为通信协议，通过网络让浏览器与服务器进行通信的计算机程序。动态网站开发可分为两个方面：一是网站的界面设计，主要是用浏览器能理解的代码及图片设计网页的界面；二是网站的程序设计，用来实现网站的新闻管理、与用户进行交互等各种功能。

1.1 动态网站的原理

1.1.1 动态网站的起源

动态网站是一种基于 B/S 结构的网络程序。那么什么是 B/S 结构呢？这就先要从网络软件的应用模式说起。

早期的应用程序都是运行在单机上的，称为桌面应用程序。后来由于网络的普及，出现了运行在网络上的网络应用程序（网络软件），网络应用程序有 C/S 和 B/S 两种体系结构。

1. C/S 体系结构

C/S 是 Client/Server 的缩写，即客户机/服务器结构，这种结构的软件包括客户端程序和服务器端程序两部分。比如常用的 QQ 或 MSN 等网络软件，需要下载并安装专用的客户端软件（见图 1-1），并且服务器端也需要特定的软件支持才能运行。

C/S 模式最大的缺点是不易于部署，因为每台客户端电脑都要安装客户端软件。而且，如果客户端软件需要升级，则必须为每台客户端单独升级。另外，客户端软件通常对客户机的操作系统也有要求，例如有些客户端软件只能运行在 Windows 平台下。

2. B/S 体系结构

B/S 是 Browser/Server 的缩写，即浏览器/服务器结构。它是随着 Internet 技术的兴起，对 C/S 结构的一种变化或者改进的结构。在这种结构下，客户端软件由浏览器来代替（见图 1-2），一部分事务逻辑在浏览器端（Browser）实现，但是主要事务逻辑在服务器端（Server）实现，目前流行的是三层 B/S 结构，即表现层、事务逻辑层和数据处理层。



图 1-1 C/S 结构的 QQ 客户端界面

B/S 结构很好地解决了 C/S 结构的上述缺点。因为每台客户端电脑都自带有浏览器，就不需要额外安装客户端软件了，也就不存在客户端软件升级的问题了。另外，由于任何操作系统一般都带有浏览器，因此 B/S 结构对客户端的操作系统也没有要求了。

但是与 C/S 结构相比，B/S 结构也有其自身的缺点。首先，因为 B/S 结构的客户端软件界面就是网页，因此操作界面不可能做得很复杂，比如很难实现树型菜单、选项卡式面板或右键快捷菜单等（或者虽然能够模拟实现，但是响应速度比 C/S 中的客户端软件要慢很多）。其次，B/S 结构下的每次操作一般都要刷新网页，响应速度明显不如 C/S 结构。再次，在网页操作界面下，操作大多以鼠标方式为主，无法定义快捷键，也就无法满足快速操作的需求。



图 1-2 B/S 结构的浏览器端界面



因为 C/S 结构和 B/S 结构的网络软件，其程序都是分布在客户机和服务器上，因此它们统称为分布式系统（Distributed System）。

1.1.2 动态网站的组成与运行

1. 动态网站的组成

动态网站通常由 HTML 文件、服务器端脚本文件和一些资源文件组成。

- ① HTML 文件提供静态的网页内容。
- ② 脚本文件提供程序，实现客户端与服务器之间的交互以及访问数据库或文件等。
- ③ 资源文件提供网站中的图片、视频等资源，包括图片文件、多媒体文件和配置文件等。

2. 运行动态网站程序的要素

要运行动态网站程序，需要 Web 服务器、浏览器和 HTTP 通信协议等要素。

(1) Web 服务器

运行动态网站需要一个载体，称为 Web 服务器。一个 Web 服务器可以部署多个动态网站（或 Web 应用程序）。

通常 Web 服务器有两层含义，一方面它代表运行 Web 应用程序的计算机硬件设备，一台计算机只要安装了操作系统和 Web 服务器软件，就可算作一台 Web 服务器。另一方面 Web 服务器专指一种软件——Web 服务器软件，该软件的功能是响应用户通过浏览器提交的 HTTP 请求，例如用户请求的是 PHP 脚本，则 Web 服务器软件将解析并执行 PHP 脚本，生成 HTML 格式的文本，并发送到客户端，显示在浏览器中。

(2) 浏览器

浏览器是用于解析 HTML 文件（可包括 CSS 代码和客户端 JavaScript 脚本）并显示的应用程序，它可以从 Web 服务器接收、解析和显示信息资源（可以是网页或图像等），信息资源一般使用统一资源定位符（URL）标识。

浏览器只能解析和显示 HTML 文件，而无法处理服务器端脚本文件（如 PHP 文件），这就是为什么能直接用浏览器打开 HTML 网页文件，而服务器端脚本文件只有被放置在 Web 服务器上才能被正常浏览的原因。

(3) HTTP 通信协议

HTTP 是浏览器与 Web 服务器之间通信的语言。浏览器与服务器的会话（见图 1-3），总是由浏览器向服务器发送 HTTP 请求信息开始（如用户输入网址，请求某个网页文件），Web 服务器根据请求返回相应的信息。这一过程被称为 HTTP 响应，响应中包含请求的完整状态信息，并在消息体中包含请求的内容（如用户请求的网页文件内容等）。

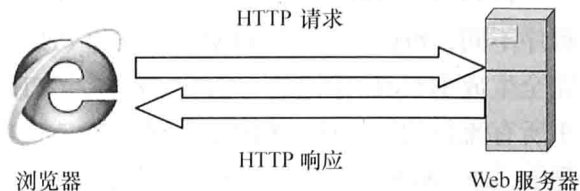


图 1-3 浏览器与服务器的会话

3. 动态网站与 Web 应用程序

一般来说，网站的内容需要经常更新，并添加新内容。早期的网站是静态的，更新静态网站的内容是非常繁琐的，例如要增加一个新网页，就需要手工编辑这个网页的 HTML 代码，然后再更新相关页面到这个页面的链接，最后把所有更新过的页面重新上传到服务器上。

为了提高网站内容更新的效率，可以通过构建 Web 应用程序来管理网站内容。Web 应用程序可以把网站的 HTML 页面部分和数据部分分开。要更新或添加新网页，只要在数据库中更新或添加记录就可以了，程序会自动读取数据库中的记录，生成新的页面代码发送给浏览器，从而实现网站内容的动态更新。

可见，Web 应用程序能够动态生成网页代码，Web 应用程序可以通过各种服务器端脚本语言来编写。而服务器端脚本代码是可以嵌入到网页的 HTML 代码中的，嵌入了服务器端脚本代码的网页就称为动态网页文件。因此，如果一个网站中含有动态网页文件，则这个网站就相当于是一个 Web 应用程序。

Web 应用程序是 B/S 结构软件的产物。它首先是“应用程序”，与标准的程序语言，如 C、C++编写出来的程序没有本质的区别。然而 Web 应用程序又有其自身独特的地方，表现在：①Web 应用程序是基于 Web 的，依赖于通用的 Web 浏览器来表现它的执行结果；②需要一台 Web 服务器，在服务器上对数据进行处理，并将处理结果生成网页，以方便客户端直接使用浏览器浏览。

利用 Web 应用程序，网站可以实现动态更新页面，以及与用户进行交互（如留言板、论坛、博客、发表评论）等各种功能。但 Web 应用程序并不等同于动态网站，它们的侧重点不同。一般来说，动态网站侧重于给用户提供的信息，而 Web 应用程序侧重于完成某种特定任务，如基于 B/S 的管理信息系统（Management Information System, MIS）就是一种 Web 应用程序，但不能称作网站。Web 应用程序的真正核心功能是对数据库进行处理。

1.1.3 动态网站开发语言

动态网站开发语言用来编写动态网站的服务器端程序。目前流行的动态网站开发语言有 CGI、PHP、ASP、ASP.NET 和 JSP 等，下面分别进行介绍。

1. CGI

最早能够动态生成 HTML 页面的技术是 CGI（Common Gateway Interface，通用网关接口），由美国 NCSA（National Center for Supercomputing Applications）于 1993 年提出。CGI 技术允许服务器端应用程序根据客户端的请求，动态生成 HTML 页面。早期的 CGI 大多是编译后的可执行程序，其编程语言可以是 C、C++等任何通用的程序设计语言，也可以是 Perl、Python 等脚本语言。但是，CGI 程序的编写比较复杂而且效率低，并且每次修改程序后都必须将 CGI 的

源程序重新编译成可执行文件，因此目前很少有人使用 CGI 技术。

2. PHP

1994 年，Rasmus Lerdorf 发明了专门用于 Web 服务器编程的 PHP 工具语言。与以往的 CGI 程序不同，PHP 语言将 HTML 代码和 PHP 指令结合成为完整的服务器端动态页面，执行效率比完全生成 HTML 标记的 CGI 要高得多。PHP 的其他优点包括：跨平台并且开放源代码，支持几乎所有流行的数据库，可以运行在 UNIX、Linux 或 Windows 操作系统下。开发 PHP 时通常搭配 Apache Web 服务器和 MySQL 数据库。

3. ASP

1996 年，Microsoft 公司推出了 ASP 1.0。ASP 是 Active Server Pages 的缩写，即动态服务器页面。它是一种服务器端脚本编程环境，可以混合使用 HTML、服务器端脚本语言（VBScript 或 JavaScript）以及服务器端组件创建动态、交互的 Web 应用程序。从 Windows NT 4.0 开始，所有 Windows 操作系统都提供了 IIS（Internet Information Services）组件，它可以作为 ASP 的 Web 服务器软件。



脚本（Script）是一种可以在 Web 服务器端或浏览器端运行的程序，目前在 Web 编程上比较流行的脚本语言有 JavaScript 和 VBScript，并且一般采用 Javascript 作为客户端脚本语言，VBScript 作为服务器端脚本语言。

4. JSP

1997—1998 年，SUN 公司相继推出了 Servlet 技术和 JSP（Java Server Pages）技术。这两者的组合（还可以加上 Javabeen 技术），让程序员可以使用 Java 语言开发 Web 应用程序。

JSP 实际上是将 Java 程序片段和 JSP 标记嵌入 HTML 文档中，当客户端访问一个 JSP 网页时，将执行其中的程序片段，然后给客户端返回标准的 HTML 文档。与 ASP 不同的是，客户端每次访问 ASP 文件时，服务器都要对该文件解释执行一遍，再将生成的 HTML 代码发送给客户端。而在 JSP 中，当第一次请求 JSP 文件时，该文件会被编译成 Servlet，然后生成 HTML 文档发送给客户端。当以后再次访问该文件时，如果文件没有被修改，就直接执行已经编译生成的 Servlet，然后生成 HTML 文档发送给客户端。由于以后每次都不需要重新编译，因此在执行效率和安全性方面有明显优势。JSP 的另一个优点是可以跨平台，缺点是运行环境及 Java 语言都比较复杂，导致学习难度大。

5. ASP.NET

2002 年，Microsoft 公司正式发布了 .NET FrameWork 和 Visual Studio .NET，它引入了 ASP.NET 这种全新的 Web 开发技术。ASP.NET 可以使用 VB.net、C# 等编译型语言，支持 Web 窗体、.NET Server Control 和 ADO.NET 等高级特性。ASP.NET 应用程序最大的特点是程序与页面分离，也就是说它的程序代码可单独写在一个文件中，而不是嵌入到网页代码中。ASP.NET 需要运行在安装了 .Net FrameWork 的 IIS 服务器上。

总体来说，PHP 和 ASP 属于轻量级的 Web 程序开发环境，只要安装 DW 就可进行程序的编写。而 ASP.NET 和 JSP 属于重量级的开发平台，除了安装 DW 外，还必须安装 Visual Studio 或 Eclipse 等大型开发软件。

在本书中，之所以选择介绍 PHP 语言，主要基于以下考虑：

① PHP 是简单易学的，由于 AppServ 等集成环境的出现，配置 PHP 的 Web 服务器也是很简单的。因此，很适合初学者学习，能够让初学者在短时间内领会到 Web 应用程序开发的思路。

② 这几种语言的编程思想其实很相似，比如每种语言基本上都定义了一些服务器与浏览器之间交互信息的方法。只要深刻掌握其中一种，再去学习其他的语言就比较容易了。

1.1.4 动态网站的有关概念*

在学习动态网站编程前，有必要明确 URL、域名、HTTP 和 MIME 这些概念。

1. URL

当用户使用浏览器访问网站时，通常都会在浏览器的地址栏中输入网站地址，这个地址就是 URL (Universal Resource Locator, 统一资源定位器)。URL 信息会通过 HTTP 请求发送给服务器，服务器根据 URL 信息返回对应的网页文件代码给浏览器。

URL 是 Internet 上任何资源的标准地址，每个网站上的每个网页 (或其他文件) 在 Internet 上都有一个唯一的 URL 地址，通过网页的 URL，浏览器就能定位到目标网页或资源文件。

URL 的一般格式为：“协议名://主机名[:端口号]/[目录路径/文件名][#锚点名]”，图 1-4 所示为一个 URL 的示例。



图 1-4 URL 的结构

URL 协议名后必须接 “://”，其他各项之间用 “/” 隔开，如图 1-4 中的 URL 表示信息被放在一台被称为 www 的服务器上，hynu.cn 是一个已被注册的域名，cn 表示中国。主机名和域名合称为主机头。web/201009/是服务器网站目录下的目录路径，而 first.html 是位于上述目录下的文件名，因此该 URL 能够让我们访问到这个文件。

在 URL 中，常见的协议有 http 和 ftp。

① http: 超文本传输协议，用于传送网页，例如：

http://bbs.runsky.com:8080/bbs/display.php#fid

② ftp: 文件传输协议，用于传送文件，例如：

A. ftp://219.216.128.15/

B. ftp://001.seaweb.cn/web

2. 域名

在 URL 中，主机名通常是域名或 IP 地址。最初，域名是为了方便人们记忆 IP 地址而出现的，用户在 URL 中可以输入域名而不必输入难记的 IP 地址。但现在多个域名可对应一个 IP 地址 (一台主机)，即在一台主机上可架设多个网站，这些网站的存放方式称为“虚拟主机”方式。此时由于一个 IP 地址 (一台主机) 对应多个网站，就不能采用输入 IP 地址的方式访问网站，而只能在 URL 中输入域名。Web 服务器为了区别用户请求的是这台主机上的哪个网站，通常必须为每个网站设置“主机头”来区别这些网站。

因此域名的作用有两个：一是将域名发送给 DNS 服务器解析得到域名对应的 IP 地址以进行连接；二是将域名信息发送给 Web 服务器，通过域名与 Web 服务器上设置的“主机头”进行匹配，确认客户端请求的是哪个网站，如图 1-5 所示。若客户端没有发送域名信息给 Web 服务器，例如直接输入 IP，则 Web 服务器将打开服务器上的默认网站。

3. HTTP 请求和响应的格式

HTTP 是浏览器发送请求信息给服务器，服务器再传输超文本（或其他文档）到浏览器的传输协议，这就是在浏览器中看到的网页地址都是以“http://”开头的原因。它不仅能保证计算机正确快速地传输网页文档，还能确定传输文档中的哪一部分，以及哪部分内容首先显示（如文本先于图形）等。

HTTP 包含两个阶段：请求阶段和响应阶段。浏览器和 Web 服务器之间的每次 HTTP 通信（请求或者响应）都包含头部和主体两部分。头部包含了与通信有关的信息；主体则包含了通信的数据，当然，前提是存在这样的数据。

(1) HTTP 请求阶段

HTTP 请求的通用格式如下。

① 首行： HTTP 方法 URL 中的域名部分
HTTP 版本

- ② 头部字段
- ③ 空行
- ④ 消息主体

以下是一个 HTTP 请求首行的示例：

```
GET / content.html HTTP/1.1
```

它表示使用 GET 方式向服务器请求 content.html 这个文档，使用的协议是 HTTP 1.1 版本。对于 HTTP 方法来说，最常用的是 GET 和 POST 两种方法。GET 方法用来请求服务器返回指定文档的内容；POST 方法表示发送附加的数据并执行指定的文档，它最常见的应用是从浏览器向服务器发送表单数据，同时还发送一个请求执行服务器中的某个程序（动态页），这个程序将处理这些表单数据。

第二部分是头部字段，一个常用的头部请求字段为 Accept 字段，该字段用来指定浏览器可以接受哪些类型的文档，例如 Accept: text/html 表示浏览器只可接受 HTML 文档。文档类型采用 MIME 类型来表示。如果浏览器可以接受多种格式的文档，那么可以指定多个 Accept 字段。

请求的头部之后必须有一个空行，该空行用于将请求的主体和头部分隔开来。使用了 GET 方法的请求没有请求主体，在这种情况下，空行是请求结束的标志。

(2) HTTP 响应阶段

HTTP 响应的通用格式如下：

- ① 状态行
- ② 响应头部字段
- ③ 空行
- ④ 响应主体

状态行中包含了所用 HTTP 的版本号，此外还包括一个用三位数表示的响应状态码和针对状态码的一个简短的文本解释。例如，大部分响应都是以下面的状态行开头的。

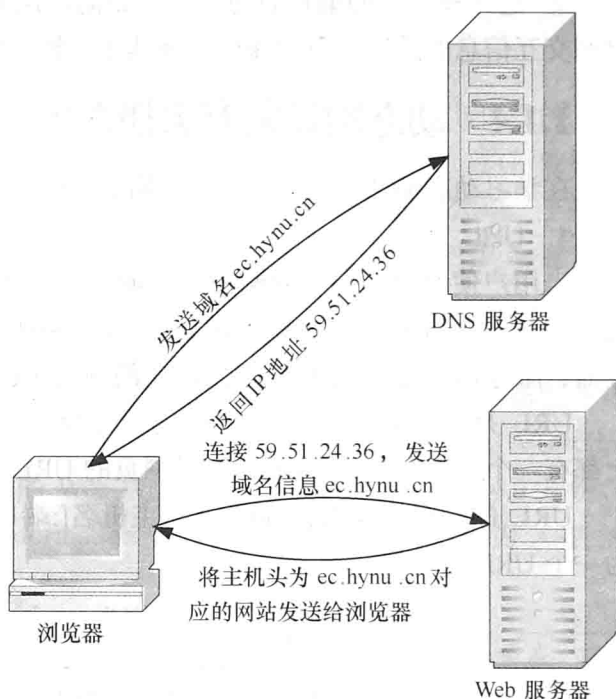


图 1-5 浏览器输入网址访问网站的过程

HTTP/1.1 200 OK

它表示响应使用的协议是 HTTP 1.1，状态码是 200，文本解释是 OK。

其中，状态码 200 表示请求得到处理，没有发生任何错误，这是用户希望看到的。状态码 404 表示请求的文件未找到。状态码 500 表示服务器出现了错误，且不能完成请求。

状态行之后是响应头部字段，响应头部可能包含多行有关响应的信息，每条信息都对应一个字段。响应头部中必须使用的字段只有一个，即 Content-type。例如：

```
Content-type: text/html, charset=UTF-8
```

它表示响应的内容是 HTML 文档，内容采用的编码方式是 UTF-8。

响应头部之后必须有一个空行，这与请求头部是一致的。空行之后才是响应数据。在上例中，响应主体是一个 HTML 文件。

4. MIME

浏览器从服务器接收返回的文档时，必须确定这个文档属于哪种格式。如果不了解文档的格式，浏览器将无法正确显示该文档，因为不同的文档格式要求使用不同的解析工具，例如服务器返回的是一个 JPG 图片格式的文档，而浏览器把它当成 HTML 文档去解析，则显示出来的将是乱码。通过多用途网际邮件扩充协议（MIME）可以指定文档的格式。

MIME 最初的目标是允许各种不同类型的文档都可以通过电子邮件发送。这些文档可能包含各种类型的文本、视频数据或者音频数据。由于 Web 也存在这方面的需求，因此 Web 中也采用了 MIME 来指定所传递的文档类型。

Web 服务器在一个将要发送到浏览器的文档头部附加了 MIME 的格式说明。当浏览器从 Web 服务器中接收到这个文档时，会根据其中包含的 MIME 格式说明来确定下一步的操作。例如，如果文档内容为文本，则 MIME 格式说明将通知浏览器文档的内容是文本，并指明具体的文本类型。MIME 说明的格式如下：

类型/子类型

最常见的 MIME 类型为 text（文本）、image（图片）和 video（视频）。其中，最常用的文本子类型为 plain、html 和 xml。最常用的图片子类型为 gif 和 jpeg。服务器通过将文件的扩展名作为类型表中的键值来确定文档的类型。例如，扩展名.html 意味着服务器应该在将文档发送给浏览器之前为文档附加 MIME 说明：text/html。

1.2 网页的类型和工作原理

网页可分为静态网页和动态网页，这两种网页的工作原理是不同的。

1.2.1 静态网页和动态网页

在 Internet 发展初期，Web 上的内容都是由静态网页组成。Web 开发就是编写一些简单的 HTML 页面，页面上包含一些文本、图片等信息资源，用户可以通过超链接浏览信息。采用静态网页的网站有很明显的局限性，如不能与用户进行交互，不能实时更新网页上的内容。因此用户留言、发表评论等功能都无法实现，只能做一些简单的展示型网站。

后来 Web 开始由静态网页向动态网页转变，这是 Web 技术经历的一次重大变革。随着动态网页的出现，用户能与网页进行交互，表现在除了能浏览网页内容外，还能改变网页内容（如发

表评论)。此时用户既是网站内容的消费者(浏览者),又是网站内容的制造者。

1. 静态网页和动态网页的区别

根据 Web 服务器是否需要为网页中脚本代码进行解释(或编译)执行,网页可分为静态网页和动态网页。

① 静态网页:是纯粹的 HTML 页面,网页的内容是固定不变的。用户每次访问静态网页时,其显示的内容都是一样的。

② 动态网页:是指网页中的内容会根据用户请求的不同而发生变化的网页,同一个网页由于每次请求的不同,可显示不同的内容,如图 1-6 中所示的两个网页实际上是同一个动态网页文件(product.php)。动态网页中可以变化的内容称为动态内容,它是由 Web 应用程序来实现的。



图 1-6 动态网页可根据请求的不同每次显示不同的内容

2. 静态网页的工作流程

用户在浏览静态网页时,Web 服务器找到网页就直接把网页文件发送给客户端,服务器不会对网页作任何处理,如图 1-7 所示。静态网页在每次浏览时,内容都不会发生变化,网页一经编写完成,其显示效果就确定了。如果要改变静态网页的内容就必须修改网页的源代码,之后再重新上传到服务器。