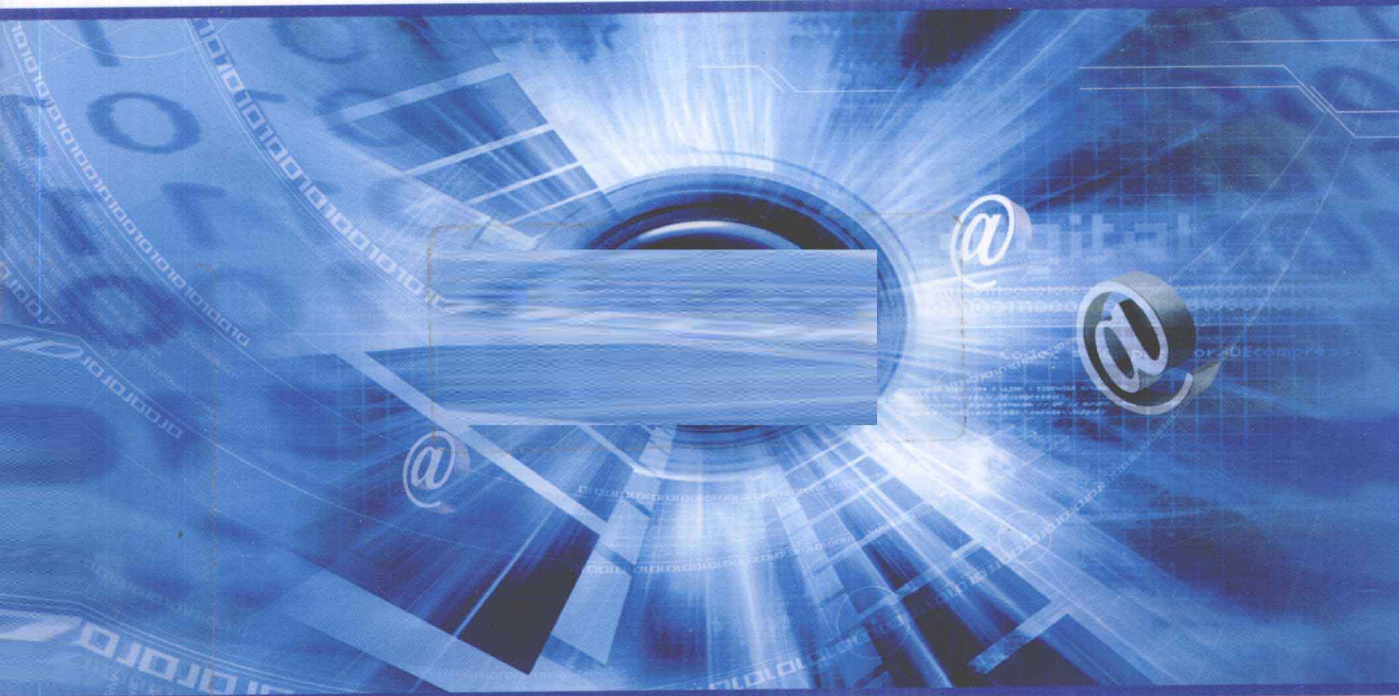


# Web网页 设计技术

王柯柯 主编

崔英志 黄丽丰 崔贯勋 刘恒洋 编著



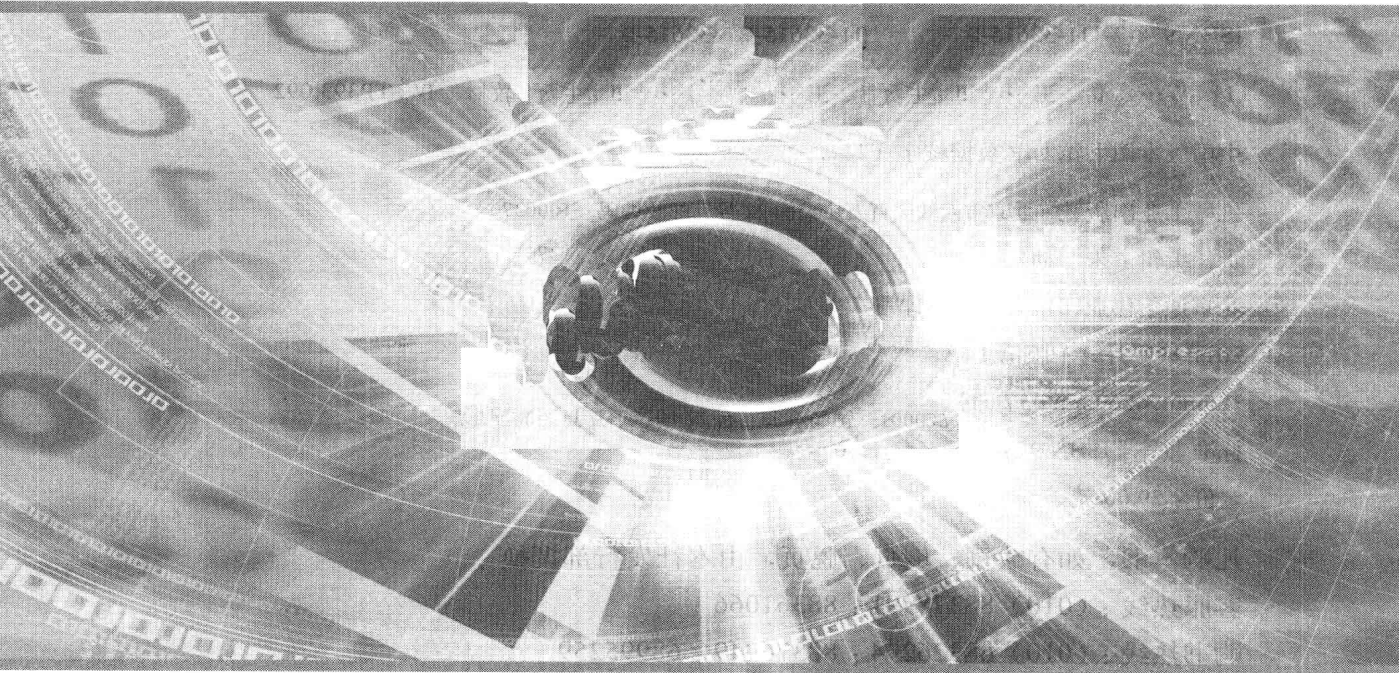
机械工业出版社  
China Machine Press

高等院校计算机教材系列

# Web网页 设计技术

王柯柯 主编

崔英志 黄丽丰 崔贯勋 刘恒洋 编著



机械工业出版社  
China Machine Press

# 前 言

本书从网站基本概念和开发工具 Visual Studio 2008 入手, 基于丰富的实例着重介绍 HTML、CSS、JavaScript 等 Web 技术的基本原理和开发方法, 深入解析典型综合案例的分析与构建过程, 帮助读者由浅入深、循序渐进地掌握开发基于 Web 应用程序的方法和技术, 具有很强的实用性。

本书重视实践, 以一个完整、翔实的案例为主线, 在各章中对知识点进行解析, 并在最后对该实例进一步汇总, 完整再现网页设计从初步规划, 到 HTML 文档创建, 再到表现样式设计的过程, 适合教师教学和学生自学。本书内容全面, 深入浅出, 实例丰富, 有助于提高学生的实际动手能力, 并结合当前最流行的 Visual Studio 2008 工具, 详细介绍在 Visual Studio 中有关前台设计的新技术(如主题、母版等), 并对利用 Visual Studio 进行网页设计进行了全方位的解析和实例分析, 在实践方面具有突出的优势。

## 主要内容

全书共 10 章, 分为四个部分, 建议学时为 40 学时, 主要内容如下:

第一部分为基础篇, 建议学时为 8 学时, 包括第 1 章到第 3 章。第 1 章重点介绍网站的基础知识、基本架构, 以及网页和标记语言的相关概念; 第 2 章重点介绍网站开发平台 Visual Studio 2008 的安装、IIS 的安装和基本设置以及 Framework 的基本构架; 第 3 章重点介绍 HTML 与 XHTML 文档的结构、标签的使用及它们的区别。

第二部分为布局篇, 这是本书的重点, 建议学时为 14 学时, 包括第 4 章到第 6 章。第 4 章重点介绍层叠样式表(CSS)的基本概念和用途, 以及层叠样式表的规则和常用的编写方法; 第 5 章深入讲解样式表的编写规则, 以实现更美观的页面效果; 第 6 章通过对具体实例的详细剖析来分析表格布局方式及其不足, 讲解 DIV+CSS 的网页布局技术及优势。

第三部分为 Visual Studio 篇, 建议学时为 12 学时, 包括第 7 章到第 9 章。第 7 章重点介绍 VS.NET 中主题的基本概念, 以及所涉及的主题文件、外观文件、样式表等内容; 第 8 章重点介绍 ASP.NET 母版页的基本概念和工作原理、母版页的创建和应用, 以及母版页的嵌套方法; 第 9 章重点介绍 JavaScript 语言的基本用法和 JavaScript 的对象的概念。

第四部分为案例篇, 建议学时为 6 学时, 即第 10 章。该章以“等级考试网上报名系统”为例介绍各页面设计与实现的完整过程。

## 本书特色

- 内容全面, 由浅入深 由网页基本概念开始, 逐步介绍 HTML、XHTML、CSS、JavaScript 等网页设计技术。
- 结合标准, 易于提高 本着一切从标准出发, 紧密结合 W3C 的 Web 标准来控制界面实现, 既可帮助读者牢固掌握基础知识, 又具有一定的理论高度, 有助于读者进一步提高。
- 技术先进, 符合潮流 摒弃表格布局技术, 采用目前市场主流的网页设计组合 DIV+CSS 进行详细讲解, 能帮助读者设计出具有一定水准的网页。基于微软 Visual Studio 2008 开发工具进行设计和开发, 更好地贴近技术发展的时代脉搏。
- 教学灵活, 强化理解 各章节通过实例详细演示制作过程, 并提供大量设计技巧、注意事项和常见问题解答。

- 综合应用, 实战演练 提供完整、翔实的综合案例, 详细解析各类网站的页面设计技术, 予以全程设计跟踪指导, 将作者的经验融汇于知识点中, 有益于读者更快上手和掌握。

本书作者均为教学第一线的骨干教师, 教学实践经验丰富。全书由王柯柯负责整体规划, 第1章由王柯柯编写, 第2章和第9章由黄丽丰编写, 第3章、第4章、第5章、第6章和第10章由崔英志、崔贯勋编写, 第7章由刘恒洋编写, 第8章由倪伟编写, 刘峰负责习题部分和全书的统稿。

由于笔者水平有限, 书中难免存在不足、遗漏之处, 请广大读者批评指正。同时, 为便于教学, 本书中引用了一些网站的界面图和一些书籍的封面, 在此一并表示感谢。

编 者

2011年1月

# 目 录

前言

## 第一部分 基础篇

|                                |    |
|--------------------------------|----|
| 第 1 章 网站概述 .....               | 1  |
| 1.1 WWW 概述 .....               | 1  |
| 1.1.1 WWW 与 Web .....          | 1  |
| 1.1.2 Web 技术基础 .....           | 1  |
| 1.2 网页技术基础 .....               | 4  |
| 1.2.1 静态网页技术概述 .....           | 4  |
| 1.2.2 动态网页基础 .....             | 5  |
| 1.2.3 动态网页开发技术简介 .....         | 6  |
| 1.3 Web 网页设计概述 .....           | 8  |
| 1.3.1 网页设计制作流程 .....           | 8  |
| 1.3.2 页面布局 .....               | 9  |
| 1.3.3 页面文档的构成 .....            | 12 |
| 1.3.4 标记语言概述 .....             | 15 |
| 1.4 本章小结 .....                 | 16 |
| 习题 .....                       | 16 |
| 第 2 章 Visual Studio 简介 .....   | 17 |
| 2.1 开发环境综述 .....               | 17 |
| 2.1.1 系统环境概述 .....             | 17 |
| 2.1.2 Framework 概述 .....       | 18 |
| 2.2 Visual Studio 的安装与应用 ..... | 18 |
| 2.2.1 Visual Studio 简介 .....   | 18 |
| 2.2.2 安装步骤简介 .....             | 20 |
| 2.2.3 Visual Studio 配置 .....   | 23 |
| 2.2.4 MSDN 的安装与应用 .....        | 24 |

|                                   |    |
|-----------------------------------|----|
| 2.2.5 在 Visual Studio 中建立网站 ..... | 25 |
| 2.2.6 解决方案的作用 .....               | 27 |
| 2.3 Internet 信息服务 (IIS) .....     | 27 |
| 2.3.1 IIS 简介 .....                | 27 |
| 2.3.2 IIS 的安装 .....               | 28 |
| 2.3.3 配置 IIS 的 ASP.NET 环境 .....   | 32 |
| 2.3.4 访问本地网站 .....                | 34 |
| 2.4 本章小结 .....                    | 34 |
| 习题 .....                          | 34 |
| 第 3 章 HTML 与 XHTML .....          | 35 |
| 3.1 HTML 标记语言 .....               | 35 |
| 3.1.1 初识 HTML .....               | 35 |
| 3.1.2 HTML 文件的基本结构 .....          | 36 |
| 3.1.3 文档格式定义的声明 .....             | 37 |
| 3.1.4 HTML 代码区域 .....             | 38 |
| 3.1.5 HTML 标签格式 .....             | 41 |
| 3.1.6 HTML 标签分类 .....             | 42 |
| 3.2 HTML 常用标签 .....               | 43 |
| 3.2.1 注释标签 .....                  | 43 |
| 3.2.2 标题标签 <h> .....              | 43 |
| 3.2.3 图像标签 <img> .....            | 43 |
| 3.2.4 超链接标签 <a> .....             | 45 |
| 3.2.5 空标签 <br> 和 <hr> .....       | 47 |
| 3.2.6 <span> 标签 .....             | 48 |
| 3.2.7 段落标签 <p> .....              | 48 |
| 3.2.8 层标签 <div> .....             | 49 |
| 3.2.9 表格标签 <table> .....          | 49 |

|                              |    |                          |    |
|------------------------------|----|--------------------------|----|
| 3.2.10 列表标签 <list> .....     | 50 | 4.3.3 块级标签属性 .....       | 83 |
| 3.2.11 表单标签 <form> .....     | 52 | 4.3.4 边框属性 .....         | 84 |
| 3.2.12 框架标签 <frame> .....    | 52 | 4.3.5 盒模型 .....          | 86 |
| 3.3 可扩展的超文本标记语言 .....        | 53 | 4.3.6 列表属性 .....         | 88 |
| 3.3.1 HTML 文档结构分析 .....      | 53 | 4.3.7 定位属性 .....         | 89 |
| 3.3.2 XHTML 文档结构 .....       | 55 | 4.3.8 style 属性 .....     | 91 |
| 3.3.3 XHTML 与 HTML 的差异 ..... | 56 | 4.4 CSS 选择器 .....        | 91 |
| 3.3.4 如何将 HTML 升级为 XHTML ... | 58 | 4.4.1 类型选择器 .....        | 91 |
| 3.4 实例：构建完整的 HTML 文档 .....   | 59 | 4.4.2 类选择器 .....         | 92 |
| 3.4.1 文档结构分析 .....           | 59 | 4.4.3 ID 选择器 .....       | 93 |
| 3.4.2 整体架构代码设计 .....         | 60 | 4.5 选择器的编写 .....         | 94 |
| 3.4.3 抬头区域 .....             | 64 | 4.5.1 组合选择器类型 .....      | 94 |
| 3.4.4 导航栏 .....              | 64 | 4.5.2 上下文选择器 .....       | 94 |
| 3.4.5 相关信息 .....             | 65 | 4.5.3 选择器的分组定义 .....     | 95 |
| 3.4.6 图书推荐 .....             | 66 | 4.6 伪类 .....             | 95 |
| 3.4.7 整体 HTML 文档代码 .....     | 67 | 4.6.1 锚伪类 .....          | 95 |
| 3.5 本章小结 .....               | 68 | 4.6.2 首字母和首行伪标签 .....    | 96 |
| 习题 .....                     | 68 | 4.7 实例：基本 CSS 样式设计 ..... | 96 |

## 第二部分 布局篇

|                                           |    |                         |     |
|-------------------------------------------|----|-------------------------|-----|
| 第 4 章 层叠样式表 .....                         | 69 | 4.7.1 CSS 常用操作三部曲 ..... | 96  |
| 4.1 层叠样式表简介 .....                         | 69 | 4.7.2 CSS 实例解析 .....    | 97  |
| 4.1.1 层叠样式表概述 .....                       | 69 | 4.8 本章小结 .....          | 107 |
| 4.1.2 层叠样式表的初步体验 .....                    | 70 | 习题 .....                | 108 |
| 4.1.3 Visual Studio 中层叠样式表的<br>使用方法 ..... | 72 | 第 5 章 CSS 样式表进阶 .....   | 109 |
| 4.1.4 CSS 编写方法 .....                      | 75 | 5.1 属性继承 .....          | 109 |
| 4.2 样式规则 .....                            | 76 | 5.1.1 树结构和继承 .....      | 109 |
| 4.3 CSS 属性 .....                          | 77 | 5.1.2 重写继承 .....        | 110 |
| 4.3.1 字体属性 .....                          | 77 | 5.1.3 不能继承的属性 .....     | 112 |
| 4.3.2 背景属性 .....                          | 80 | 5.1.4 压缩属性的方法 .....     | 113 |
|                                           |    | 5.1.5 浏览器与 CSS .....    | 115 |
|                                           |    | 5.2 选择器的使用策略 .....      | 116 |
|                                           |    | 5.2.1 选择器的命名规则 .....    | 116 |



|                               |     |                                          |     |
|-------------------------------|-----|------------------------------------------|-----|
| 5.2.2 class 属性的多值应用 .....     | 116 | 6.1.1 何谓布局技术 .....                       | 147 |
| 5.2.3 ID 与 class 选择器的权重 ..... | 117 | 6.1.2 网页布局标准概述 .....                     | 149 |
| 5.2.4 锚伪类的定义顺序 .....          | 118 | 6.2 表格布局技术 .....                         | 149 |
| 5.3 高级样式属性 .....              | 119 | 6.2.1 表格布局示例一：导航栏 .....                  | 150 |
| 5.3.1 高级属性选择器 .....           | 119 | 6.2.2 表格布局示例二：包括导航条<br>的网站页面 .....       | 152 |
| 5.3.2 高级上下文选择器 .....          | 120 | 6.2.3 表格布局的优势和不足 .....                   | 160 |
| 5.4 布局属性 .....                | 121 | 6.3 Web 标准概述 .....                       | 161 |
| 5.4.1 显示属性：display .....      | 121 | 6.3.1 无标准时代的 Web 开发 .....                | 161 |
| 5.4.2 浮动策略：float .....        | 124 | 6.3.2 Web 标准的概念 .....                    | 161 |
| 5.4.3 清除浮动：clear .....        | 125 | 6.3.3 符合 Web 标准的网页的构建 .....              | 163 |
| 5.4.4 定位技术：position .....     | 125 | 6.3.4 使用 Web 标准的优势 .....                 | 164 |
| 5.5 表格 .....                  | 130 | 6.4 DIV+CSS 布局 .....                     | 164 |
| 5.5.1 标题：caption .....        | 131 | 6.4.1 DIV+CSS 布局概述 .....                 | 164 |
| 5.5.2 单元格的合并 .....            | 132 | 6.4.2 浮动的使用 .....                        | 165 |
| 5.5.3 行组的使用 .....             | 133 | 6.4.3 清除浮动 .....                         | 167 |
| 5.5.4 列组的使用 .....             | 134 | 6.4.4 DIV+CSS 布局的优势 .....                | 168 |
| 5.5.5 其他属性 .....              | 136 | 6.4.5 常用 DIV+CSS 布局功能 .....              | 169 |
| 5.6 打印样式设置 .....              | 136 | 6.5 实例：DIV+CSS 布局 .....                  | 172 |
| 5.6.1 分页符 .....               | 136 | 6.5.1 DIV+CSS 布局示例一：导航菜单 .....           | 172 |
| 5.6.2 页面选择器和页边距 .....         | 137 | 6.5.2 DIV+CSS 布局示例二：包括<br>导航条的网站页面 ..... | 174 |
| 5.7 CSS 扩展：滤镜 .....           | 137 | 6.5.3 浮动技术的应用 .....                      | 174 |
| 5.7.1 滤镜概述 .....              | 137 | 6.5.4 清除浮动 .....                         | 175 |
| 5.7.2 常用滤镜 .....              | 138 | 6.5.5 IE 浏览器的兼容问题 .....                  | 176 |
| 5.8 实例：CSS 样式优化与进阶 .....      | 143 | 6.6 本章小结 .....                           | 176 |
| 5.8.1 浓缩 CSS 样式属性和值 .....     | 143 | 习题 .....                                 | 177 |
| 5.8.2 合并选择器的 CSS 样式 .....     | 144 |                                          |     |
| 5.8.3 布局 CSS 样式的使用 .....      | 145 |                                          |     |
| 5.9 本章小结 .....                | 146 |                                          |     |
| 习题 .....                      | 146 |                                          |     |
| 第 6 章 网页布局技术 .....            | 147 | 第三部分 Visual Studio 篇                     |     |
| 6.1 布局技术概述 .....              | 147 | 第 7 章 主题 .....                           | 179 |
|                               |     | 7.1 主题概述 .....                           | 179 |

|                              |     |                                            |     |
|------------------------------|-----|--------------------------------------------|-----|
| 7.1.1 为何引入主题 .....           | 179 | 8.4.2 Visual Studio 2008 中的母版页<br>嵌套 ..... | 206 |
| 7.1.2 主题的建立 .....            | 179 | 8.5 实例：母版页的设计 .....                        | 209 |
| 7.2 外观文件 .....               | 181 | 8.6 本章小结 .....                             | 211 |
| 7.2.1 外观控制的实现 .....          | 181 | 习题 .....                                   | 212 |
| 7.2.2 控件外观的定义方式 .....        | 182 | 第 9 章 JavaScript 程序设计 .....                | 213 |
| 7.2.3 特殊控件外观的定义 .....        | 183 | 9.1 JavaScript 概述 .....                    | 213 |
| 7.3 主题文件规划 .....             | 184 | 9.1.1 JavaScript 简介 .....                  | 213 |
| 7.3.1 整体规划 .....             | 184 | 9.1.2 JavaScript 入门案例 .....                | 214 |
| 7.3.2 外观文件规划 .....           | 184 | 9.2 JavaScript 语法 .....                    | 214 |
| 7.4 主题的应用 .....              | 185 | 9.2.1 语法基础 .....                           | 214 |
| 7.4.1 页面主题的设置 .....          | 185 | 9.2.2 常用变量 .....                           | 215 |
| 7.4.2 样式表主题的设置 .....         | 187 | 9.2.3 表达式与运算符 .....                        | 216 |
| 7.4.3 配置文件的设置 .....          | 189 | 9.2.4 程序设计 .....                           | 217 |
| 7.4.4 在指定文件夹下设置主题 .....      | 190 | 9.2.5 函数 .....                             | 218 |
| 7.4.5 EnableTheming 属性 ..... | 192 | 9.3 对象的概念 .....                            | 219 |
| 7.5 实例：主题的创建 .....           | 194 | 9.3.1 对象 .....                             | 219 |
| 7.6 本章小结 .....               | 195 | 9.3.2 属性 .....                             | 220 |
| 习题 .....                     | 195 | 9.3.3 方法 .....                             | 220 |
| 第 8 章 VS.NET 2008 母版解析 ...   | 196 | 9.4 JavaScript 对象 .....                    | 220 |
| 8.1 母版概述 .....               | 196 | 9.4.1 Window 对象 .....                      | 220 |
| 8.2 母版页的工作原理 .....           | 197 | 9.4.2 Document 对象 .....                    | 222 |
| 8.2.1 母版页 .....              | 197 | 9.4.3 Location 对象 .....                    | 224 |
| 8.2.2 占位符 .....              | 197 | 9.4.4 Navigator 对象 .....                   | 225 |
| 8.2.3 内容页 .....              | 198 | 9.4.5 Screen 对象 .....                      | 227 |
| 8.2.4 母版页的运行 .....           | 200 | 9.5 实例：JavaScript 的使用 .....                | 227 |
| 8.2.5 资源的路径设置 .....          | 201 | 9.6 本章小结 .....                             | 234 |
| 8.3 母版页的创建和应用 .....          | 202 | 习题 .....                                   | 234 |
| 8.3.1 母版页的创建 .....           | 202 | 第四部分 案 例 篇                                 |     |
| 8.3.2 母版页的应用 .....           | 204 | 第 10 章 等级考试网上报名系统 ...                      | 235 |
| 8.3.3 母版页与普通页面的区别 .....      | 205 | 10.1 网站设计分析 .....                          | 235 |
| 8.4 母版页嵌套 .....              | 206 |                                            |     |
| 8.4.1 母版页嵌套的意义 .....         | 206 |                                            |     |

## 第四部分 案例篇

第 10 章 等级考试网上报名系统 ... 235

10.1 网站设计分析 ..... 235



|                          |     |                              |     |
|--------------------------|-----|------------------------------|-----|
| 10.1.1 网站设计需求 .....      | 235 | 10.7 网页主体内容区域 .....          | 249 |
| 10.1.2 图纸分析 .....        | 236 | 10.7.1 HTML 文档设计 .....       | 249 |
| 10.1.3 页面表现分析 .....      | 237 | 10.7.2 CSS 样式设计 .....        | 250 |
| 10.2 HTML 文档结构分析 .....   | 239 | 10.8 用户信息区域 .....            | 251 |
| 10.2.1 设计图分层剖析 .....     | 239 | 10.8.1 HTML 文档设计 .....       | 251 |
| 10.2.2 整体文档结构分析 .....    | 240 | 10.8.2 CSS 样式设计 .....        | 251 |
| 10.3 网站文件规划与设计 .....     | 240 | 10.9 内容区域布局 .....            | 252 |
| 10.3.1 网站文件规划 .....      | 240 | 10.9.1 首页通知 HTML 文档设计 .....  | 252 |
| 10.3.2 母版文件 .....        | 241 | 10.9.2 首页通知 CSS 样式设计 .....   | 253 |
| 10.3.3 页面文件 .....        | 243 | 10.9.3 四六级报名 HTML 文档设计 ..... | 254 |
| 10.3.4 主题文件夹 .....       | 244 | 10.9.4 四六级报名 CSS 样式设计 .....  | 254 |
| 10.3.5 全局 CSS 样式设计 ..... | 245 | 10.10 页脚布局 .....             | 256 |
| 10.4 网站标题布局 .....        | 246 | 10.10.1 HTML 文档设计 .....      | 256 |
| 10.4.1 HTML 文档设计 .....   | 246 | 10.10.2 CSS 样式设计 .....       | 256 |
| 10.4.2 CSS 样式设计 .....    | 246 | 10.11 系统完善 .....             | 257 |
| 10.5 导航栏布局 .....         | 247 | 10.11.1 HTML 文档完善 .....      | 257 |
| 10.5.1 HTML 文档设计 .....   | 247 | 10.11.2 精简 CSS 样式代码 .....    | 258 |
| 10.5.2 CSS 样式设计 .....    | 247 | 10.11.3 适当使用上下文选择器 .....     | 259 |
| 10.6 快捷方式 .....          | 249 | 10.11.4 浏览器版本兼容 .....        | 260 |
| 10.6.1 HTML 文档设计 .....   | 249 | 10.12 本章小结 .....             | 260 |
| 10.6.2 CSS 样式设计 .....    | 249 | 参考文献 .....                   | 261 |

# 第一部分 基础篇

## 第1章 网站概述

### 【学习目标】

通过本章的学习，了解网站的基础知识，熟悉网站的基本架构，掌握网页和标记语言的相关概念，了解网页设计所涉及的开发工具。

### 【本章要点】

- Web 概述
- 动态网页技术
- 网页制作步骤
- 网页组成元素

## 1.1 WWW 概述

### 1.1.1 WWW 与 Web

WWW (World Wide Web) 亦称 3W 或 Web，中文译名为万维网、环球信息网，建立在客户机 / 服务器模型之上，是一个以 Internet 为基础的计算机网络。它允许用户在一台计算机上通过 Internet 存取另一台计算机上的信息，包括文字、图形、声音、动画、资料库以及各种软件等。WWW 不仅能访问 Web 服务器上的信息，而且还能访问 FTP、Telnet 等网络服务，因此，它已经成为 Internet 上应用最广、最受欢迎的信息检索服务系统，发展至今，它已成为 Internet 的代名词。

众所周知，Internet 上分布着大量计算机，它们各自扮演着不同的角色，但总体上分为两种：客户机和服务器。客户机就是我们通常所使用的计算机；服务器为一种高性能计算机，作为网络的节点，存储、处理网络上大量的数据、信息等，因此被称为网络的灵魂。根据所提供的服务不同，服务器分为 Web 服务器、邮件服务器、文件传输服务器、DNS 服务器等。其中，Web 服务器用于将本地的信息用超文本形式组织起来，方便用户在 Internet 上搜索和浏览，而 Web 或 WWW 实际上是由这些 Web 服务器构成的。换言之，Web 应用可以看成是 Internet 应用的一个子集。

### 1.1.2 Web 技术基础

谈及 Web 架构的技术，主要包括统一资源定位技术、超文本传输协议、超文本标记语言、浏览器及开发模式等，其中前三项是核心技术部分。这些技术构成了 Web 的基础，有助于 Web 实现其功能。

#### 1. 统一资源定位符

统一资源定位符 (Uniform Resource Locator, URL) 即通常所说的网站地址，它通过定义资源位置的抽象标识来定位网络资源，可对被定位后的资源进行更新、替换、查看属性等操作，是用于完整地描述 Internet 上网页和其他资源的地址的一种标识方法。

URL 的基本格式如下：

protocol :// hostname[:port] / path / [;parameters][?query]#fragment

各部分含义简介如下：

- protocol：所使用的协议名称，常用的协议有 http、ftp、file 等。
- hostname：主机名，存放资源的服务器的域名系统主机名或 IP 地址。
- port：端口号，是一个整数，省略时使用默认端口。各种传输协议都有默认的端口号，如 http 的默认端口为 80，为可选项。
- path：路径，表示主机上的一个目录或文件地址。
- parameters：参数，用于指定特殊参数，为可选项。
- query：查询，用于给动态网页传递参数，若有多个参数则用 & 符号隔开，每个参数的名和值用 = 符号隔开，为可选项。
- fragment：信息片断、字符串，用于指定网络资源中的片断。一个网页中有多个“网络”，可使用 fragment 直接定位到某一指定的名词解释。

例如：<http://cs.cqut.edu.cn/Notice/NoticeContent.aspx?NtcID=152>。其中，http 为协议名称；cs.cqut.edu.cn 为主机名；/Notice 为路径，即网页处于名为 Notice 的文件夹中；NoticeContent.aspx 为网页的名称；NtcID=152 表示参数 NtcID 的值为 152。

一般地，由于大多数网页内容是超文本传输协议文件，并不要求用户输入“http://”部分，而“80”也是超文本传输协议文件的常用端口号，也可以不用写明，因此，用户只要输入统一资源定位符的一部分（如 cs.cqut.edu.cn）即可。

## 2. 超文本传输协议

超文本传输协议（HyperText Transfer Protocol, HTTP）是负责在 Internet 上传输网页的协议，可以从 WWW 服务器传出超文本到本地浏览器，保障计算机正确、快速地在网络上传输超文本文档。

HTTP 协议是基于请求 / 响应方式的，采用的是客户机 / 服务器结构。一个客户机与服务器建立连接后，发送一个请求给服务器，HTTP 规则定义了如何正确解析请求；当服务器接到请求后，给予相应的响应信息，而 HTTP 规则也定义了如何正确解析应答信息，但它并不定义网络如何建立连接、管理及信息如何在网络上发送等。连接情况如图 1-1 所示。

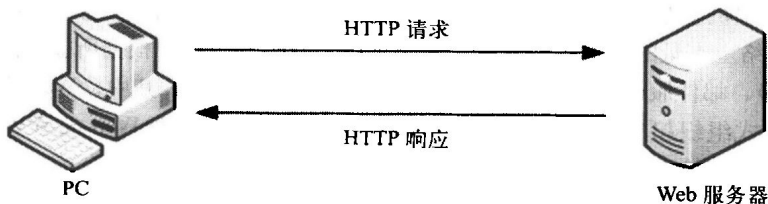


图 1-1 HTTP 协议

HTTP 协议的主要特点简述如下。

- 通信简单：当客户端向服务器端请求服务时，只需传送请求方法和路径即可，常用的请求方法包括 GET、HEAD、POST 等。
- 速度快：由于 HTTP 协议简单，使得 HTTP 服务器的程序规模小，因而通信速度较快。
- 无连接协议：其含义是限制每次连接只处理一个请求，即服务器处理完客户的请求，并收到客户的应答后，方断开连接，从而节省传输时间。
- 无状态协议：无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要

前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。相反，在服务器不需要先前信息时它的应答就较快。

### 3. 超文本标记语言

超文本标记语言（HyperText Markup Language，HTML）是一种描述文本结构的标记语言，它是整个 Web 技术的基础，主要功能是在 Web 上发布信息。任何工具开发的 Web 页面最终都要转换为 HTML 语言显示在客户端的浏览器上，网页上的文字、图片、声音、视频等多种媒体均是通过它连接起来的，因此，了解并掌握 HTML 语法对精通基于 Web 的程序设计至关重要。

HTML 的内容将在第 3 章中加以详述。

### 4. 浏览器

浏览器是显示网页服务器内的文字、图片、视频等多种媒体文件并让用户与这些文件互动的一种软件，它主要通过 HTTP 协议与网页服务器交互并获取网页，这些网页由 URL 指定。目前常用的浏览器有 Microsoft 的 IE（Internet Explorer）浏览器，以及 Firefox、Safari、Opera 等浏览器。

### 5. Web 开发模式

在当前多元化应用需求的推动下，Web 应用技术与应用程序的开发技术虽各具特色，但日益融合，从最初的单机模式发展到 C/S（Client/Server，客户机/服务器）模式，再到流行的 B/S（Browser/Server，浏览器/服务器）模式。

在 B/S 架构下，其工作界面通过 IE 或其他浏览器实现，即用户通过浏览器获取信息或提交查询等需求时，主要业务逻辑都在服务器端实现，因此，B/S 架构的系统不需要安装客户端软件，而系统升级或维护时只需更新服务器端软件，从而简化了客户端载荷（通常称为“瘦”客户端），减轻了系统维护与升级的成本和工作量，降低了用户的总体成本。目前，基于 Web 的 B/S 模式已成为主流的 Web 应用解决方案，即 B/S 三层体系结构，它将应用的功能分为表示层、应用层和数据层。表示层就是展现给用户的界面，即用户在使用一个系统的时候实现所见即所得；应用层是针对具体问题的操作，也可以说是对数据层的操作和对数据业务逻辑的处理；数据层直接操作数据库，针对数据完成增加、删除、修改、查找等操作。通常情况下，将表示层配置在客户机中，将应用层和数据层分别置于不同的服务器中，称之为 Web 服务器和数据服务器。该模式灵活性高，能够适应负荷的变化。B/S 三层体系结构如图 1-2 所示。

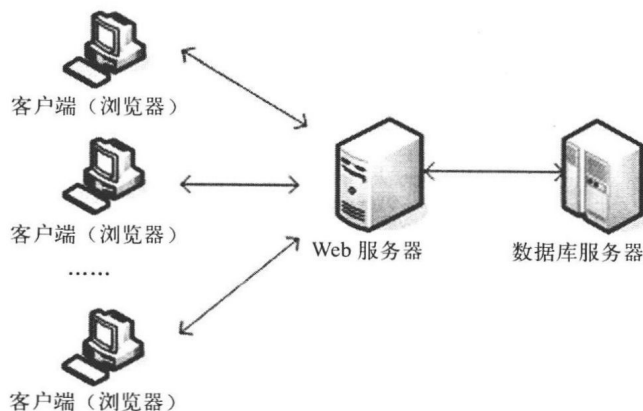


图 1-2 B/S 三层体系结构

国内外一些大型知名网站采用了更先进的结构模式，即多层 B/S 动态模式，其结构是基于图 1-2 所示的传统模式，在 Web 服务器和数据库服务器之间增加了一层应用服务器，将复杂的企业逻辑及数据库的连接服务等封装到该中间层上，以减轻 Web 服务器的负担，并兼具负载平衡与容错功能。

该结构属于典型的分布式 Web 应用模式，如图 1-3 所示。

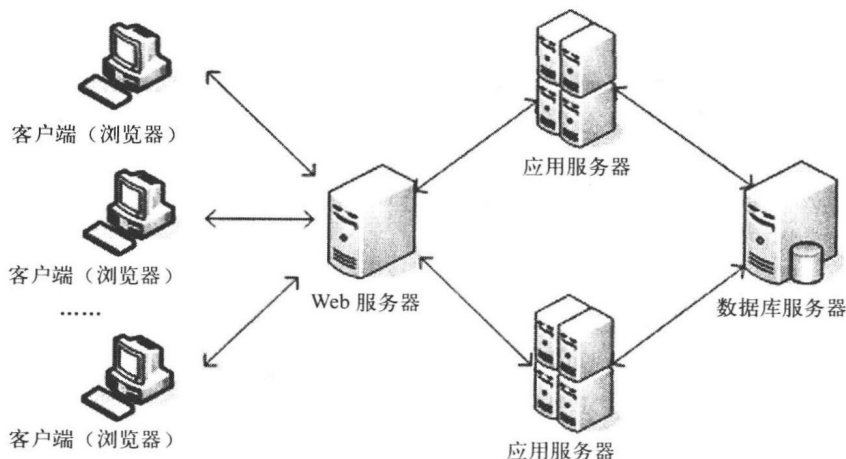


图 1-3 B/S 多层体系结构

## 1.2 网页技术基础

### 1.2.1 静态网页技术概述

静态网页是 Web 程序最早的表现形式，是存储于服务器的文件，其内容可包括文字、图像、声音等多种媒体元素，页面间可通过超级链接进行关联，但所有内容在生成该文档后始终保持不变。静态页面只能一个页面对应一个内容，即一对一的关系，用户进行的任何访问操作都仅限于让服务器传送数据给请求者，并不做脚本计算及读取后台数据库的工作，数据库、后台系统与前台是分开的。其优势是访问速度快，安全隐患低。典型的静态页面以 html 为后缀名，如图 1-4 所示。

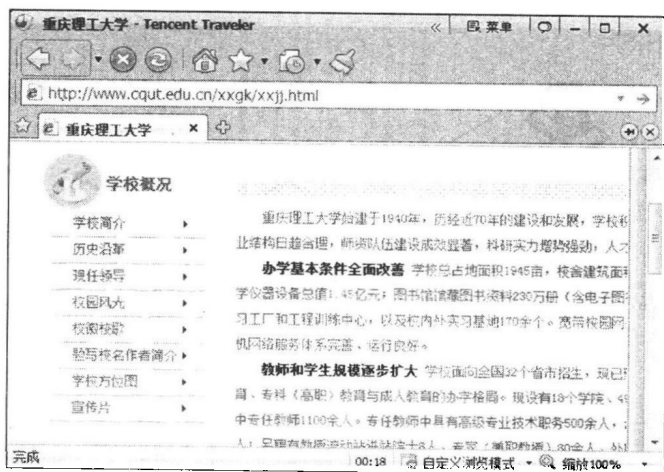


图 1-4 静态页面示例

静态网页的优点是速度快、跨平台、跨服务器；缺点是请求信息资源的用户不能和页面进行交互，页面的内容也不会因用户的种种操作而发生改变，只能人工制作、更新和发布，具有较大的局限性。

## 1.2.2 动态网页基础

### 1. 动态网页概述

动态网页的“动态”是指交互性，即网页等根据访问者不同的请求或不同的时间而显示不同的内容，弥补了 HTML 静态网页难以适应信息频繁更新以及交互的不足。

动态网页是最常用的网站表现形式，它是在浏览器访问 Web 服务器时由服务器创建的。根据先前所制定好的程序页面，当用户向服务器发出请求时，Web 服务器创建动态文档，返回给用户，因此不同的用户请求会返回不同的结果，是一对多的关系，可实现资源的最大利用和服务物理资源的最大节省。若需要改变站点的风格，只需要重新设计前台访问的页面即可。典型的动态页面是以 asp、aspx 等作为后缀名。如图 1-5 所示为一个动态页面的示例。

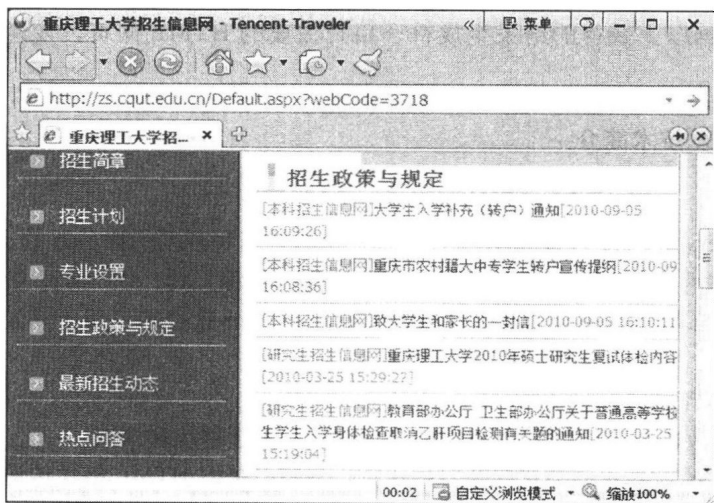


图 1-5 动态页面示例

动态网页的优点是效率高、更新快、移植性强，能够快速达到所见即所得的目的；缺点是由于需要与数据库的不断通信才能实现较高的效率，而频繁地读取数据库会导致服务器花大量的时间来计算，当访问量达到一定的数量后，会导致效率成倍下降。如果有人恶意攻击程序，激活了隐藏的 BUG，将会构成一定的安全隐患，甚至导致整个网站瘫痪。

动态网页的一般特点简要归纳如下：

- 动态网页以数据库技术为基础，能够大大降低网站维护的工作量。
- 采用动态网页技术的网站可以实现更多的功能，如用户注册、用户登录、在线调查、用户管理、订单管理等。
- 动态网页实际上并不是独立存在于服务器上的网页文件，只有当用户请求时服务器才返回一个完整的网页。

一般地，动态网页通常分为客户端动态网页和服务器端动态网页，下面将详细介绍。

### 2. 客户端动态网页

采用客户端动态网页模式时，由附加到浏览器上的模块完成创建动态网页的全部工作。其工作过程是，HTML 代码与含有指令的文件一起传送到浏览器，当用户请求时，浏览器就根据请求通过这些指令动态生成 HTML 页面，然后返回浏览器。

目前客户端动态网页技术基本不被采用，其原因在于：

- 需要下载客户端软件，且比较耗时。

- 各种浏览器以不同的方式解释指令，因此不能保证其他浏览器（如 FireFox）也能够正确理解 IE 浏览器能理解的指令。
- 编写使用服务器资源的客户端代码时会出现安全性问题，若代码是在客户端解释执行的，则客户端脚本代码将会完全公开。

### 3. 服务器端动态网页

采用服务器端动态网页模式时，HTML 代码与操作指令存储于 Web 服务器中，当用户发出请求时，指令在 Web 服务器中执行处理后返回浏览器。与客户端动态网页模式不同，服务器端动态网页模式仅有 HTML 代码传送到浏览器而不包括被执行的指令。

其工作过程是，用户发出页面请求后，Web 服务器响应 HTTP 请求，并解释或编译被申请的文件，最后执行它；若需要访问数据库，则与后台数据库建立连接，由数据库访问组件执行相应操作；最后根据以上获得的结果生成符合用户需要的 HTML 网页，发送至客户端来响应用户的需求。

## 1.2.3 动态网页开发技术简介

### 1. ASP

ASP（Active Server Pages，动态服务器页面）是微软公司开发的服务器端运行的脚本平台，是一种创建和运行动态网页或 Web 应用程序的简单、方便的编程工具。ASP 网页可以包含 HTML 标记、普通文本、脚本命令以及 COM 组件等，常用的 Dreamweaver 软件即可用于开发 ASP 网页，其文件格式为 .asp。

ASP 网页具有以下特点：

- ASP 文件是包含在 HTML 代码所组成的文件中的，易于修改和测试。
- 服务器上的 ASP 解释程序会在服务器端执行，并将结果以 HTML 格式传送到客户端浏览器上，因此使用各种浏览器都可以正常浏览 ASP 所产生的网页。
- ASP 提供了一些内置对象，使用这些对象可以使服务器端脚本功能更强，如 Request、Response、Server、Application、Session 等对象。
- ASP 可以使用服务器端 ActiveX 组件来扩充功能，执行各种用户请求任务，如存取数据库、发送 E-mail 或访问文件系统等。
- 由于服务器是将 ASP 程序执行的结果以 HTML 格式传回客户端浏览器，因此浏览者不会看到 ASP 所编写的源码，这样可以保障 ASP 程序代码的安全。

### 2. ASP.NET

ASP.NET 并不是 ASP 的简单升级，而是更快速、高效、灵活以及更好地支持交互式 Web 应用程序的新一代技术，它继承了 ASP 的优点，融入了 Java、VB 等特性，建立了一个全新的开发平台，为动态网站的开发开辟了一个新的领域。

ASP.NET 从 1.x 发展到 2.0，再到如今的 3.5，发展如此迅速，应用如此广泛，可见其在开发动态网站方面是极具优势的。ASP.NET 将程序在服务器端首次运行时进行编译，而后直接运行，其执行效率比逐条解释语句要高很多，执行效率大幅提高；ASP.NET 开发的是基于通用语言编译运行的程序，这样程序能够运行在 Web 应用软件研发者的全部平台上，通用语言的基本库、消息机制、数据接口的处理都能无缝整合到 ASP.NET 的 Web 应用中。目前已支持的有 C#、VB、C++、JScript 等多种语言，并且简单易学，能使初学者在短时间内快速上手，开发基于 Web 的应用程序。本书实例就是基于 ASP.NET 3.5，以 Visual Studio 2008 为开发环境加以讲解的，从中可以体会它的优势和便捷。



### 3. JSP

JSP (Java Server Pages) 是一种能用来快速创建动态网页的技术。使用 Java 编程语言编写类 XML 的 tags 和 scriptlets (一种使用 Java 语言编写的脚本代码) 来封装产生动态网页的处理逻辑, 通过 tags 和 scriptlets 访问存在于服务器端的资源的应用逻辑, 将各种格式的标签 (HTML 或 XML) 直接传递给相应的页面, 从而实现页面逻辑与页面显示的分离, 并支持可重用的基于组件的设计, 使基于 Web 的应用程序的开发变得迅速和容易, 其文件格式为 .jsp。

JSP 页面由 HTML 代码和嵌入其中的 Java 代码组成。服务器在页面被客户端请求以后对这些 Java 代码进行处理, 然后将生成的 HTML 页面返回给客户端的浏览器。用 JSP 开发的 Web 应用是跨平台的, 既能在 Linux 下运行, 也能在 Windows 等其他操作系统上运行。

JSP 的主要特点概括如下:

- 良好的移植性。JSP 使用 Java 语言作为生成动态内容的编程语言。由于 Java 语言具有很强的适用性和移植性, 故无需改动 JSP 程序即可在各种平台上运行。
- 较强的实用性。JSP 继承了 Servlet (Java 语言编写的 Java 类) 的优点, 使动态数据处理和输出内容分开, 方便 Web 应用的维护和修改。
- 较高的可重用性。大多数 JSP 页面利用可重用的、跨平台的组件 (JavaBeans 或者 Enterprise JavaBeans 组件) 来执行应用程序所要求的更为复杂的处理, 使开发人员能够交换执行普通操作的组件, 并与其他使用者共享这些组件。

### 4. Java

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称, 由 Java 虚拟机 (Java Virtual Machine) 和 Java 应用编程接口 (Application Programming Interface, API) 构成。Java 平台已经嵌入了几乎所有的操作系统, 因此, Java 程序只编译一次, 就可以在各种系统中运行。

Java 语言是一种使用广泛的网络编程语言, 它简单、面向对象、分布式, 具有健壮性、安全性、可移植性等诸多优点, 性能优异, 并配备了丰富的类库, 使编程人员可以方便地建立自己的系统。

Java 的主要特点简述如下:

- 面向对象。Java 语言提供类、接口和继承等, 支持类之间的单继承、接口之间的多继承及类与接口之间的实现机制。
- 可移植性。这种可移植性来源于体系结构中立性, 另外, Java 还严格规定了各个基本数据类型的长度。Java 系统本身也具有很强的可移植性, Java 编译器是用 Java 实现的, Java 的运行环境是用 ANSI C 实现的。
- 语法简单。Java 语言的语法与 C 语言和 C++ 语言很接近, 使得大多数程序员很容易学习和使用。它丢弃了 C++ 中很少使用的、很难理解的部分特性, 如操作符重载、多继承、自动的强制类型转换等, 并且不使用指针, 提供了自动的垃圾收集功能, 使程序员不必考虑内存管理。
- 分布式。Java 语言支持 Internet 应用的开发, 在基本的 Java 应用编程接口中有一个网络应用编程接口, 它提供了用于网络应用编程的类库, 包括 URL、URLConnection、Socket、ServerSocket 等。
- 健壮性。Java 的强类型机制、异常处理、垃圾的自动收集等是 Java 程序健壮性的重要保证。
- 安全性。Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外, 它对通过网络下载类具有一个安全防范机制, 如分配不同的名字空间以防替代本地的同名类、字节代码检查, 并提供安全管理机制让 Java 应用设置安全哨兵。
- 动态性。Java 程序需要的类能够动态地载入到运行环境, 也可以通过网络来载入所需要的类, 有利于软件的升级。

## 5. PHP

PHP (Personal Hypertext Preprocessor, 超级文本预处理器) 是一种在服务器端执行的嵌入 HTML 文档的脚本语言, 它独特的语法混合了 C、Java、Perl 以及 PHP 自创的语法, 可以比 CGI 或者 Perl 更快速地执行动态网页。PHP 非常简单, 实用性强, 源码开放, 完全免费, 很适合初学者, 故在 ASP 流行之前 PHP 的应用十分广泛。

PHP 的主要特点简述如下:

- 基于服务器端。由于 PHP 是运行在服务器端的脚本, 可以运行在 UNIX、Linux、Windows 下。
- 效率高。PHP 消耗的系统资源相当少, 因此执行效率高。
- 易于掌握。PHP 学习过程简单, 只需熟悉基本的语法和特点就能上手编码, 其语法也与 C、Perl 等类似, 对于熟悉上述语言的人来说能更快掌握。
- 面向对象。在 PHP4、PHP5 中对面向对象方面进行了很大的改进, 提供类和对象, 也支持构造器、提取类等。
- 数据库连接。PHP 可以通过编译形成连接数据库的函数来建立与数据库的连接。

## 1.3 Web 网页设计概述

### 1.3.1 网页设计制作流程

网页是构成网站的主体。我们知道, 一本书由若干页组成, 如果把网站看成一本书, 那么网页就是书的每一页, 因此, 制作网站的核心就是设计和制作每一个网页。网页设计的一般流程简述如下。

#### 1. 明确目标

在接到网站设计任务以后, 首先应该确定该网站的类型, 然后充分理解客户对网页设计所提出的目标和要求, 以满足客户的需求。从网站的形态和规模来看, 一般可分为以下三种:

- 综合类。综合类网站通常为门户网站, 如新浪、搜狐、人民网等。这类网站具有信息海量、受众群庞大、点击量巨大等诸多特点, 因此, 该类网站在制作时应具备良好的页面分割、合理的网页结构, 且页面打开速度快、具有较强的亲和力。
- 企业类。企业类网站通常为信息与企业形象相结合的网站, 如大型企事业单位、公司、学校等的网站。这类网站具有一定的资讯内容, 同时要承担传递企业形象的任务, 因此, 该类网站在网页设计和功能要求上通常较高, 以便兼顾传达信息和树立企业形象的任务。
- 其他类。除此以外还有诸如音乐、游戏、论坛等类型的网站, 其网页的设计应符合网站主题, 风格可根据需要灵活多变。

#### 2. 设计草图

目标确定后即对网站的大致内容有了初步的选择和把握, 然后就可理清思路, 列出详细的栏目列表, 以及各栏目的子目录, 接下来就可以设计草图、页面布局等。通常情况下, 可通过手绘勾勒出页面的大概轮廓, 自由发挥, 多绘制几张草图样板, 再从中选择较优者进行后期加工完善。

在此基础之上, 遵循突出重点、考虑平衡、兼顾协调的原则, 将需要放置的功能模块合理安排到页面上, 将网站 Logo、主菜单等重要的模块放在显眼、突出之处, 然后再放置其他次要模块。

#### 3. 实现蓝图

有了经过初步布局后的草案后, 接着就可以应用常用的网页设计软件 (如 Visual Studio 2005 / 2008、Dreamweaver 等) 实现设计蓝图。虽然在设计草图时大体框架已基本确定, 但在制作的过程中还需要不断改进、不断思考, 激发灵感和创造力, 以获取更优秀的创意。