

普通高等学校“十三五”规划教材

Web网站设计 与开发教程 (HTML5、JSP版) (第二版)

温浩宇 李慧 编著



西安电子科技大学出版社
<http://www.xduph.com>

普通高等学校“十三五”规划教材

Web 网站设计与开发教程 (HTML5、JSP 版)

(第二版)

温浩宇 李慧 编著



西安电子科技大学出版社

内容简介

本书系统地介绍了基于 HTML5 与 JSP 进行网站开发所需的基础知识和技术,用简洁、清晰的语言讲解了网站开发的前台技术和后台技术。

本书共分 13 章,内容包括 Web 技术概述、HTML 元素、层叠样式表 CSS、脚本语言 JavaScript、XML 技术基础、开发运行环境、Servlet 编程、JSP 基本语法及基本技术、JSP 内置对象、应用 JavaBean 技术、基于 JSP 的数据库应用开发、新闻发布网站的设计以及其他网站框架及开发技术。为方便读者学习,每一章都给出了学习提示,可帮助读者理解本章的内容及其在网站开发过程中的作用。本书第 1~11 章均给出了课后思考题,便于读者考查学习情况;第 12 章给出一个完整的新闻发布与管理网站开发过程及相应代码,通过案例驱动的教学方法,帮助读者构建完整的知识结构。

本书可作为高等学校计算机、信息管理、电子商务等相关专业的网站设计与开发的实训教程,也适合广大软件开发爱好者自学使用。

图书在版编目(CIP)数据

Web 网站设计与开发教程:HTML5、JSP 版/温浩宇,李慧编著. —2 版. —西安:西安电子科技大学出版社,2018.4

ISBN 978-7-5606-4747-0

I. ① W… II. ① 温… ② 李… III. ① 超文本标记语言—程序设计—教材 ② JAVA 语言—网页制作工具—教材 IV. ① TP312.8 ② TP393.092

中国版本图书馆 CIP 数据核字(2018)第 064998 号

策划编辑 戚文艳

责任编辑 戚文艳

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 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2018 年 4 月第 2 版 2018 年 4 月第 4 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 18.875

字 数 447 千字

印 数 1~3000 册

定 价 42.00 元

ISBN 978-7-5606-4747-0 / TP

XDUP 5039002-4

***** 如有印装问题可调换 *****

前 言

随着互联网、云计算、物联网等热门技术的不断成熟，越来越多的软件开发者希望对网站开发技术进行深入系统的学习，进而成为技术市场上炙手可热的人才。但对于有兴趣学习 Web 技术的开发者而言，不断涌现（有很多也是昙花一现）的开发工具、程序设计语言和设计框架让人眼花缭乱、无所适从，致使很多网站开发的初学者徘徊在 JSP、PHP、ASP.NET 等技术之间，不知该从何学起。

本书的作者从事网站开发和相关教学工作十多年，目睹了大量 Web 技术像走马灯一样快速流行、瞬间过时。在指导本科生和研究生的过程中，作者一直在思考究竟什么样的学习路径可以让开发者快速、系统地掌握 Web 开发技术。通过大量实践，作者比较了 JSP、PHP、ASP.NET 等多种开发技术，逐渐总结了一些规律。

PHP 技术是网站开发者最热衷的技术之一，它的特点就是“快”，学起来快，开发过程也快。如此说来，PHP 技术也许应该是网站开发的不二法门，但快是有代价的，比如语言过于灵活，对代码的约束较少，程序员容易在不知不觉中写出错误代码。当然，在小团队进行小型网站的开发过程中，这些缺点可能并不会带来严重的问题。但对于初学者而言，一个太容易上手的技术，往往会让他们忽略了系统底层的运行机理。应该说，这本身并不是 PHP 的错，而是由于初学者在使用 PHP 开发小型网站时不必充分理解 Web 系统的运行机理。所以，“易学易用”本身就是一把双刃剑。

JSP 技术需要开发者首先掌握 Java 程序设计技术，然后从最简单的“Hello World”程序开始学起。由于基本的 JSP 网站开发需要程序员书写大量代码，所以开发效率较低，学习过程较长。但这一缺点带来的好处是：开发者可以逐渐理解网站的运行机理，包括 Web 服务器的工作方式和浏览器对 HTML 的解析方式，进而构建出完整的网站开发知识结构。当然，在很好地掌握基本的 JSP 开发技术之后，可以进一步学习使用多种基于 Java 的开发框架，从而有效地提高开发效率。

ASP.NET 技术沿袭微软公司开发工具的一贯风格，基于 ASP.NET 的网站开发提供了“拖拖拽拽”的可视化开发工具，这种开发模式不仅让初学者可以很快地掌握 Web 开发的大多数技术，而且也会大大提高资深程序员的工作效率。不得不承认：开发的速度是软件公司生存的重要基础之一。但与 PHP 的情况类似，易于“掌握”的开发技术，往往会让初学者忽略了本该了解的可视化背后的故事。我们看到许多自认为已经充分掌握

ASP.NET 网站设计技术的开发者在被问到“session 的运行机理是什么”或者“cookie 是如何保存客户数据的”这类问题时，居然连问题都听不懂，这多么尴尬啊。再次说明：这不是 ASP.NET 的错，而是学习的“快车道”让初学者太快体会到成功的喜悦，以至于根本无暇欣赏沿途的风景。

总结上述分析，作者建议初学者从 JSP 学起，在充分掌握 Web 运行机理和开发思路的基础上，再学习 ASP.NET、PHP 或基于 Java 的开发框架。扎实的基础将让开发者走得更远，且越走越快。

为了给读者提供进一步学习的思路，本书提供了多个完整的开发实例，并对开发大型而复杂的网站时需要的设计框架进行了介绍。

按照分工，温浩宇和李慧编写了本书的主要内容，许多研究生也参与了编写：朱艳洁编写了第 2 章和相关的实例；程栋编写了第 4 章的实例；田亚丹、奚园园、马宇鑫、刘芬芳、孙策、李京京、陈玉兆、梁承希、杨璐、姚香秀、刘巧莉、王珺泽、张颖、王若雨等都参与了程序实例的编写、修改和测试。值得一提的是，许多参与本书编写的优秀研究生毕业后已进入著名的 IT 公司工作，他们的成功经历也激励着许多同学开始认真、系统地学习 Web 网站开发技术。

衷心感谢在本书编写过程中提供帮助的同事和学生！衷心感谢西安电子科技大学出版社的相关人员对本书的大力支持！

由于 Web 网站开发技术日新月异，加之作者水平有限，书中难免存在错误和不足之处，敬请读者批评指正。

编者

2017 年 7 月

目 录

第1章 Web 技术概述.....	1	2.7 编辑元素.....	31
1.1 Web 系统简介.....	1	2.8 表单元素.....	32
1.2 B/S 结构和 Web 应用程序.....	2	2.8.1 表单元素概览.....	32
1.3 HTML 简介.....	4	2.8.2 <form>元素.....	33
1.4 HTML 文档结构.....	5	2.8.3 <input>元素.....	33
思考题.....	6	2.8.4 <select>元素.....	36
第2章 HTML 元素.....	7	2.8.5 <progress><meter>元素.....	38
2.1 文本元素.....	7	2.8.6 <fieldset><legend>元素.....	38
2.1.1 文本元素概览.....	7	2.8.7 <script><noscript><template> <canvas>元素.....	39
2.1.2 <small><s><sub><sup> <i><u><mark>元素.....	9	2.9 头部元素.....	40
2.1.3 <ruby><rb><rt><rp><rtc>元素.....	9	2.9.1 头部元素概览.....	40
2.1.4 元素.....	10	2.9.2 <head><title><base>元素.....	40
2.2 群组元素.....	10	2.9.3 <link><style>元素.....	41
2.2.1 群组元素概览.....	10	2.9.4 <meta>元素.....	41
2.2.2 <p><hr><pre><blockquote>元素.....	11	2.10 HTML 中的颜色设置.....	41
2.2.3 <dl><dt><dd>元素.....	12	2.11 绝对路径与相对路径.....	42
2.2.4 <div>元素.....	14	思考题.....	43
2.3 超链接元素.....	15	第3章 层叠样式表 CSS.....	44
2.4 表格元素.....	16	3.1 CSS 简介.....	44
2.5 内嵌元素.....	19	3.2 选择符.....	45
2.5.1 内嵌元素概览.....	19	3.3 CSS 的层叠性与优先次序.....	48
2.5.2 元素.....	20	3.4 常用属性及其应用实例.....	49
2.5.3 <map><area>元素.....	20	3.4.1 CSS 文本属性.....	49
2.5.4 <iframe>元素.....	21	3.4.2 CSS 表格属性.....	51
2.5.5 <video><audio><source>元素.....	22	3.5 CSS 盒子模型和网页布局方式.....	53
2.5.6 MathML 系列元素.....	24	3.5.1 盒子模型简介.....	53
2.5.7 SVG 系列元素.....	25	3.5.2 CSS 的定位功能.....	55
2.5.8 <object><param>元素.....	26	3.5.3 CSS 的定位方式.....	56
2.6 结构元素.....	28	3.5.4 网页布局方式实例.....	57
2.6.1 结构元素概览.....	28	思考题.....	61
2.6.2 <body><h1><h2><h3><h4><h5> <h6>元素.....	29	第4章 脚本语言 JavaScript.....	62
2.6.3 <article><section><nav><aside> <header><footer>元素.....	30	4.1 JavaScript 简介.....	62
		4.2 JavaScript 的基本语法.....	63

4.2.1 常量和变量	63	第 6 章 开发运行环境	108
4.2.2 数据类型	64	6.1 JDK 的安装与配置	108
4.2.3 表达式和运算符	66	6.1.1 JDK 简介	108
4.2.4 循环语句	68	6.1.2 JDK 安装	109
4.2.5 条件语句	69	6.2 Tomcat 的安装及配置	113
4.2.6 函数	71	6.2.1 Tomcat 简介	113
4.3 JavaScript 的面向对象特性	72	6.2.2 Tomcat 的安装	114
4.3.1 类和对象	72	6.2.3 Tomcat 的配置	118
4.3.2 JavaScript 的内置对象	74	6.3 MyEclipse 的安装及配置	121
4.3.3 异常处理机制	77	6.4 Web 应用和 web.xml 文件	125
4.4 JavaScript 在浏览器中的应用	77	6.4.1 Web 应用概述	125
4.4.1 浏览器对象	77	6.4.2 web.xml 文件详解	126
4.4.3 JavaScript 在 DOM 中的 应用方式	81	6.4.3 部署 Web 应用	128
4.4.3 事件驱动与界面交互	87	思考题	129
4.5 JavaScript 在 HTML5 中的应用	88	第 7 章 Servlet 编程	130
4.5.1 HTML5 绘图的应用	88	7.1 从 CGI 到 Servlet	130
4.5.2 本地存储	91	7.1.1 CGI 简介	130
4.6 常用的 JavaScript 框架	92	7.1.2 ISAPI 与 NSAPI 简介	131
思考题	93	7.1.3 Servlet 简介	132
第 5 章 XML 技术基础	94	7.2 Servlet 生命周期	134
5.1 XML 简介	94	7.3 Servlet 的开发实例	135
5.2 XML 语法与结构	96	7.3.1 手动实现 Servlet 开发	136
5.2.1 处理指令	96	7.3.2 采用 MyEclipse 实现 Servlet 开发	140
5.2.2 XML 元素	96	7.3.3 测试 Servlet	141
5.2.3 元素属性	97	7.3.4 Servlet 初始化参数	142
5.3 命名空间	98	思考题	144
5.4 文档类型定义与校验	98	第 8 章 JSP 基本语法及基本技术	145
5.4.1 文档类型定义	98	8.1 JSP 技术概况	145
5.4.2 XML 架构	99	8.2 JSP 基本语法	148
5.5 XML 文档样式转换	100	8.2.1 程序片	149
5.5.1 在 XML 中使用 CSS	100	8.2.2 表达式	149
5.5.2 在 XML 中使用 XSL	102	8.2.3 声明	150
5.6 XML 文档的解析	103	8.2.4 JSP 中的注释	154
5.6.1 DOM 解析器	103	8.3 JSP 指令	154
5.6.2 SAX 解析器	105	8.3.1 page 指令	154
思考题	106	8.3.2 include 指令	157

8.3.3 taglib 指令	159	第 11 章 基于 JSP 的数据库应用开发	211
8.4 JSP 动作	160	11.1 JDBC 接口	211
8.4.1 include 动作	161	11.1.1 JDBC 的功能	211
8.4.2 forward 动作	163	11.1.2 JDBC 驱动分类	212
8.4.3 plugin 动作	164	11.1.3 连接数据库	218
8.4.4 param 动作	165	11.2 数据查询	223
思考题	165	11.3 数据添加	231
第 9 章 JSP 内置对象	166	11.4 数据删除	234
9.1 内置对象概述	166	11.5 数据更新	237
9.2 out 对象	168	11.6 数据库连接池	240
9.3 request 对象	170	11.6.1 数据库连接池概述	240
9.3.1 getParameter 方法示例	171	11.6.2 数据库连接池的配置	242
9.3.2 解决 request 中文乱码问题	173	思考题	244
9.3.3 request 范围内的属性设置与 获取	174	第 12 章 新闻发布网站的设计	245
9.3.4 request 执行 forward 或 include 动作	175	12.1 需求分析	245
9.4 response 对象	176	12.1.1 网站基本功能描述	245
9.4.1 setContentType 方法	176	12.1.2 总体设计	246
9.4.2 response 重定向	178	12.2 ORM 技术应用	247
9.4.3 增加 Cookie	180	12.2.1 ORM 技术简介	247
9.5 session 对象	183	12.2.2 数据库表的设计(R)	248
9.6 application 对象	186	12.2.3 数据对象的设计(O)	249
9.7 exception 对象	188	12.2.4 关系与对象的匹配(M)	251
9.8 其他内置对象	190	12.3 MVC 框架模式应用	257
9.8.1 page 对象	190	12.3.1 MVC 简介	257
9.8.2 pageContext 对象	191	12.3.2 网站 MVC 结构设计	259
9.8.3 config 对象	193	12.3.3 MVC 实现解析	260
思考题	195	12.4 实现网页静态化	266
第 10 章 应用 JavaBean 技术	196	12.5 应用 AJAX 实现快捷交互	270
10.1 JavaBean 构造方法	196	12.5.1 AJAX 简介	270
10.2 useBean 动作	198	12.5.2 AJAX 程序实现	272
10.2.1 JavaBean 的有效范围	198	12.6 应用 Servlet 实现验证码	275
10.2.2 useBean 的工作过程	199	12.6.1 验证码设计分析	275
10.2.3 设置和获取 JavaBean 属性值	199	12.6.2 Servlet 程序实现	276
10.3 应用 JavaBean 的开发实例	201	第 13 章 其他网站框架及开发技术	281
思考题	210	13.1 基于 Java 的网站设计框架	281
		13.1.1 应用框架的优点	281

13.1.2	Struts 框架	282
13.1.3	Spring 框架	283
13.1.4	Hibernate 框架	284
13.2	搜索引擎与网站推广技术	285
13.2.1	搜索引擎的工作原理	285
13.2.2	搜索引擎的发展趋势	286
13.2.3	常用搜索引擎简介	287
13.2.4	搜索引擎优化 SEO	287

13.3	ASP.NET 技术介绍	288
13.3.1	DOTNET 开发平台	288
13.3.2	DOTNET 主要技术	289
13.3.3	DOTNET 编程语言	291
13.4	PHP 技术介绍	292
	参考文献	294

第1章 Web 技术概述

【学习提示】

互联网在人类生活中越来越重要，云计算、物联网等热门技术也不断成熟，对网站开发技术深入系统地学习可以使程序员成为技术市场上炙手可热的人才。但对于有兴趣学习 Web 技术的开发者而言，不断涌现(有很多也是昙花一现)的开发工具、程序设计语言和设计框架让人眼花缭乱、无所适从。

本章将从 Web 的历史开始介绍，通过讨论 B/S 架构和最基本的 HTML 文档结构，逐渐搭建稳健的 Web 开发技术知识大厦。

1.1 Web 系统简介

1980 年，作为瑞士日内瓦的欧洲核子研究中心的软件工程师，Tim Berners-Lee 遇到了一个许多人都经常碰到的问题：工作过程中，他需要频繁地与世界各地的科学家们沟通联系、交换数据，还要不断地回答一些问题，这些重复而繁琐的过程实在令他烦恼。他希望能够有一种工具，让大家可以通过计算机网络快捷地访问其他人的信息和数据。于是 Tim Berners-Lee 开始在业余时间编写一个软件程序，利用一系列标签描述出信息的内容和表现形式，再通过链接把这些文件串起来，让世界各地的人能够轻松共享信息。Tim Berners-Lee 给这种系统命名为“World Wide Web”，1990 年 11 月，第一个 Web 服务器 nxoc01.cern.ch 开始运行。

1993 年美国伊利诺州伊利诺大学的 Marc Andreessen 及其同事开发出了第一个支持图文并茂展示网页的 Web 浏览器——Mosaic 浏览器，并成立了网景公司(Netscape Communication Corp)。图 1-1 为 Mosaic 浏览器的界面。



图 1-1 Mosaic 浏览器的界面

1994 年 10 月 Tim Berners-Lee 联合 CERN、DARPA 和欧盟成立了 Web 的核心技术机构——W3C(World Wide Web Consortium, 万维网联盟)。从那之后, Web 的每一步发展、技术成熟和应用领域的拓展都离不开 W3C 的努力。W3C 会员(大约 500 名会员)包括软、硬件产品及服务的提供商、内容供应商、团体用户、研究机构、标准制定机构和政府部门, 该组织已成为专门致力于创建 Web 相关技术标准并促进 Web 向更深、更广发展的国际组织。

从技术方面看, Web 通过超文本标记语言(Hyper Text Markup Language, HTML)实现信息与信息的连接; 通过统一资源标识符(Uniform Resource Identifier, URI)实现全球信息的精确定位; 通过超文本传输协议(HyperText Transfer Protocol, HTTP)实现信息在互联网中的传输。

作为一种典型的分布式应用架构, Web 应用中的每一次信息交换都要涉及到客户端和服务端两个层面。因此, Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。Web 客户端的主要任务采用 HTML 语言及其相关技术(包括 CSS 和 JavaScript 等)获取用户的输入并根据用户的访问需求展现信息内容; Web 服务器端的主要任务是按照用户的输入和需求搜索相关数据组成完整的 HTML 文档传输给客户端。

近年来, 随着 Web 应用需求不断增加, Web 的开发技术也飞速发展, 出现了大量的 Web 开发工具、程序库和框架。面对这些纷繁复杂的技术, 如何选择学习的入口, 如何掌握技术发展的趋势, 如何应对大型的 Web 开发项目, 这些问题的解决都需要从理论和技术的基础出发, 通过恰当的案例实践, 逐步找到知识的脉络和规律。扎实的理论和基础知识不仅可以帮助我们进行 Web 的开发, 而且有利于在实践中不断学习、掌握和应用新的理论和技术, 形成“可持续发展”的知识结构。

1.2 B/S 结构和 Web 应用程序

基于 Web 服务器和浏览器共同构建的软件系统被称为浏览器/服务器体系结构(Browser/Server Architecture, 简称 B/S 结构), 它是对传统客户机/服务器体系结构(Client/Server Architecture, 简称 C/S 结构)的一种变化或者改进的结构。

Web 应用程序(Web Application)是指在 B/S 结构中, 通过浏览器访问在 Web 服务器端运行的应用程序, 用户只需要有浏览器、网站地址和权限就可以访问 Web 应用程序; 与 Web 应用程序相对应的是传统 C/S 结构中的桌面应用程序(Desktop Application), 它需要在客户端进行安装和运行。

电子邮件应用是一个典型的例子。我们可以通过特定的桌面应用程序(比如 Outlook、Foxmail 等)收发信件或管理邮箱, 也可以不使用任何特殊的桌面应用程序而只通过浏览器直接访问电子邮件服务网站(如 mail.139.com 或 mail.google.com 等)。

可以看出, 传统 C/S 结构中的客户机不是毫无运算能力的输入、输出设备, 而是具有一定的数据存储和数据处理能力。通过把应用程序的计算和数据合理地分配在客户机和服务器两端, 可以有效地降低网络通信量和服务器运算量。而在 B/S 结构下, 用户界面完全通过 Web 浏览器实现, 一部分事务逻辑在浏览器端(有时称为前端)实现, 但是主要事务逻辑在服务器端(有时称为后端)实现。

Web 应用程序利用了不断成熟的 WWW 浏览器技术, 结合在浏览器上运行 JavaScript 程序的能力, 在通用浏览器上实现了原来需要复杂专用软件才能实现的强大功能。B/S 结构相对于传统的 C/S 结构更加适合开发多层架构的系统, 如图 1-2 所示。

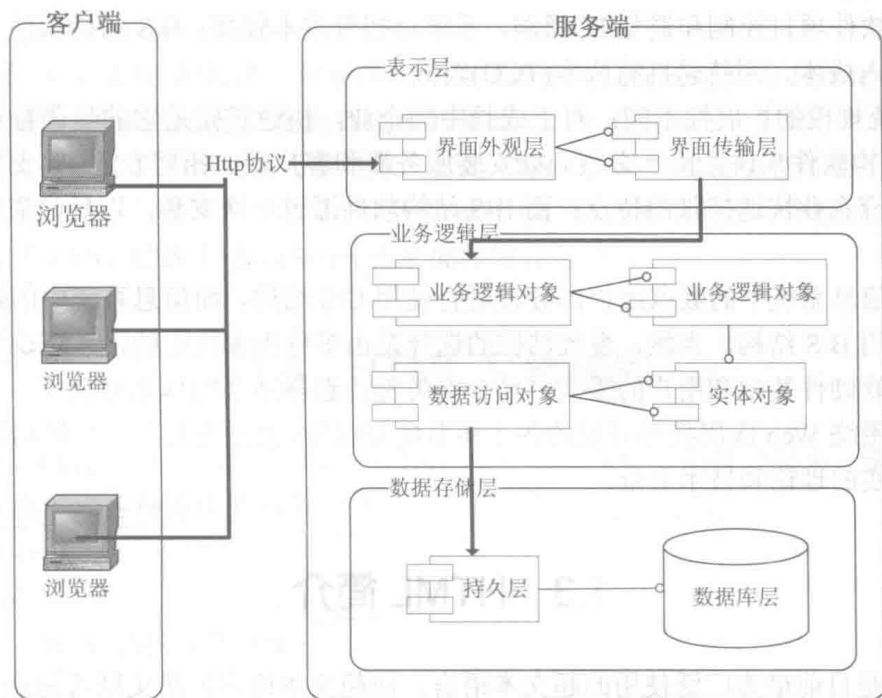


图 1-2 B/S 的系统结构图

B/S 结构相对于传统 C/S 结构是一个巨大的进步。在搭建信息系统时, 两者也体现出明显的不同之处:

(1) 硬件环境不同。传统的 C/S 一般建立在专用的网络上, 小范围里的网络环境和局域网之间再通过专门服务器提供连接和数据交换服务; B/S 适合建立在广域网之上, 不必是专门的网络硬件环境。

(2) 对安全要求不同。传统的 C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强, 一般高度机密的信息系统适宜采用 C/S 结构; B/S 通常建立在广域网之上, 对安全的控制能力相对较弱, 面向是不可知的用户群, 因此更适合发布各种公开信息。

(3) 系统架构不同。传统的 C/S 结构可以更加注重流程, 可以对权限多层次校验; B/S 系统所依托的 HTTP 协议缺少对流程、状态等方面的管理, 因此在实际的 B/S 系统开发中需要采用更加优化的开发和运行平台, 包括微软的 .NET 平台或 JavaEE 平台等。

(4) 系统维护不同: 传统的 C/S 结构意味着在用户的计算机中必须安装特定的客户端软件, 如果系统出现了问题或者需要对系统进行升级, 就必须在每一个客户端计算机上进行操作; B/S 结构的维护和升级都发生在服务器端。

(5) 处理问题不同: 传统的 C/S 结构适合对大量数据进行批量的增、删、改操作, 尤其适合对数据库中的数据进行管理; B/S 结构适合面向不同的用户群, 接受用户数据的汇集和用户数据库的各种查询。

(6) 用户接口不同: 传统的 C/S 结构的前台大多建立在 Windows 平台上, 客户端软件

对操作系统有特定的要求，跨平台性较差；B/S 的前台建立在浏览器上，对操作系统没有特别的要求，一般只要有操作系统和浏览器就行，具有良好的跨平台性。因此，云计算或软件即服务(SaaS)的系统大多基于 B/S 结构建立。

(7) 投入成本构成不同：传统的 C/S 结构的软件随着应用范围的扩大，投资会连绵不绝，不利于软件项目控制和避免 IT 黑洞，系统总拥有成本较高；B/S 结构软件一般只有初期一次性投入成本，系统总拥有成本(TCO)较低。

(8) 系统规模的扩展性不同：对于成长中的企业，快速扩张是它的显著特点。对于传统的 C/S 结构软件来讲，由于必须到处安装服务器和客户端、招聘专业 IT 支持人员等，因此无法适应企业快速扩张的特点；而 B/S 结构软件通过一次安装，以后只需设立账号、培训即可。

总之，信息系统中的数据维护部分较适合使用 C/S 结构，而信息系统中的数据查询部分较适合使用 B/S 结构。当然，系统结构的选择是由多种因素决定的，系统设计人员需要根据系统的软硬件基础和用户的需求，结合业务特点选择适合的体系结构。

本书将围绕 Web 应用程序开发的各个环节展开讨论，为开发基于 B/S 结构的信息系统打下较为坚实的理论和技術基础。

1.3 HTML 简介

HTML 是目前最为广泛使用的超文本语言，而超文本语言的历史则可追溯到 20 世纪 40 年代。1945 年，Vannevar Bush(著名的曼哈顿计划的组织者和领导者)发表论文描述了一种被称为 MEMEX 的机器，其中已经具备了超文本和超连接的概念。Doug Engelbart(鼠标的发明者)等人则在 1960 年前后，对信息关联技术做了最早的实验。与此同时，Ted Nelson 正式将这种信息关联技术命名为超文本(Hypertext)技术。随后在 1969 年，Charles F. Goldfarb 博士带领 IBM 公司的一个小组开发出通用标记语言(Generalized Markup Language, GML)，并在 1978—1986 年间，将 GML 语言进一步发展成为著名的标准通用标记语言(Standard Generalized Markup Language, SGML)。当 Tim Berners-Lee 和他的同事们在 1989 年试图创建一个基于超文本的分布式信息系统时，Tim Berners-Lee 意识到，SGML 是描述超文本信息的一个最佳方案。于是，Tim Berners-Lee 应用 SGML 的基本语法和结构为 Web 量身定制了 HTML。

HTML 是使用 SGML 定义的一个描述性语言，也可以说，HTML 是 SGML 的一个应用。HTML 不是如 C++ 和 Java 之类的程序设计语言，而只是标记语言。HTML 的格式和语法非常简单，只是由文字及标签组合而成的，任何文字编辑器都可以编辑 HTML 文件，只要能将文件另存成 ASCII 纯文字格式即可。当然，使用专业的网页编辑软件设计网页更加方便，例如 FrontPage 或 Dreamweaver 等，甚至 Word 软件都可以将文件保存为 HTML 格式。

在开发技术的选型中，通常会选择传统 HTML 的扩展技术，包括可扩展超文本标记语言(eXtensible HyperText Markup Language, XHTML)和动态 HTML(Dynamic HTML, DHTML)。

XHTML 是与 HTML 同类的语言,不过语法上更加严格。从对象关系上讲,HTML 是基于 SGML 的应用,但语法规则较为“宽松”;而 XHTML 则基于 XML 的应用,严格服从 XML 的语法规则。XML 是 SGML 的一个子集,关于它的技术将在后面章节中描述。XHTML 1.0 在 2000 年成为 W3C 的推荐标准,在各种网页的开发工具中都有对 XHTML 标准的设置选项。

DHTML 并非 W3C 的标准,它是微软等公司给出的相对传统静态 HTML 而言的一种开发网页的概念。DHTML 建立在原有技术的基础上,主要包括三个方面:一是 HTML(或 XHTML),其中定义了各种页面元素对象;二是 CSS,CSS 中的属性也可被动态操纵,从而获得动态的效果;三是客户端脚本(包括 JavaScript 等),用以编写程序操纵 Web 页上的 HTML 对象和 CSS。CSS 和 JavaScript 在后面将进行讨论。

1.4 HTML 文档结构

下面代码就是一个最简单的网页,HTML 文件中的各种元素组合起来,通过浏览器的解析和展现就形成各种各样的网页。代码的运行效果如图 1-3 所示。

```
<html>
<head>
  <title>网页制作教学</title>
</head>
<body>
  Hello World!
</body>
</html>
```

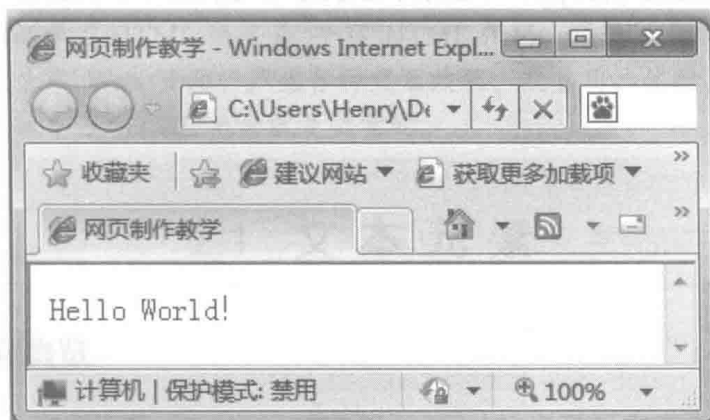


图 1-3 网页的浏览效果

在 HTML 中任何标签皆由“<”及“>”所围住,如 <P>,标签名与小于号之间不能留有空白字符。在起始标签之标签名前加上符号“/”便是其终结标签,如 。标签字母大小写皆可。由开始标签和终结标签所构成的对象可以称为 HTML 元素(或 HTML 对象)。元素带有参数,也称为元素的属性。参数只可加于起始标签中。熟悉面向对象程序设

计的开发人员更习惯将它们称为“HTML 对象和属性”，本书并不特别强调它们的名称，但在不同的章节中会根据上下文的情况给出不同的名称。

通常在一个完整的 HTML 文件中，html 元素是 HTML 文档的根元素，其中包含两个部分：头部元素和体部元素，它们分别被包含在<head>标签和<body>标签中。另外，CSS 和 JavaScript 增强了 HTML 的表现能力，同时也可以使 Web 浏览器端的开发更加符合模块化、可扩展性、面向对象等软件工程方面的要求。

思考题

1. HTML 与 W3C 组织分别是如何产生的？
2. C/S 体系结构和 B/S 体系结构有何不同之处？



第2章 HTML 元素

【学习提示】

构建基于 B/S 结构的系统离不开两部分技术：Web 浏览器端技术(前台技术)和 Web 服务器端技术(后台技术)。Web 浏览器端技术包括了 HTML、CSS、JavaScript 等。HTML 语言是 Web 的核心技术，其标准由 W3C 制定。

经过 8 年的努力，2014 年 10 月 28 日 W3C 的 HTML 工作组正式发布了 HTML5 的正式推荐标准(W3C Recommendation)。作为下一代 Web 技术标准，HTML5 备受期待和瞩目。HTML5 的新特性主要在于多媒体信息的丰富表现方式、浏览器端的复杂应用支持和对性能的优化和改进等方面。HTML5 带来了一组新的用户体验，如 Web 的音频和视频不再需要插件，通过 Canvas 能更灵活地完成图像绘制，而不必考虑屏幕的分辨率，浏览器对可扩展矢量图(SVG)和数学标记语言(MathML)的本地支持，通过引入新的注释信息以增强对东亚文字(包括简体及繁体中文、日文、韩文等)呈现的支持，对富 Web 应用(基于 JavaScript 和 AJAX 等技术实现的用户体验丰富的 Web 应用程序)等新特性的支持等。相对之前的版本，HTML5 增加了许多新的元素，同时也在标准中删除了一些不再适用的元素，但这并不意味着开发者将不能再使用那些元素，因为几乎所有支持 HTML5 技术的浏览器都会长期继续支持之前的标准。

本章中节的划分参考了 W3C 的 HTML5 推荐标准文档中的元素归类划分方式。在讨论各种元素时，会列出该元素在 HTML4 和 HTML5 中被支持的情况，以供参考。

很多初学者认为既然可以使用各种可视化网页设计工具(例如 Dreamweaver 等)“画”出网页，那就大可不必学习 HTML 语法并“手写”HTML 代码。设计工具确实可以事半功倍，但如果需要动态地生成 Web 页面、产生各种交互效果、改善用户体验，那么学习 HTML 语法，特别是系统地学习 HTML 元素就是必需的。

2.1 文本元素

2.1.1 文本元素概览

在 HTML 中，文本元素(Text-level Semantics)用来定义网页中文本内容和语义，增加文字的易读性。文本元素主要包括<a>、、、<small>、<s>、<cite>、<q>、<dfn>、<abbr>、<time>、<code>、<var>、<samp>、<kbd>、<sub>、<sup>、<i>、、<u>、<mark>、<ruby>、<rb>、<rt>、<rtc>、<rp>、<bdi>、<bdo>、、
、<wbr>等。虽然文本的显示样式通常是由 CSS 来定义的，但文本元素的语义也会影响文本的显示风格，比如上标(sup)、下标(sub)等文本。文本元素及其功能说明如表 2-1 所示。

表 2-1 文本元素及其功能说明

元素	语 义	HTML 支持版本
<a>	定义超链接	4、5
	定义强调文本, 指定文本的性质, 显示为斜体字	4、5
	定义强调文本, 指定文本的性质, 显示为粗体字	4、5
<small>	定义小号文本	4、5
<s>	定义加删除线的文本	4、5
<sub>	定义下标文本	4、5
<sup>	定义上标文本	4、5
<i>	定义斜体文本, 指定文字该如何显示	4、5
	定义粗体文本, 指定文字该如何显示	4、5
<u>	定义下划线文本	4、5
<mark>	定义有加亮记号的文本	5
<ruby>	定义注释, 可显示中文注音或字符	5
<rb>	定义部分 ruby 注释	5
<rt>	定义字符(中文注音或字符)的解释或发音	5
<rp>	在 ruby 注释中使用, 以定义不支持 ruby 元素的浏览器所显示的内容	5
<rtc>	定义 ruby 注释中的文本容器	5
<cite>	定义作品(比如书籍、歌曲、电影、电视节目、绘画、雕塑等)的标题	4、5
<q>	定义一个短的引用	4、5
<dfn>	定义一个定义项目	4、5
<abbr>	标记一个缩写	4、5
<time>	定义日期、时间	5
<code>	定义软件代码文本	4、5
<var>	定义变量	4、5
<samp>	代表从软件系统中输出的文本	4、5
<kbd>	代表通过键盘输入的文本	4、5
<bdi>	设置一段文本, 使其脱离其父元素的文本方向设置	5
<bdo>	重新设置其中的文本方向	4、5
	可以将文本的一部分独立出来进行格式设置	4、5
 	插入换行符。一个 元素代表一个换行	4、5
<wbr>	定义在文本中的特定位置折行	5
<center>	定义居中的文本	4
	定义文本的字体外观、尺寸和颜色。常用属性为 face、size、color	4
<strike>	定义删除线文本	4
<big>	定义大号文本	4