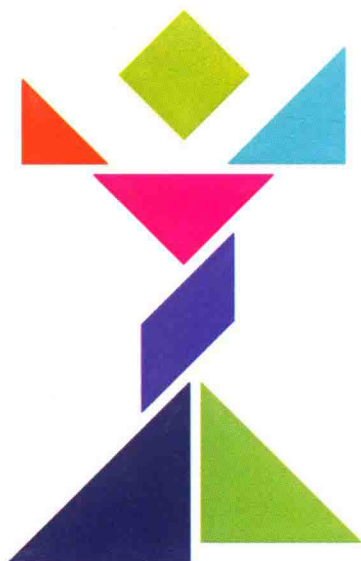




21世纪高等学校计算机
应用技术规划教材

ASP.NET Web 应用开发技术 (第2版)



◎ 喻钧 白小军 主编 岳鑫 代军 副主编

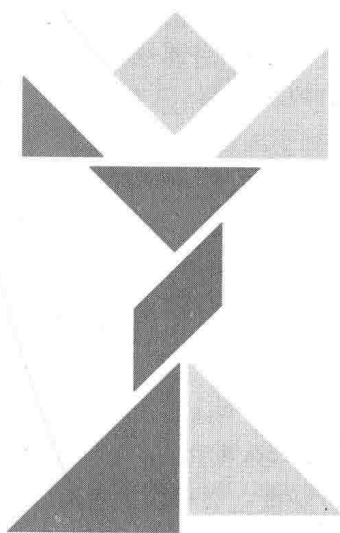


清华大学出版社



21世纪高等学校计算机
应用技术规划教材

ASP.NET Web 应用开发技术 (第2版)



◎ 喻钧 白小军 主编 岳鑫 代军 副主编



清华大学出版社
北京

内 容 简 介

本书从 ASP.NET 初学者的角度出发,由浅入深地对 ASP.NET Web 开发技术进行了详细介绍。本书共分为 10 章,分别介绍了 Web 程序设计基础,HTML、XML 与 CSS,客户端编程技术与开发框架,C# 语法基础,Web Form 技术,Web 应用程序状态管理,数据访问技术,数据绑定,MVC 开发模式,AJAX 与 Web API 等内容。

全书内容翔实,通俗易懂,适合自学。书中给出了丰富的实例以帮助读者深入理解和学习,在每章的后面还配有习题和上机练习。本书既可作为高等院校学生的教材,也可作为自学 ASP.NET 动态网页设计的读者的参考书。

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

ASP.NET Web 应用开发技术/喻钧,白小军主编.—2 版.—北京:清华大学出版社,2017
(21 世纪高等学校计算机应用技术规划教材)
ISBN 978-7-302-47948-2

I. ①A… II. ①喻… ②白… III. ①网页制作工具—程序设计 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2017)第 207214 号

责任编辑:魏江江 王冰飞

封面设计:刘 键

责任校对:徐俊伟

责任印制:宋 林

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:22.75

字 数:568 千字

版 次:2013 年 1 月第 1 版

2017 年 10 月第 2 版

印 次:2017 年 10 月第 1 次印刷

印 数:6001~8000

定 价:49.00 元

产品编号:071244-01

ASP.NET 是构建现代 Web 应用及 Web 服务的开放平台,它基于 .NET 框架,并充分利用 HTML5、CSS 及 JavaScript 的优势,帮助开发人员构建简单、高效、易扩展的企业级应用。

作为 Web 应用开发的两大主流技术,Java EE 与 ASP.NET 都已发展得非常成熟,体系结构日趋完善。在行业内,.NET 技术越来越受到企业的重视,应用越来越广泛,国内外对 .NET 研发人员的需求量也在不断上升,熟悉 .NET 技术体系的学生就业前景良好。

本书作者长期从事 Web 应用开发和 .NET 技术课程的一线教学工作,有着深厚的开发功底和丰富的教学经验;熟悉 ASP.NET 和 Java EE 两大主流技术体系,对面向对象技术、设计模式、软件架构等知识理解较为深刻,能够站在理论的高度来指导实践;同时,作者也非常了解学生的认知规律。

Web 应用开发有着很强的技巧性,要求学生从整体上把握软件的架构、框架,合理地使用设计模式,这样才能设计出稳定性好、扩展性强的软件产品。很多培训公司的课程体系和教材注重实践,却缺乏理论深度,培养出来的学生能够应付就业,却难以取得长远的发展。本书更注重思想方法的培养,将面向对象思想、设计模式和软件架构的知识融入各章节教学中,尽量使学生知其然并知其所以然,以思想方法指导设计实践。

本书第 1 版发行后,得到了广大师生的喜爱,并且加印了 3 次。近年来,Web 开发技术又有了新的发展趋势,尤其是 MVC 开发模式、Web Service 技术及 AJAX 技术的应用全面爆发,对象关系映射机制深入人心,而客户端编程技术和框架也变得越来越重要。在第 1 版中,这些关键内容没有充分体现出来,因此在新一版教材中做了改进。

本次改版充分反映 Web 2.0 时代的技术特征,对第 1 版教材内容进行了大刀阔斧的调整,增加了 MVC 框架、实体模型、客户端编程框架、响应式设计等内容,同时顺应技术潮流,重新设计了 Web Service 与 AJAX 技术的内容体系,突出 REST 风格的 Web Service 以及 AJAX 与 Web API 的交互,更利于合理构建软件架构,并开发扩展性强的大型 Web 应用。

全书的所有程序在 Windows 7/Windows 10、IIS 7/IIS Express,.NET Framework 4 下测试通过,数据库使用 SQL Server LocalDB 及 SQL Server Express,集成开发环境采用 Visual Studio 2015 及 2017 Community 版,所有开发工具都为正版、免费产品,可以从微软官方网站免费下载。

本书第 1 章、第 2 章、第 4~6 章由喻钧编写,第 7~10 章由白小军编写,第 3 章由岳鑫编写,第 1 章、第 5 章部分内容和程序由代军完成,全书由喻钧统稿。

尽管在编写本书的过程中尽了最大努力,但由于编者水平有限,疏漏及不妥之处在所难免,恳请读者批评指正。

编者
2017 年 8 月

目 录

第 1 章 Web 程序设计基础	1
1.1 Web 的工作原理	1
1.1.1 软件体系结构	1
1.1.2 HTTP 协议	2
1.2 Web 程序设计技术	4
1.2.1 客户端技术	4
1.2.2 服务器端技术	5
1.3 ASP.NET 基础	7
1.3.1 Microsoft .NET Framework	7
1.3.2 ASP.NET 的工作原理	8
1.3.3 ASP.NET 开发的 4 种模式	9
1.4 建立 ASP.NET 运行和开发环境	10
1.4.1 安装和配置 Web 服务器	11
1.4.2 安装 Visual Studio 开发环境	12
1.4.3 创建一个 ASP.NET Web 站点	15
1.5 习题与上机练习	19
第 2 章 HTML、XML 与 CSS	20
2.1 使用 HTML 组织页面内容	20
2.1.1 HTML 文档的基本结构	20
2.1.2 HTML 文档的主要标记	22
2.2 使用 XML 表达数据	31
2.2.1 XML 的概念	31
2.2.2 XML 的语法规则	34
2.2.3 XML 文档的显示	35
2.3 使用 CSS 表达页面样式	36
2.3.1 在 HTML 中使用 CSS	36
2.3.2 CSS 样式规则	39
2.3.3 CSS 选择器	41
2.3.4 常见的样式属性	44
2.4 习题与上机练习	50

第3章 客户端编程技术与开发框架	53
3.1 JavaScript 概述	53
3.1.1 什么是 JavaScript	53
3.1.2 在网页中嵌入 JavaScript 脚本	54
3.1.3 使用 JavaScript 输入与输出信息	56
3.2 JavaScript 基本语法	58
3.2.1 数据类型	58
3.2.2 变量	58
3.2.3 运算符和表达式	59
3.2.4 流程控制	60
3.2.5 函数	63
3.2.6 异常处理	65
3.2.7 事件处理	67
3.3 JavaScript 对象编程	69
3.3.1 常用的 JavaScript 对象	69
3.3.2 浏览器对象模型	72
3.3.3 文档对象模型	75
3.4 jQuery 框架	80
3.4.1 jQuery 基础	80
3.4.2 使用 jQuery 操作 HTML 元素	83
3.4.3 使用 jQuery 进行 DOM 遍历	85
3.4.4 使用 jQuery 实现网页特效	87
3.5 Bootstrap 框架	88
3.5.1 Bootstrap 基础	88
3.5.2 栅格布局系统	92
3.5.3 表单系统	94
3.5.4 导航条组件	97
3.5.5 面板组件	98
3.5.6 模态框插件	99
3.6 习题与上机练习	100
第4章 C# 语言基础	102
4.1 创建一个简单的 C# 程序	102
4.2 C# 基本语法	104
4.2.1 C# 数据类型	104
4.2.2 运算符和表达式	108
4.2.3 程序控制结构	112
4.3 类和对象	117

4.3.1	类和对象的创建	117
4.3.2	属性和方法	120
4.3.3	构造函数和析构函数	122
4.3.4	继承和多态	124
4.4	字符串	125
4.4.1	使用字符串	125
4.4.2	创建动态字符串	130
4.5	集合编程	132
4.5.1	ArrayList	132
4.5.2	哈希表	134
4.5.3	队列	136
4.5.4	堆栈	137
4.6	习题与上机练习	138
第5章	Web Form 技术	140
5.1	ASP.NET 页面的生命周期	140
5.2	Web 服务器控件概述	141
5.2.1	服务器控件的不同类型	141
5.2.2	服务器控件的共有属性和事件	142
5.3	标准的 Web 服务器控件	144
5.3.1	文本输入与显示控件	144
5.3.2	控制权转移控件	146
5.3.3	选择控件	148
5.3.4	容器控件	154
5.4	数据的有效性检测	157
5.4.1	必须输入验证控件	157
5.4.2	比较验证控件	158
5.4.3	范围验证控件	160
5.4.4	正则表达式验证控件	160
5.4.5	自定义验证控件	162
5.4.6	验证总结控件	164
5.5	用户控件	164
5.5.1	用户控件概述	164
5.5.2	创建用户控件	165
5.5.3	用户控件的使用	167
5.6	母版页	168
5.6.1	母版页概述	168
5.6.2	创建一个母版页	169
5.6.3	使用母版创建网页	172

5.7 习题与上机练习	173
第6章 Web 应用程序状态管理	175
6.1 HTTP 请求处理	175
6.1.1 Response 对象	175
6.1.2 Request 对象	177
6.1.3 Server 对象	180
6.2 状态信息的保存	184
6.2.1 Application 对象	185
6.2.2 Session 对象	189
6.2.3 Cookie 对象	191
6.2.4 ViewState 对象	193
6.3 习题与上机练习	195
第7章 数据访问技术	197
7.1 ADO.NET 体系结构	197
7.1.1 ADO.NET 数据提供程序	197
7.1.2 ADO.NET 数据集	198
7.2 使用基于连接的对象访问数据库	199
7.2.1 访问数据库的一般方法	200
7.2.2 使用 Connection 对象	202
7.2.3 使用 Command 对象	207
7.2.4 使用 DataReader 对象	213
7.3 使用基于内容的对象访问数据库	214
7.3.1 使用 DataTable	214
7.3.2 使用 DataView	217
7.3.3 使用 DataRelation	218
7.3.4 使用 DataAdapter	219
7.4 Entity Framework 基础	221
7.4.1 使用 Entity Framework 访问关系数据	221
7.4.2 基于 Entity Framework 的几种开发方式	226
7.4.3 概念模型设计	227
7.4.4 DbContext 类及其使用	229
7.4.5 查询并检索实体	230
7.4.6 加载关联实体	232
7.4.7 实体的增、删、改操作	233
7.5 习题与上机练习	234

第 8 章 数据绑定	236
8.1 数据绑定基础	236
8.1.1 数据绑定表达式	236
8.1.2 单值绑定	237
8.1.3 重复值绑定	238
8.2 数据源控件	240
8.2.1 数据源控件概述	240
8.2.2 SqlDataSource 控件	244
8.2.3 ObjectDataSource 控件	247
8.3 数据绑定控件	253
8.3.1 GridView 控件	253
8.3.2 ListView 控件	272
8.3.3 DetailsView 控件	275
8.3.4 FormView 控件	276
8.4 使用实体框架与模型绑定技术	279
8.5 习题与上机练习	290
第 9 章 MVC 开发模式	292
9.1 MVC 基础	292
9.1.1 创建一个 ASP.NET MVC 项目	292
9.1.2 分析 MVC 项目的结构与运转流程	294
9.2 模型与控制器组件的使用	298
9.2.1 创建模型组件	298
9.2.2 创建控制器组件	301
9.2.3 程序结构与运转机制	302
9.3 视图组件的使用	307
9.3.1 定义视图的整体外观	307
9.3.2 生成视图的内容	310
9.4 在控制器和视图间传递数据	313
9.4.1 从视图向控制器传递数据	313
9.4.2 从控制器向视图传递数据	314
9.5 习题与上机练习	315
第 10 章 AJAX 与 Web API	316
10.1 AJAX 技术	316
10.1.1 AJAX 技术基础	316
10.1.2 传统的 AJAX 编程方式	317
10.1.3 使用 jQuery 简化 AJAX 编程	323

10.2	Web API 框架	328
10.2.1	Web API 基础	328
10.2.2	Web API 中的路由	335
10.2.3	Web API 中的返回值	337
10.3	单页应用程序示例	338
10.4	习题与上机练习	351

第1章

Web程序设计基础

Web 技术的迅速发展极大地改变了整个世界,也深刻地影响了人们的日常生活。作为一个出色的 Web 开发框架,ASP.NET 已经成为 Web 应用开发中不可缺少的中坚力量,由此衍生出数以万计的产品、服务以及开源工程。学习 ASP.NET 程序设计基础将有助于理解 Web 应用的开发。

本章将从 Web 的工作原理开始阐述 Web 开发基础知识,并与读者一起搭建 ASP.NET 的运行和开发环境,创建第一个 ASP.NET Web 站点。

1.1 Web 的工作原理

当用户访问 Web 站点时,数据是遵从 HTTP(HyperText Transfer Protocol)协议进行传输的。HTTP 即超文本传输协议,是基于 Browser/Server(浏览器/服务器,B/S)模式使用 TCP 连接在应用层进行可靠的数据传输。

1.1.1 软件体系结构

在目前的应用软件开发领域中主要包括两大软件开发体系结构,一种是 Client/Server(客户机/服务器,C/S)结构,一种是 Browser/Server(浏览器/服务器)结构。

C/S 是较早的软件体系结构,主要适用于局域网环境,如图 1-1 所示。服务器通常采用高性能的 PC 或工作站,并安装大型的数据库系统,如 Oracle、Sybase 或 SQL Server 等。客户机需要安装专门的客户端应用程序。这种结构能够充分发挥客户端 PC 的处理能力,客户端响应速度快。但系统的可扩展性和可维护性差,例如软件升级时,每台客户机都需要重新安装,系统维护和升级的成本高。另外,由于采用 Intranet 技术,适用于局域网环境的可连接用户数有限,当用户数量增多时,系统性能会明显下降,代码的可重用性差。

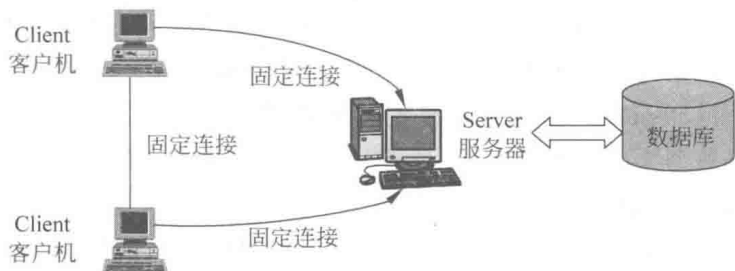


图 1-1 C/S 软件开发体系结构

B/S 结构是随着 Web 技术的发展逐渐成熟的软件体系结构。在这种结构中,所有的应用程序以及数据库系统都安装在服务器(Server)上,客户端只需安装任意一个浏览器(Browser)即可,它是零维护的。用户通过浏览器向服务器发出一个请求(如在地址栏中输入一个网址,或者单击超链接及按钮),服务器处理该请求后,将结果以 HTML 的形式返回给客户端,如图 1-2 所示。B/S 结构采用 Internet/Intranet 技术,适用于广域网环境。它可以根据访问量动态地配置 Web 服务器、应用服务器,以支持更多的客户。其代码的可重用性好,系统的扩展维护简单。



图 1-2 B/S 软件体系结构

B/S 结构和 C/S 结构的比较如表 1-1 所示。

表 1-1 B/S 结构与 C/S 结构的比较

	B/S 软件体系结构	C/S 软件体系结构
硬件环境	广域网,不必是专门的网络环境,只要是能接入 Internet 的用户均可	局域网,专门的小范围网络硬件环境,用户固定,用户数量有限
系统维护	客户端零维护,易于实现系统的无缝升级	升级和维护难,成本高
软件重用性	多重结构,各构件相对独立,可重用性较好	单一结构,软件整体性较强,各部分间的耦合性强,可重用性较差
平台相关性	客户端和服务端是平台无关的	客户端和服务端是平台相关的,多是 Windows 平台
安全性	面向不可知的用户群,对信息安全的控制能力相对较弱	面向相对固定的用户群,对信息安全的控制能力强

1.1.2 HTTP 协议

HTTP 协议定义了 Web 浏览器和 Web 服务器之间交换数据的过程以及数据本身的格式,它是客户机与服务器交互遵守的协议。

HTTP 协议的工作原理如图 1-3 所示。它表示基于 HTTP 的信息交换过程,共分为 4 个步骤,即建立连接、发送请求、返回响应、断开连接。首先,客户端的浏览器向服务器的某个端口发出请求,建立与服务器的连接,通常默认端口号为 80。在连接建立后客户端向服务器发出一个请求(Request)。服务器接收和处理请求后返回一个响应(Response)页面。最后,客户端与服务器之间的连接断开,通信结束。一般来说,任何一方都可结束连接,但通常是客户端收到所请求的信息后关闭连接。

HTTP 协议是一种请求/应答协议,它通过在客户机和服务器之间发送请求、应答消息的方式工作。通常,HTTP 消息包括客户机向服务器的请求消息、服务器向客户机的响应

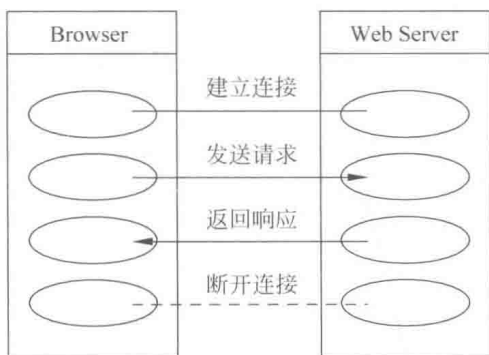


图 1-3 HTTP 的工作原理

消息。

一个 HTTP 请求消息包括 3 个部分,即一个请求行、若干消息头以及消息的实体内容。其中一些消息头和实体内容是可选的,消息头和实体内容之间用空行隔开。

例如,图 1-4 所示的矩形框就是一个 HTTP 请求消息的内容。第 1 行为请求行,中间的若干行为消息头,空行表示消息头的结束,最后是消息的实体内容。

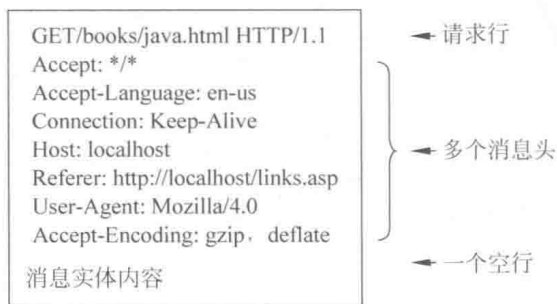


图 1-4 HTTP 请求消息

在图 1-4 中,第 1 行(请求行)表示客户端采用 GET 方式向服务器传送数据,请求的 URL 地址为本地文件/books/java.html,使用的 HTTP 协议版本为 1.1。

从第 2 行开始,直到空行前是 HTTP 请求头域的内容,它的常用参数如下。

- Accept: 用于指定客户端可以接受的 MIME 类型,如 */* 表示任何类型。
- Accept-Charset: 指定客户端可以使用的字符集,如 UTF-8。
- Accept-Language: 指定客户端的接收语言,可以指定多个,如 zh-cn。
- Accept-Encoding: 指定客户端的接收编码,如 gzip、deflate。
- Connection: 指示处理完本次请求/响应后,客户端与服务器是否继续保持连接,取值为 Keep-Alive 或 close,默认为 Keep-Alive。
- Host: 指定资源所在的主机和端口号。
- Referer: 确定获得请求 URI 的资源地址。
- User-Agent: 用户代理,指定初始化请求的客户端程序,如浏览器等。

和请求消息类似,一个 HTTP 响应消息包括 3 个部分,即一个状态行、若干消息头以及实体内容。其中一些消息头和实体内容是可选的,消息头和实体内容之间用空行隔开。

例如,图 1-5 展示了一个 HTTP 响应消息的内容。第 1 行为状态行,中间为若干消息头,空行表示消息头的结束,最后是消息的实体内容。

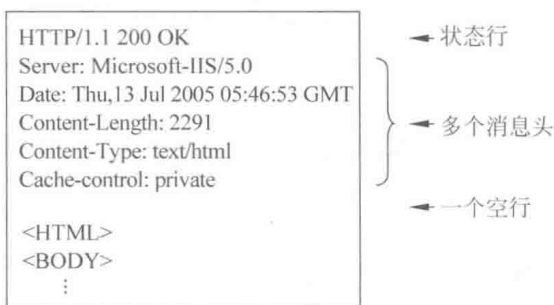


图 1-5 HTTP 响应消息

1.2 Web 程序设计技术

在 Web 程序的设计过程中,主要用到客户端技术和服务器端技术。客户端技术也叫 Web 前端技术,主要包括 HTML、CSS、JavaScript 等。服务器端技术也叫 Web 后端技术,主要包括 ASP.NET、PHP、JSP、Python 等。

无论 Web 前端还是后端,都要用到脚本编程。脚本(Scripts)是指嵌入到 Web 页中的程序代码,所使用的编程语言称为脚本语言。按照执行方式和位置的不同,脚本分为客户端脚本(Client-side Scripts)和服务器端脚本(Server-side Scripts)。脚本编程的基本特点是:由 HTML 构造页面模板,由语言解释引擎解释并运行脚本,运行后产生的页面内容插入到页面模板中。

客户端脚本的解释器位于 Web 浏览器中,在客户机上被 Web 浏览器执行。服务器端脚本的解释器则位于 Web 服务器中,在服务器上被 Web 应用服务器执行。

1.2.1 客户端技术

1. HTML 超文本标记语言

HTML(HyperText Marked Language,超文本标记语言)是 Web 页的标记性语言。“超文本”就是指页面内可以包含图片、链接,甚至音乐、程序等非文字元素。HTML 文件本身是一种文本文件,通过在文本文件中添加标记符可以告诉浏览器如何显示 HTML 文件中的内容(如文字如何处理、画面如何安排、图片如何显示等)。

浏览器按顺序解释执行 HTML 文件,对于书写出错的标记既不报错,也不停止执行。需要注意的是,不同的浏览器对于同一标记可能有不同的解释,也就可能有不同的显示效果。

2. CSS 样式表

CSS(Cascading Style Sheets,层叠样式表)是一种用来表现 HTML 或 XML 等文件样

式的计算机语言。CSS 实现了页面形式和内容的分离,不仅可以静态地修饰网页,还可以配合各种脚本语言动态地对网页中的各元素进行格式化。

通常,应用样式表的网站的用户体验会更快。如果将所有样式保存在一个文件中,可以减少维护的时间、减少错误的机会,从而提高表达的一致性。使用 CSS 的 HTML 或 XHTML 网站更容易调整,以适应不同的浏览器。

3. JavaScript 脚本语言

JavaScript 是一种解释型的、基于对象的、采用事件驱动脚本语言。它由 Netscape 公司开发。其主要工作机制是将 JavaScript 脚本嵌入到 Web 页面中,并随着 HTML 文件一起传送到客户端,由浏览器解释执行。在脚本执行期间无须与服务器交互,可以对用户的操作直接做出响应。

在使用 JavaScript 时,通常用标记 `< script >` 和 `</script >` 界定,将 JavaScript 代码放到 HTML 的 `< head >` 或 `< body >` 部分。

JavaScript 不需要 Java 编译器,由浏览器逐行地解释执行;JavaScript 具有跨平台性,它依赖于浏览器本身,与操作环境无关,只要是能支持 JavaScript 的浏览器就可以正确执行;JavaScript 使网页变得生动;JavaScript 可以使有规律、重复的 HTML 文段简化,减少下载时间;JavaScript 能及时响应用户的操作,对提交表单做即时检查,大大减少了服务器的开销。

1.2.2 服务器端技术

服务器端技术主要是指服务器端的脚本编程技术,常用的服务器端脚本语言有 ASP、NET、PHP、JSP 和 Python 等。它们的共同点是:脚本都运行于服务器端,能够动态地生成网页;脚本运行不受客户端浏览器限制;脚本程序都是将脚本语言嵌入到 HTML 文档中,执行后返回给客户端的是 HTML 代码。

1. ASP.NET

ASP.NET 是 Microsoft 公司在 ASP(Active Server Pages)基础上推出的新一代脚本语言。ASP.NET 是基于 Microsoft.NET 框架的,它建立在公共语言运行库上,可用于在服务器上生成功能强大的 Web 应用程序,为 Web 站点创建动态的、交互的 HTML 页面。ASP.NET 的主要特点如下:

- ASP.NET 采用基于组件的、面向对象的模块化开发模式。
- 更加广泛的底层支持,可以使用 C#、VB 等编程语言作为宿主开发。
- ASP.NET 采用编译后运行的方式,执行效率大幅提高。它将程序第一次运行时编译成 DLL 文件,以后直接执行 DLL 文件,这样速度就变得非常快。
- 支持 WYSIWYG(所见即所得)、拖放控件和自动部署等功能。
- 程序结构清晰:将程序代码和 HTML 标记分开,使程序设计简化,结构更清晰。
- 移植方便:ASP.NET 可以向目标服务器直接复制组件,当需要更新时重新复制一个即可。

2. PHP(HyperText Preprocessor)

PHP 由 PHP 网络小组开发,是免费的开放源代码产品。Apache 和 MySQL 也同样是免费开源,Apache、PHP 和 MySQL 搭配使用可以快速地搭建一套不错的动态网站系统,其执行效率比 IIS+ASP+Access 要高,而后的使用还必须另外付钱给微软。

PHP 是一种嵌入 HTML 页面中的脚本语言,类似于 C、Java 和 Perl,语法简单,是一种解释型语言。它可以跨 UNIX、Linux、Windows 平台使用,具有平台无关性。PHP 非常利于快速开发各种功能不同的定制网站,运行成本低。其不足之处是运行环境的安装、配置比较复杂,同时,由于 PHP 内部结构上的缺陷,使得 PHP 在复杂的大型项目上的开发和维护都比较困难。

3. JSP(Java Server Pages)

JSP 是由 Sun 公司推出的,它是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术。在 HTML 文档中嵌入 Java 程序片段(Scriptlet)和 JSP 标记就形成了 JSP 文件。JSP 脚本在服务器端运行,可以跨 UNIX、Linux、Windows 平台使用,具有平台无关性。JSP 代码须编译成 Servlet 并由 Java 虚拟机执行,编译操作仅在对 JSP 页面的第一次请求时发生。

JSP 最大的好处就是开发效率高,可以使用 JavaBeans 或者 EJB (Enterprise JavaBeans)来执行应用程序所要求的复杂处理,但这种网站架构因为其业务规则代码与页面代码混为一体,不利于维护,因此不适应大型应用的要求,取而代之的是基于 MVC 的 Web 架构。通过 MVC 的 Web 架构可以弱化各个部分的耦合关系,并将业务逻辑处理与页面以及数据分离开来,这样当某个模块的代码改变时,并不影响其他模块的正常运行。因此,不少国外的大型企业系统和商务系统都使用基于 Java 的 MVC 架构,能够支持高度复杂的基于 Web 的大型应用。

4. Python

Python 是由荷兰人 Guido van Rossum 发明、于 1991 年公开发行的—种脚本语言。它是面向对象的、解释型的程序设计语言。Python 是纯粹的自由软件。由于 Python 的简洁性、易读性以及可扩展性,它一经问世后就迅速地流行开来,并被逐渐广泛地应用于系统管理任务的处理和 Web 编程。

Python 具有高度的可阅读性。它使用了限制性很强的语法,其中重要的一项就是缩进规则,通过强制程序员缩进(包括 if、for 和函数定义等所有需要使用模块的地方)使得程序更加清晰和具有可读性。

Python 既支持面向过程的编程也支持面向对象的编程。在“面向过程”的语言中,程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中,程序是由数据和功能组合而成的对象构建起来的。

Python 本身被设计为可扩充的。Python 提供了丰富的 API 和工具,以便程序员能够轻松地使用 C 语言、C++ 等语言来编写扩充模块。Python 编译器本身也可以被集成到其他需要脚本语言的程序内。因此,Python 常被昵称为胶水语言,能够将用其他语言编写的程序进行集成和封装。

1.3 ASP.NET 基础

ASP.NET 是在 Microsoft .NET 框架的基础上构建的,可提供构建企业级 Web 应用程序所需服务的一个 Web 平台。ASP.NET 与 Microsoft .NET 框架平台紧密结合是 ASP.NET 的最大特点。本书使用的是 2015 年 7 月正式发行的 ASP.NET 4.6 与 Visual Studio 2015。

1.3.1 Microsoft .NET Framework

.NET 是一种面向网络、支持各种用户终端的开发平台环境。在 .NET 平台上,不同网站间通过相关协定联系在一起,形成自动交流、协同工作的模式,为系统提供全面的服务。

.NET Framework 是一个集成在 Windows 中的组件,是为其运行的应用程序提供各种服务的托管执行环境,如图 1-6 所示。它主要包括两个组件,即公共语言运行时(Common Language Runtime,CLR)和 .NET Framework 类库(.NET Framework Class Library)。

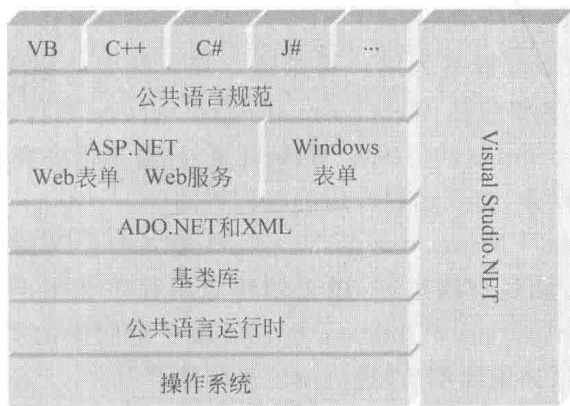


图 1-6 .NET Framework

1. 公共语言运行时(CLR)

CLR 是 .NET Framework 的基础。它为执行用 .NET 脚本语言编写的代码提供了一个运行环境。CLR 管理 .NET 代码的执行,提供内存管理、线程管理、代码执行、安全验证、远程处理等服务,并保证应用和底层操作系统之间必要的分离。同时,CLR 使得开发人员可以调试和进行异常处理。如果要执行这些任务,需要遵循公共语言规范(Common Language Specification,CLS)。CLS 描述了运行库能够支持的数据类型的子集。

在 CLR 监视之下运行的程序属于受控代码,也叫托管代码(Managed codes)。不在 CLR 监视之下、直接在裸机上运行的应用或者组件叫非托管代码(Unmanaged codes)。

2. .NET Framework 类库

.NET Framework 类库是一个与 CLR 紧密集成的、面向对象的、可重用的类型集合。它是生成 .NET 应用程序、组件和控件的基础,包括 .NET 脚本语言、CLS、.NET