



ASP.NET Web 程序设计 (C# 版)

张杰敏 主 编



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高校应用人才培养信息技术类规划教材

ASP. NET Web 程序设计 (C#版)

张杰敏 主 编
魏德志 吴 旭 副主编



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书详细介绍了使用 ASP.NET 进行 Web 应用程序开发的技术。具备程序设计基础知识的读者,通过学习本书能够掌握 .NET 平台下开发 Web 程序的方法。本书共 16 章,内容包括 ASP.NET 的基本概念、C#语言的基础、Visual Studio 2008 的使用、服务器控件、登录控件、导航控件、验证控件、数据控件、ADO.NET 数据库编程、LINQ 查询、AJAX、Web 页面外观设计、异常处理及调试方法、站点部署等。全书以完整的实际开发案例结束,全面展示 ASP.NET 技术应用。

本书适合作为高等院校相关专业的教材或参考书目,对 Web 从业人员同样具有很好的参考价值,同时也是 ASP.NET 兴趣爱好者的一本自学教材。

图书在版编目(CIP)数据

ASP.NET Web 程序设计(C#版)/张杰敏主编. —北京:北京大学出版社,2011.1
(21 世纪全国高校应用人才培养信息技术类规划教材)
ISBN 978-7-301-18276-5

I. ①A… II. ①张… III. ①主页制作—程序设计—高等学校—教材 ②C 语言—程序设计—高等学校—教材 IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字(2010)第 246164 号

书 名: ASP.NET Web 程序设计(C#版)

著作责任者: 张杰敏 主编

策 划 编 辑: 温丹丹

责 任 编 辑: 温丹丹

标 准 书 号: ISBN 978-7-301-18276-5/TP·1140

出 版 发 行: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

电 话: 邮购部 62752015 发行部 62750672 编辑部 62765126 出版部 62754962

网 址: <http://www.pup.cn>

电 子 信 箱: zjy@pup.cn

印 刷 者: 山东省高唐印刷有限责任公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 22 印张 520 千字

2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷

定 价: 40.00 元

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话:010-62752024 电子信箱: fd@pup.pku.edu.cn

前 言

在计算机应用的各个领域, Web 应用技术是发展最为迅速、变化最为活跃的领域之一。各种 Web 开发技术层出不穷, 开发工具、编程语言五花八门。微软公司为了更好地适应市场的发展需求, 推出了全新的 ASP.NET Web 开发技术, 利用 .Net Framework 和 Visual Studio. Net 开发环境, 形成了支持多语言的、通用的运行平台。

ASP.NET 是基于通用语言的编译运行的程序, 它具有功能强大、可移植的特点, 适应几乎全部的运行平台。通用语言的基本库、消息机制、数据接口的处理都能无缝地整合到 ASP.NET 的 Web 应用中。同时, ASP.NET 也是语言独立化的, 使用者可以选择一种最适合自己的语言来编写程序, 也可以选用多种语言编写程序, 方便开发。当前, ASP.NET 支持 C#、VB、JScript、C++、F++ 等语言。本书采用 C#语言来描述程序部分。

C#语言是微软公司在 2000 年 6 月发布的一种新的编程语言, 是微软为 .NET Framework 量身订做的程序语言, 拥有 C/C++ 的强大功能以及 Visual Basic 简易、适用的特性, 是第一个组件导向 (Component-Oriented) 的程序语言, 和 C++ 与 Java 一样, C#是面向对象的 (Object-Oriented) 程序语言。

本着理论与实践有机结合的教学理念, 本书从本科生的实际学习特点和学习能力出发, 较完整地呈现了 ASP.NET 技术。本书的编者具有丰富的教学经验和教材编写经验, 难得的是同时具有良好的实际开发经验, 所述内容较全面地覆盖 ASP.NET 主要技术, 较好地体现了良好的设计理念。重要的知识点都有具体实例或演示支撑, 各章都尽可能以实例贯穿新的知识点, 使理论应用于实践, 以增强理解。本书以一个实际的工程案例作为结束, 展示工程案例开发的完整流程, 体现软件工程的思想, 演示 ASP.NET 技术的综合应用, 对于初涉系统开发的设计者具有较强的实际借鉴意义。

本书在编写过程中尽可能做到简明、实用、新颖, 内容次重分明。通过本书的学习, 凡具备程序设计基础知识的读者不仅能迅速掌握 ASP.NET 与 C#语言的相关技术, 而且对数据库知识、面向对象思想等也将有更好的认识, 能够很容易跨越成长为 .Net Web 程序设计者。

本书每章的组织结构以教学目标开始, 以小结和练习结束。教学目标简要指明该章内容, 有助于读者在学完本章后确定自己是否达到目标。每章在集成、整合、提炼精华的基础上小结。因此, 各章小结不是内容简述或重复, 而是展示了内容之间的逻辑相关性, 对深化理解有很大帮助。练习可以测试和扩充读者所学的知识, 提供实践指导, 宜于读者复习巩固关键概念。

本书共分 16 章:

第 1 章主要介绍 Web 的发展历程和技术类别及 ASP.NET 的基本概念;

第 2 章主要介绍 C#语言的各种变量、运算符和表达式、类处理及命名空间等;

第 3 章介绍 Visual Studio 2008 的环境和构建 ASP.NET 站点;

第 4 章介绍 ASP.NET 控件类型及控件命名规范;

第5章介绍标准控件的功能和用途;

第6章介绍登录控件的功能、用途和使用方法;

第7章介绍导航控件的功能、用途和使用方法;

第8章介绍验证控件的功能、用途和使用方法;

第9章介绍数据控件的基本使用方法及数据绑定语法;

第10章介绍 ADO. NET 数据库编程的基本知识;

第11章介绍 LINQ 的概念,掌握 LINQ 的使用方法和 LinqDataSource 控件的使用;

第12章介绍 AJAX 的概念以及掌握 AJAX 服务器控件的使用;

第13章介绍如何使用 ASP. NET 进行 Web 页面外观设计的方法;

第14章介绍异常处理、调试和跟踪的基本概念和调试方法;

第15章介绍站点部署的方法和原理;

第16章则以一个实验室综合管理系统为案例,完整地介绍了一个系统的设计与实现的过程。

本书由张杰敏任主编,魏德志和吴旭任副主编。其中,张杰敏编写了第1章,魏德志编写了第2、3、6、7、11、12、15章,吴旭编写了第4、5、8、9、10、13、14、16章。

由于时间仓促、水平有限,书中遗漏或错误,请不吝赐教,值此致以诚挚的感谢。

编 者

2010年12月于厦门

目 录

第 1 章 引论	1
1.1 Web 概述	1
1.2 Web 开发技术	4
1.3 ASP.NET 概述	6
1.4 小结	7
1.5 练习	8
第 2 章 C#语言入门	9
2.1 C#语言的基本元素	9
2.1.1 标识符	9
2.1.2 关键字	10
2.1.3 变量	13
2.1.4 常量	14
2.1.5 注释	16
2.1.6 运算符	18
2.2 控制流语句	21
2.2.1 选择语句	21
2.2.2 循环语句	27
2.3 命名空间	30
2.4 类处理	33
2.4.1 类的声明	33
2.4.2 类的成员	34
2.4.3 构造函数和析构函数	36
2.5 变量作用域和生命周期	39
2.6 小结	40
2.7 练习	40
第 3 章 构建 ASP.NET Web 站点	41
3.1 Web 站点的设计原则	41
3.2 用 VS 2008 创建 Web 站点	42
3.2.1 用 VS 2008 创建网站	43
3.2.2 VS 2008 的基本环境说明	44
3.3 操作 Web 站点中的文件	45
3.3.1 创建 ASP.NET Web 页面	45
3.3.2 ASP.NET Web 页面的运行	45
3.4 ASP.NET 页面结构介绍	46

3.4.1	内联编码模型	46
3.4.2	后台编码模型	47
3.4.3	页面指令	47
3.4.4	ASP.NET 页面事件	48
3.4.5	页面生命周期	49
3.5	ASP.NET 应用程序文件夹	50
3.6	创建简单的用户登录 Web 站点	52
3.7	小结	54
3.8	练习	54
第 4 章	ASP.NET 控件	55
4.1	HTML 服务器控件	55
4.2	Web 服务器控件	56
4.3	Web 服务器控件和 HTML 服务器控件的区别	57
4.4	控件的通用提示	58
4.4.1	控件的命名规范	58
4.4.2	一些有用的经验	61
4.5	小结	61
4.6	练习	62
第 5 章	标准控件	63
5.1	标准控件概览	63
5.2	文本类控件	64
5.2.1	Label 控件	64
5.2.2	Literal 控件	65
5.2.3	TextBox 控件	66
5.3	按钮类控件	69
5.3.1	Button 控件	69
5.3.2	LinkButton 控件	72
5.3.3	ImageButton 控件	73
5.4	列表框类控件	73
5.4.1	DropDownList 控件	74
5.4.2	ListBox 控件	77
5.5	选择框类控件	80
5.5.1	CheckBox 控件	80
5.5.2	CheckBoxList 控件	81
5.5.3	RadioButton 控件	82
5.5.4	RadioButtonList 控件	82
5.6	图像类控件	83
5.6.1	Image 控件	83
5.6.2	ImageMap 控件	84
5.7	其他控件	84
5.7.1	HyperLink 控件	84

5.7.2	Table 控件	85
5.7.3	BulletedList 控件	88
5.7.4	HiddenField 控件	90
5.7.5	Calendar 控件	90
5.7.6	AdRotator 控件	95
5.7.7	FileUpload 控件	95
5.7.8	Wizard 控件	97
5.7.9	XML 控件	98
5.7.10	MultiView 和 View 控件	99
5.7.11	Panel 控件	99
5.7.12	PlaceHolder 控件	100
5.7.13	Substitution 控件	101
5.7.14	Localize 控件	101
5.8	小结	101
5.9	练习	102
第 6 章	登录控件	104
6.1	身份验证和授权	104
6.1.1	身份验证	104
6.1.2	授权	105
6.2	使用 Web 站点管理工具	105
6.2.1	“安全”选项卡	106
6.2.2	“应用程序”选项卡	111
6.2.3	“提供程序”选项卡	112
6.2.4	网站管理工具的工作原理	112
6.3	登录控件的使用	113
6.3.1	Login 类控件	113
6.3.2	用户管理类控件	116
6.4	登录的安全性	117
6.4.1	设置身份验证的方法	117
6.4.2	设置保护特定的文件和文件夹	120
6.4.3	通过 IIS 进行保护	120
6.5	小结	121
6.6	练习	121
第 7 章	导航控件	122
7.1	基于 XML 的 Web 站点地图	122
7.2	SiteMapPath 控件	124
7.2.1	使用 SiteMapPath 控件	124
7.2.2	SiteMapPath 控件的常用属性	125
7.2.3	SiteMapPath 控件的子元素	126
7.3	Menu 控件	127
7.3.1	使用 Menu 控件	127

7.3.2 设计 Menu 控件的外观	129
7.4 TreeView 控件	131
7.4.1 使用静态的 TreeView 控件导航	131
7.4.2 使用数据源绑定的 TreeView 控件导航	133
7.4.3 设计 TreeView 控件导航的外观	134
7.5 小结	135
7.6 练习	135
第 8 章 验证控件	136
8.1 有效性验证概述	136
8.2 客户端和服务端端的验证	136
8.2.1 客户端数据的有效性验证	137
8.2.2 服务器端数据的有效性验证	137
8.3 ASP.NET 服务器验证控件	138
8.3.1 服务器验证控件概述	138
8.3.2 RequiredFieldValidator 控件	139
8.3.3 CompareValidator 控件	143
8.3.4 RangeValidator 控件	146
8.3.5 RegularExpressionValidator 控件	149
8.3.6 CustomValidator 控件	151
8.3.7 ValidationSummary 控件	152
8.4 小结	155
8.5 练习	156
第 9 章 数据控件和数据绑定	157
9.1 数据源控件	157
9.1.1 数据源控件概述	157
9.1.2 SqlDataSource 控件	158
9.1.3 AccessDataSource 控件	162
9.1.4 LinqDataSource 控件	163
9.1.5 ObjectDataSource 控件	163
9.1.6 XmlDataSource 控件	163
9.1.7 SiteMapDataSource 控件	164
9.2 数据绑定控件	164
9.2.1 GridView 控件	164
9.2.2 DataList 控件	170
9.2.3 DetailsView 控件	170
9.2.4 FormView 控件	172
9.2.5 Repeater 控件	173
9.2.6 DataPager 控件	173
9.2.7 ListView 控件	173
9.3 配置数据源控件的高速缓存	174
9.3.1 ASP.NET 高速缓存	174

9.3.2 配置数据源控件的高速缓存	175
9.4 存储连接信息	175
9.5 内部数据绑定语法	176
9.5.1 DataBinder 语法的变化	176
9.5.2 XML 数据绑定	177
9.6 表达式和表达式构建器	177
9.7 小结	181
9.8 练习	181
第 10 章 ADO.NET 数据库编程	182
10.1 ADO.NET 编程	182
10.1.1 ADO.NET 对象	182
10.1.2 参数化 SQL 语句	190
10.2 数据源控件编程	191
10.2.1 数据源视图	192
10.2.2 与数据绑定控件的交互	193
10.2.3 层次性数据源视图	193
10.2.4 数据源控件代码	194
10.3 存储过程	210
10.4 小结	213
10.5 练习	214
第 11 章 LINQ 查询	215
11.1 LINQ 查询基础知识	215
11.1.1 LINQ 查询简介	215
11.1.2 LINQ 查询数据源	216
11.1.3 LINQ 查询语法	217
11.1.4 LINQ 查询操作	218
11.2 LINQ to SQL	219
11.2.1 使用 LINQ to SQL 进行操作的方法	220
11.2.2 使用 LINQ to SQL 的步骤	221
11.2.3 使用 LINQ to SQL 的例子	222
11.3 LINQ to XML	230
11.3.1 LINQ to XML 概述	231
11.3.2 LINQ to XML 的操作	231
11.4 使用 LinqDataSource 控件	235
11.4.1 连接到数据库的方法	235
11.4.2 连接到内存集合的方法	236
11.4.3 LinqDataSource 控件与其他数据源控件进行比较	236
11.4.4 使用 LinqDataSource 控件的例子	237
11.5 小结	242
11.6 练习	242

第 12 章	AJAX	243
12.1	AJAX 概述	243
12.1.1	AJAX 的工作方式	244
12.1.2	AJAX 的优势	244
12.1.3	AJAX 的缺陷	245
12.1.4	ASP.NET AJAX 简介	246
12.2	AJAX 服务器控件	247
12.2.1	ScriptManager 控件	247
12.2.2	UpdatePanel 控件	250
12.2.3	UpdateProgress 控件	253
12.2.4	Timer 控件	255
12.3	小结	257
12.4	练习	257
第 13 章	Web 页面外观设计	258
13.1	用母版页创建 Web 页面布局	258
13.1.1	母版页概述	258
13.1.2	母版页的建立	259
13.2	皮肤的使用	262
13.2.1	皮肤的概念	262
13.2.2	皮肤文件的建立	262
13.3	CSS 在 Web 页面上的应用	265
13.3.1	CSS 概述	265
13.3.2	CSS 的优点	266
13.3.3	CSS 与 Web 页面结合的方法	266
13.4	主题的使用	267
13.4.1	主题	267
13.4.2	主题与 CSS、母版页、皮肤的区别与关联	267
13.5	小结	268
13.6	练习	268
第 14 章	异常处理、调试和跟踪	269
14.1	异常处理	269
14.1.1	异常处理功能	269
14.1.2	C#异常处理	270
14.2	调试基础知识	272
14.2.1	断点	272
14.2.2	断点与跟踪点	274
14.3	调试客户端脚本	274
14.3.1	在客户端脚本中设置断点	274
14.3.2	增强的脚本调试错误信息	276
14.4	跟踪 ASP.NET 页面	283

14.4.1	页面级的配置	284
14.4.2	应用程式级的配置	284
14.4.3	ASP.NET 的跟踪模式的输出	285
14.4.4	编写跟踪消息	286
14.5	小结	287
14.6	练习	287
第 15 章	部署 Web 站点	288
15.1	部署清单	288
15.2	Web 站点的预编译	289
15.2.1	就地预编译	289
15.2.2	针对部署的预编译	289
15.2.3	默认编译	290
15.2.4	预编译网站以进行部署	290
15.3	部署方法	291
15.3.1	采用复制网站进行部署	291
15.3.2	采用发布网站进行部署	293
15.4	提高 Web 站点性能	294
15.5	小结	297
15.6	练习	297
第 16 章	实验室综合管理系统的设计与实现	298
16.1	系统需求分析	298
16.2	系统的总体设计	299
16.2.1	系统框架与功能结构设计	299
16.2.2	业务流程分析	302
16.2.3	数据库设计	302
16.3	系统通用性基础设计	307
16.3.1	系统管理	307
16.3.2	部门管理	308
16.3.3	设备分类管理	308
16.3.4	设备型号管理	309
16.3.5	设备基本信息管理	310
16.4	设备管理模块的设计与实现	310
16.4.1	实验设备管理模块的功能设计	310
16.4.2	设备检修信息管理子功能的实现	310
16.5	预约管理模块的设计与实现	321
16.5.1	预约管理模块的功能设计	322
16.5.2	预约申请子功能的实现	322
16.5.3	预约审批子功能的实现	325
16.6	查询统计服务模块的设计与实现	330
16.6.1	查询统计服务模块的功能设计	330

16.6.2 查询设备信息子功能模块的实现	330
16.6.3 设备资产统计子功能模块的实现	337
16.7 小结	338
参考文献	340

第 1 章 引 论

本章教学目标

- 了解 Web 的发展历程；
- 了解 Web 技术的类别；
- 了解什么是 ASP.NET 技术。

WWW (World Wide Web) 常简称为 Web, 也称 W3, 中文译名叫万维网。Web 是一个以 Internet 为基础的计算机网络, 是一种遍布在 Internet 上的全球性的超媒体系统。Web 允许人们浏览和获取分布于整个互联网的大量超媒体 (Hypermedia) 信息, 内容包括文字、图形、图像、声音、动画、资料库, 以及各式各样的软件等。Web 的建立意味着全球信息系统和环球信息通道的建立和应用。本章将简单介绍 Web 发展历程, 概述 Web 开发技术, 并概要介绍 ASP.NET 技术。

1.1 Web 概述

1946 年 2 月, 世界上第一台电子数字计算机 ENIAC (The Electronic Numerical Integrator And Computer) 研制成功 (参见图 1-1)。1969 年 12 月, 世界上第一个计算机网络 ARPANET 建立并投入运行。由 ARPANET 发展和演化而成的 Internet (参见图 1-2) 是全球最大的、开放的、连接众多不同网络的互联网。1993 年 4 月, 欧洲核研究理事会公布了 Web 的代码。之后, 美国网景公司推出了 Web 产品, 标志着 Web 的诞生。

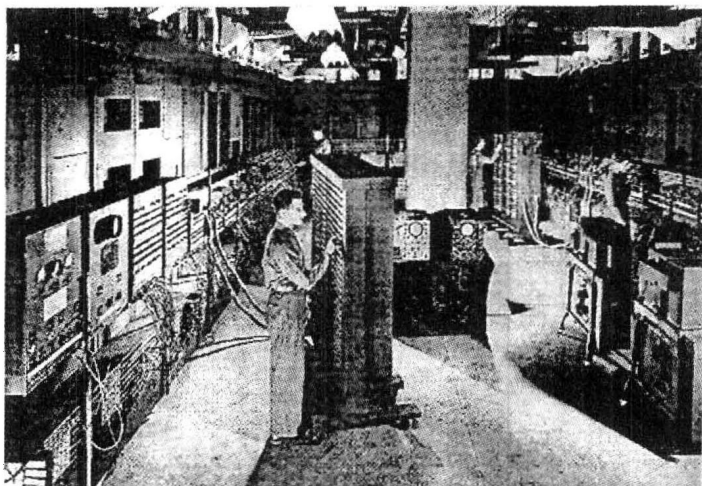


图 1-1 世界上第一台计算机 ENIAC

可以说, 计算机改变了人类进步的历史进程, Internet 改变了计算机应用的历史进程, 而 Web 改变了 Internet 应用的历史进程, 进而深刻地影响了人类生活。

如今,世界各地的人们通过各种不同的浏览器在 Web 网上查阅资料、发送邮件、传输文档、举行视频会议、聊天、购物、听音乐、看电视等(参见图 1-3)。Web 为全球信息的交流和传播带来了革命性的变化,是快速、有效、方便的环球信息通道。今天,人类的日常活动对 Web 是如此的依赖,使之成为人类文明史上不可或缺的组成部分。

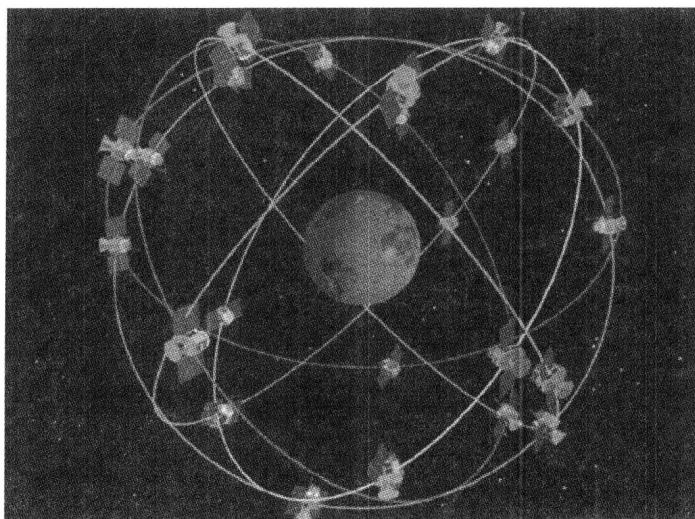


图 1-2 Internet 互联世界



图 1-3 多彩的 Web 世界

自 Web 技术诞生以来,它就给人们的生活带来了巨大的变革。无论是在便利信息传递、丰富娱乐休闲,还是在提高管理效率、加速科技研发等方面,Web 所起的作用都是无

法估量和形容的。无数科技人员及爱好者推动着 Web 技术的发展,造就了神奇、伟大的 Web 世界。在具体介绍 ASP.NET 之前,简单回顾一下 Web 历程,是对曾经为之付出努力的人们应有的尊重,同时也会增进读者对 Web 技术 ASP.NET 的了解。

1991 年 5 月 WWW 在 Internet 上首次露面,便立即引起轰动,获得了极大的成功,被广泛推广应用。1993 年 4 月,欧洲核研究理事会公布了 Web 的代码。在此之前,人们对什么是 Web 几乎一无所知。这一切变化离不开一个人,他就是 Tim Berners-Lee。

Tim 原是位于瑞士日内瓦的欧洲核能研究中心粒子实验室的一名科学家,该研究中心的实验室遍布很多国家,大量试验结果需要从世界各地的实验室汇集到日内瓦,由于各地相关部门使用不同的文档操作系统甚至不同的计算机,没有适用于文档处理的通用格式,致使共享信息成为既耗时又费力的事情。于是, Tim 想创造一种分布的、独立于平台的全新信息发布系统,利用 Internet 使所有的用户都能不受软、硬件限制地访问系统,共享分布在各国各地的实验室、研究所的各种最新信息(包括数据、图像等资料),支持各国的核物理学家通过网络及时交流并传递信息,进行合作研究。结果是, Tim 发明了 3 项技术,这成为 Web 的基础技术。

第 1 项基础技术,是用于编写通用可读文档的超文本标记语言(HyperText Mark-up Language, HTML),通过它可以很容易地把相关内容写入文档或把资源联结起来。第 2 项基础技术,是通过 Internet 引用其他可访问文档或资源的统一资源定位符(Uniform Resource Locator, URL)。URL 也称为 Web 页面地址,它是 Internet 上标准的资源地址,它既能指向其他 HTML 文档,也能指向任何类型的 Internet 资源,特别是指向从其他 Internet 服务(如 FTP 或 E-mail)得到的资源。第 3 项基础技术,是用于发布资源的超文本传输协议(HyperText Transfer Protocol, HTTP),它是 Web 服务器传输超文本到本地浏览器的传送协议。HTTP 可以使浏览器更加高效,使网络传输量减少。HTTP 不仅保证计算机正确快速地传输超文本文档,还确定传输文档中的内容显示方式,如文本先于图形显示等。这也就是在浏览器中所看到的 Web 页面地址都是以“http://”开头的原因。

1994 年 10 月, Tim 联合 CERN (European Organization for Nuclear Research)、DARPA (Defense Advanced Research Projects Agency) 和欧盟倡导并组织成立了 Web 的核心技术机构 W3C (World Wide Web Consortium),它是一个非赢利组织,会员包括生产技术产品的厂商、服务提供商、内容供应商、团体用户、研究实验室、标准制定机构和政府部门等,致力于在 Web 发展方向上达成共识,关注技术标准的规范化,并指导技术开发、普及、推广和培训等。从 Web 诞生至今,它的每一步发展都离不开 W3C 的努力。我国的 W3CHINA 建立于 2003 年 6 月 20 日,旨在为 Web 工作者提供一个学习、研究、讨论 W3C 标准及 Web 技术的平台。

国际 W3C 很早就意识到,由于 HTML 规则的限制,光靠它不能解决所有数据解释的问题,例如视频文件、特殊公式、音频文件等其他型态的内容。另外,HTML 的扩充性、弹性、易读性均差强人意。为了解决以上问题,专家们使用标准通用标记语言(Standard Generalized Markup Language, SGML)精简制作,并依照 HTML 的发展经验,产生出一套规则严谨却又简单易用的描述数据语言(Extensible Markup Language, XML)。XML 从 1996 年开始具备雏形,并向 W3C 提案,在 1998 年 2 月发布为 W3C 的标准(XML1.0)。从技术角度看,XML 语言统一了信息的表达方式,但离揭示信息内容的目标还相距甚远。

之后, W3C 和一些研究机构开始对元数据 Metadata (描述数据的数据, 可以揭示信息的内容特性) 进行研究。1999 年, Netscape 提出的 RSS (Rich Site Summary) 建议标准是用元数据技术描述新闻等信息内容的第一次尝试。随后, W3C 的研究小组提出了 RDF (Resource Description Framework) 标准草案。RDF 在 XML 语法的基础上, 规定了元数据的存储结构和相关的技术标准。使用 RDF 语言可以用统一的、可交换的格式揭示出信息本身的各种特性。

到了 2000 年以后, 随着 Web 应用的日益复杂, 人们逐渐意识到, 单纯依靠某种技术多半无法达到快速开发、快速验证和快速部署的最佳境界。研究者开始尝试着将已有的 Web 开发技术综合起来, 形成完整的开发框架或应用模型, 并以此来满足各种复杂的应用需求。2001 年, W3C 又开始着手制订 OWL (OWL Web Ontology Language) 标准。OWL 语言也是一种符合 XML 标准的语言, 它比 RDF 又前进了一步, 可以更加深入、细致地描述信息内容。在 RDF 和 OWL 语言的帮助下, Web 上的信息内容变得更容易理解、更便于交换, 并支持共享。2003 年, W3C 成立了语义化 Web Service 研究小组 (Semantic Web Services Interest Group), 研究在 Web Service 中加入语义技术的相关问题。2004 年 2 月, W3C 宣布 RDF 和 OWL 标准正式成为 W3C 的建议方案, 这标志着语义化 Web 的大厦已经破土动工。2005 年, W3C 提出将 XML: ID 作为唯一识别 XML 文档各个部分的标准方式。W3C 组织认为, 新标准有希望使用户更容易地完成 XML 写作。2008 年 1 月, HTML 5 规范的第一份正式草案公布。至 2010 年, HTML 5 规范仍在制订完善中。关于 HTML 5 成为 W3C 标准的时间, 众说纷纭。其中, IanHixie 的说法最受认可, “根据估计, HTML 5 直到 2022 年才会形成 W3C 的推荐标准”。HTML 5 提供了一些新的元素和属性, 将有利于搜索引擎的索引整理, 同时更好地帮助小屏幕装置和视障人士使用。除此之外, 还为提供了新的功能, 如 <audio>、<video> 和 <canvas> 等标记, 以支持音频、视频和图形图像的元素的使用。尽管标准的形成尚待时日, 但浏览器厂商已经在实现各种 HTML 5 功能。事实上, W3C 标准每年都会有一些新的进展或者变化。只是从 2005 年至今, 并没有明显的、对用户影响深远的新标准。

随着语义化 Web 的诞生和发展, Web 开发技术也必将经历更为重大的变革。可以预见在未来的几年里, 还会有更多新的开发技术或开发平台出现。为了让更多的人获得更有价值的信息服务, Web 开发者们也许还会经历一次又一次的技术浪潮, 还会面临更为严峻的技术挑战。

1.2 Web 开发技术

Web 作为分布式应用架构的特点, 决定它的每一次信息交换都要涉及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器端技术两大类。

1. Web 客户端技术

Web 客户端主要负责展现信息内容, 而 HTML 语言正是用于信息展现的最有效载体之一。最初的 HTML 语言只能在浏览器中展现静态的文本或图像信息, 它满足不了人们对