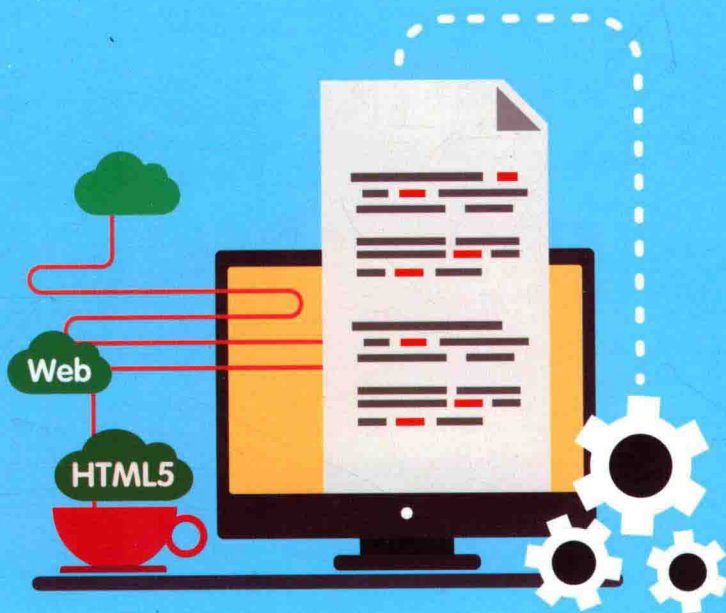


JIYU HTML5 DE WEB SHEJI YU QIANDUAN KAIFA YANJIU FENXI

基于 HTML5 的 Web 设计与前端开发研究分析

李念 著



电子科技大学出版社

国家图书馆(CIP)数据

第一版(2014) 第1次印刷 9787302330117

定价: 49.00元

ISBN 978-7-302-33011-7

清华大学出版社 北京 100084

01-01-01

2014年10月第1次印刷

JIYU HTML5 DE WEB SHEJI YU QIANDUAN KAIFA YANJIU FENXI

基于HTML5的Web设计与前端开发研究分析

李念 著



电子科技大学出版社

图书在版编目(CIP)数据

基于HTML5的Web设计与前端开发研究分析/李念著. —成都: 电子科技大学出版社, 2017.1
ISBN 978-7-5647-4171-6

I. ①基… II. ①李… III. ①超文本标记语言 - 程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第012623号

基于HTML5的Web设计与前端开发研究分析

李念 著

策划编辑 李述娜

责任编辑 李述娜

出版发行 电子科技大学出版社

成都市一环路东一段159号电子信息产业大厦九楼 邮编 610051

主 页 www.uestcp.com.cn

服务电话 028-83203399

邮购电话 028-83201495

印 刷 北京一鑫印务有限责任公司

成品尺寸 170mm × 240mm

印 张 18

字 数 337千字

版 次 2017年1月第一版

印 次 2017年1月第一次印刷

书 号 ISBN 978-7-5647-4171-6

定 价 63.00元

版权所有，侵权必究

前言



随着 HTML5 和 ECMAScript 6 的正式发布，大量的前端业务逻辑极大地增加了前端的代码量，前端代码的模块化、按需加载和依赖管理势在必行，因此，Web 前端越来越受人们重视。

随着 Web 技术的迅速普及应用，我国互联网行业的发展呈现迅猛的增长势头。互联网行业对网站开发、设计制作的人才需求随之增加，在近年来，Web 前端开发工程师这一职业在国内已受到高度重视，这方面的专业人才备受青睐。Web 前端开发在产品开发环节中的作用越来越重要，所以培养具有扎实功底的 Web 前端开发人才任重而道远。

Web 前端是网站开发的首要部分，前端开发的好坏直接影响到整个网站的交互效果。Web 前端开发是从 Web1.0 时代的网页制作演变而来的，之前使用 Dreamweaver 和 Photoshop 就可以方便地制作网页。进入到 Web2.0 时代，如果要想让网页的内容更加生动，提供更多交互形式的用户体验，以满足企业级别的需求，就需要更多的 Web 前端开发技术，其中包括 HTML、CSS、JavaScript、DOM、jQuery 等。HTML、CSS 和 JavaScript 技术是所有网页技术的技术和核心。无论是在互联网上进行发布，还是编写可交互的应用程序，都离不开它们的综合应用。

近几年，随着宽带无线移动通信技术的发展，移动互联网得到了迅速发展。继计算机、互联网之后，移动互联网正掀起第三次信息技术革命的浪潮。IT 领域的新技术、新应用不断涌现，Web 应用技术的不断创新，并且已经渗透到社会生活的方方面面，从而孕育了巨大的市场潜力和发展机会。

与此同时，HTML5 和 CSS3 进一步结合，HTML5 首先强化 Web 网页的表现能力，其次追加了本地数据库等相关功能。所谓的 HTML5 实际上是指包括 HTML、CSS 和 JavaScript 在内的一套技术组合。目前支持 HTML5 的主流浏览器有 Google Chrome、Firefox、IE 9 以上的版本的浏览器。HTML5 新增的视频、音频、画布、离线应用等功能为移动 Web 开发带来了便利。新技术不仅能很好地使用移动终端几面，而且很大程度上减少了代码冗余，提高了移动社会运行效

率。随着 HTML5 和 CSS3 的发展，基于 HTML5 的应用会对移动互联网领域产生巨大的影响。

在时代的背景下，我们有必要需研究新的技术和新的知识，以适应时代的发展要求。本书通过对 Web 前端开发的发展与未来趋势的研究分析，梳理 HTML、CSS、JavaScript、DOM、jQuery 等一系列必要的技术，并通过研究最新修订的超文本标记语言——HTML5，对移动 Web 应用与 PC 端 Web 应用进行分析论证，从而了解 Web 前端的发展趋势。

在本书的编写中未列出的引用文献和论著，我们深表歉意，并同样表示感谢。

由于时间的仓促，编者水平有限，难免存在不足之处。在本书出版之际，我们真诚的希望读者对本书提出宝贵的意见和建议。

目录



第一章 开发简介	001
第一节 Web 设计开发的背景及发展研究	001
一、客户端技术的萌芽和演进	002
二、服务端技术的成熟与发展	004
三、Web 开发框架和应用模型	006
第二节 基于 HTML5 的 Web 开发技术概述	009
一、Web 开发技术发展	009
二、Web 前端技术及开发概述	012
第三节 基于 HTML5 的 Web 前端开发工具分析	015
一、EditPlus	015
二、TextPad	016
三、TopStyle	016
四、CSS3 Menu	016
五、Sothink Tree Menu	017
六、ColorImpact	017
第二章 HTML5 与 HTML 的对比解析	019
第一节 HTML 文档结构的研究及注意事项	019
第二节 HTML 中常见模块解析	023
一、HTML 中的表格	023
二、框架	034
三、表单的概述	039
第三节 HTML5 新增内容概述	040
一、新的 Doctype	040
二、图形元素 (The Figure Element)	040
三、<small> 重新定义	041

四、脚本 (scripts) 和链接 (links) 无须 type	041
五、引号还是不要引号	041
六、内容可编辑	041
七、Email 输入 (Inputs)	042
八、占位符 (Placeholders)	042
九、本地存储 (Local Storage)	043
十、语义的 Header 和 Footer	044
十一、IE 和 HTML5(Internet Explorer and HTML5)	044
十二、文档某一部分的信息 (hgroup)	045
十三、必要的属性 (Required Attribute)	045
十四、Autofocus 属性	046
十五、Audio 支持	046
十六、Video 支持	047
十七、显示控制条	047
十八、正则表达式	047
十九、属性支持检测	048
二十、mark 元素 (Mark Element)	048
二十一、data 属性 (The Data Attribute)	048
二十二、Output 元素	049
二十三、使用区域 input 创建滑块 (Create Sliders with the Range Input)	049
第四节 HTML 表单及 HTML5 表单变化	051
一、HTML 中表单的概述	051
二、HTML5 表单设计分析	061
三、输入型控件	061
四、表单新特性和函数	062
五、表单验证	064
第五节 HTML5 对于 HTML4 的优势分析	068
一、HTML5 的优势	068
二、HTML5 与 SEO	070
三、HTML5 应用分析	070

第三章 基于 HTML5 的图形图像协同处理技术的实现与应用 ◇◇◇◇ 072

第一节 HTML5 的图形图像协同处理技术的背景	072
--------------------------------	-----

[illegible]

三、浏览器 bug	122
四、CSS3 box-shadow 兼容 loading 效果	124
第四节 对于 CSS 的应用示例解析	131
一、前期准备	131
二、页面布局设计	133
第五章 在 Web 开发中对于 JavaScript 的研究分析 ◇◇◇◇◇◇◇◇◇◇	148
第一节 JavaScript 发展研究及基础语法作用分析	148
一、JavaScript 概述	148
二、基本语法	150
第二节 针对 JavaScript 中运算符及正则表达式的研究	159
一、运算符	159
二、正则表达式	164
第三节 JavaScript 中事件的研究分析	167
一、JS 自定义事件	167
二、DOM 自定义事件	173
三、伪 DOM 自定义事件	175
第四节 对于 JavaScript 中函数与对象的解析	181
一、函数	181
二、JavaScript 对象	188
第六章 基于 jQuery 框架的 Web 前端系统构建的研究与应用 ◇◇◇◇	191
第一节 JavaScript 中 jQuery 的概述	191
一、jQuery 简介	191
二、jQuery 的特点	191
第二节 jQuery 强大功能解析	192
一、jQuery 包装集	192
二、Dom 对象与 jQuery 对象的转换	193
三、jQuery 选择器	193
四、jQuery 选择器全解	195
五、jQuery 遍历	198
第三节 jQueryAJAX 技术开发与应用研究	203
一、原始 Ajax 与 jQuery 中的 Ajax 的比较分析	203

二、jQuery Ajax 详解	206
三、Ajax 相关函数	211
四、全局 Ajax 事件	214

第七章 基于 HTML5 的全景漫游系统制作平台的设计与研究 ◇◇◇◇ 217

第一节 基于 HTML5 的全景漫游系统制作平台的设计背景及技术	217
一、基于 HTML5 的全景漫游系统制作平台的设计背景	217
二、关键技术研究与应用	219
三、HTML5 技术	221
第二节 平台的需求分析和总体设计	224
一、需求分析	224
二、总体设计与规划	225
第三节 平台详细设计与实现	228
一、全景场景模块	228
二、地图导航模块	233
三、菜单导航模块	235
四、终端全景推送模块	238
第四节 基于全景漫游平台的案例制作与测试	240
一、全景漫游案例制作	240
二、全景漫游案例部署测试	241

第八章 移动 Web 开发与 PC 版 web 异同的研究分析 ◇◇◇◇◇◇◇◇ 243

第一节 移动 Web 开发模式分析	243
一、智能手机与手机浏览器	243
二、智能手机浏览器	244
三、移动 Web 应用	245
四、移动 Web 应用开发注意事项	248
第二节 对于移动 Web 开发技术研究	250
一、响应式 Web 设计	250
二、移动 Web 设计中 CSS3 属性分析	251
三、移动 Web 开发的问题解决分析	252
第三节 移动 Web 开发框架研究	255
一、移动 Web 开发框架的主要特点:	255

二、Web 移动开发 JavaScript 框架	255
第四节 移动 Web 开发与 pc 版 Web 的差异分析	258
第九章 基于 HTML5 的 Web 开发前景分析	261
第一节 Web 前端开发发展趋势分析	261
一、Web 前端发展历程	261
二、Web 规范和标准	264
三、生态的自我完善和自我拓展	265
四、Mobile 的发展驱动着战场的转移	266
五、端的融合	267
六、栈的融合	267
七、后端服务化，云数据，云安全	268
第二节 Web 前端开发职业规划发展趋势	270
一、职业方向定位	270
二、职业发展目标	271
参考文献	275

第一章 开发简介

第一节 Web 设计开发的背景及发展研究

众所周知, Web 这个 Internet 上最热门的应用架构是由 Tim Berners-Lee 发明的。Web 的前身是 1980 年 Tim Berners-Lee 负责的 Enquire (Enquire Within Upon Everything 的简称) 项目。1990 年 11 月, 第一个 Web 服务器 nxoc01.cern.ch 开始运行, Tim Berners-Lee 在自己编写的图形化 Web 浏览器 “WorldWideWeb” 上看到了最早的 Web 页面。1991 年, CERN (European Particle Physics Laboratory) 正式发布了 Web 技术标准。目前, 与 Web 相关的各种技术标准都由著名的 W3C 组织 (World Wide Web Consortium) 管理和维护。

从技术层面看, Web 架构的精华有三处: 用超文本技术 (HTML) 实现信息与信息的连接; 用统一资源定位技术 (URI) 实现全球信息的精确定位; 用新的应用层协议 (HTTP) 实现分布式的信息共享。这三个特点无一不与信息的分发、获取和利用有关。其实, Tim Berners-Lee 早就明确无误地告诉我们: “Web 是一个抽象的 (假想的) 信息空间。” 也就是说, 作为 Internet 上的一种应用架构, Web 的首要任务就是向人们提供信息和信息服务。

很可惜, 在 Web 应用日新月异的今天, 许多搞技术的人似乎已经忘记了 Web 架构的设计初衷。他们在自己开发的网站或 Web 应用中大肆堆砌各种所谓的 “先进技术”, 但最终用户能够在这些网站或应用中获得的有价值信息却寥寥无几。这个问题绝不像评论者常说的 “有路无车” 或 “信息匮乏” 那么简单。一个 Web 开发者倘若忘记了 Web 技术的最终目标是提供信息和信息服务, 他的愚蠢程度就丝毫不亚于一个在足球场上只知道卖弄技巧, 却忘记了射门得分的大牌球星。从这个角度来说, 评价一种 Web 开发技术优劣的标准只有一个, 那就是看这种技术能否在最恰当的时间和最恰当的地点, 以最恰当的方式, 为最需要信息的人提供最恰当的信息服

一、客户端技术的萌芽和演进

Web 是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。我们先来谈谈客户端技术的萌芽和演进过程。

Web 客户端的主要任务是展现信息内容, 而 HTML 语言则是信息展现的最有效载体之一。作为一种实用的超文本语言, HTML 的历史最早可以追溯到 20 世纪 40 年代。1945 年, Vannevar Bush 在一篇文章中阐述了文本和文本之间通过超级链接相互关联的思想, 并在文中给出了一种能实现信息关联的计算机 Memex 的设计方案。Doug Engelbart 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。1969 年, IBM 的 Charles Goldfarb 发明了可用于描述超文本信息的 GML (Generalized Markup Language) 语言。1978—1986 年间, 在 ANSI 等组织的努力下, GML 语言进一步发展成为著名的 SGML 语言标准。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应用系统时, Tim Berners-Lee 意识到, SGML 是描述超文本信息的一个上佳方案, 但美中不足的是, SGML 过于复杂, 不利于信息的传递和解析。于是, Tim Berners-Lee 对 SGML 语言做了大刀阔斧的简化和完善。1990 年, 第一个图形化的 Web 浏览器 “WorldWideWeb” 终于可以使用一种为 Web 度身定制的语言——HTML 来展现超文本信息了。

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, 这满足不了人们对信息丰富性和多样性的强烈需求——这件事情最终的结果是, 由静态技术向动态技术的转变成为 Web 客户端技术演进的永恒定律。

能存储、展现二维动画的 GIF 图像格式早在 1989 年就已发展成熟。Web 出现后, GIF 第一次为 HTML 页面引入了动感元素。但更大的变革来源于 1995 年 Java 语言的问世。Java 语言天生就具备的平台无关的特点, 让人们一下子找到了在浏览器中开发动态应用的捷径。1996 年, 著名的 Netscape 浏览器在其 2.0 版中增加了对 JavaApplets 和 JavaScript 的支持。Netscape 的冤家对头, Microsoft 的 IE 3.0 也在这一年开始支持 Java 技术。现在, 喜欢动画、喜欢交互操作、喜欢客户端应用的开发人员可以用 Java 或 JavaScript 语言随心所欲地丰富 HTML 页面的功能了。顺便说一句, JavaScript 语言在所有客户端开发技术中占有非常独特的地位: 它是一种以脚本方式运行的, 简化了的 Java 语言, 这也是脚本技术第一次在 Web 世界里崭露头角。为了用纯 Microsoft 的技术与 JavaScript 抗衡, Microsoft 还为 1996 年的 IE 3.0 设计了另一种后来也声名显赫的脚本语言——VBScript 语言。

真正让 HTML 页面又酷又炫、动感无限的是 CSS (Cascading Style Sheets) 和 DHTML (DynamicHTML) 技术。1996 年底, W3C 提出了 CSS 的建议标准, 同年, IE 3.0 引入了对 CSS 的支持。CSS 大大提高了开发者对信息展现格式的控制能力。1997 年的 Netscape 4.0 不但支持 CSS, 而且增加了许多 Netscape 公司自定义的动态 HTML 标记, 这些标记在 CSS 的基础上, 让 HTML 页面中的各种要素 "活动" 了起来。1997 年, Microsoft 发布了 IE 4.0, 并将动态 HTML 标记、CSS 和动态对象模型 (DHTML Object Model) 发展成了一套完整、实用、高效的客户端开发技术体系, Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果, DHTML 技术无须启动 Java 虚拟机或其他脚本环境, 可以在浏览器的支持下, 获得更好地展现效果和更高的执行效率。今天, 已经很少有哪个 HTML 页面的开发者还会对 CSS 和 DHTML 技术视而不见了。

为了在 HTML 页面中实现音频、视频等更为复杂的多媒体应用, 1996 年的 Netscape 2.0 成功地引入了对 QuickTime 插件的支持, 插件这种开发方式也迅速风靡了浏览器的世界。在 Windows 平台上, Microsoft 将客户端应用集成的赌注押到了 20 世纪 90 年代中期刚刚问世的 COM 和 ActiveX 身上。1996 年, IE 3.0 正式支持在 HTML 页面中插入 ActiveX 控件的功能, 这为其他厂商扩展 Web 客户端的信息展现方式开辟了一条自由之路。1999 年, Realplayer 插件先后在 Netscape 和 IE 浏览器中取得了成功, 与此同时, Microsoft 自己的媒体播放插件 Media Player 也被预装到了各种 Windows 版本之中。同样值得纪念的还有 Flash 插件的横空出世: 1990 年代初期, Jonathan Gay 在 FutureWave 公司开发了一种名为 Future Splash Animator 的二维矢量动画展示工具, 1996 年, Macromedia 公司收购了 FutureWave, 并将 Jonathan Gay 的发明改名为我们熟悉的 Flash。从此, Flash 动画成了 Web 开发者表现自我、展示个性的最佳方式。

除了编写 HTML 页面之外, 客户端应用的开发者还可以利用一些成熟的技术将浏览器的功能添加到自己的应用程序中。从 1992 年开始, W3C 就免费向开发者提供 libwww 开发库。借助 libwww, 我们可以自己编写 Web 浏览器和 Web 搜索工具, 也可以分析、编辑或显示 HTML 页面。1999 年, Microsoft 在 IE 5.0 中引入的 HTAs (HTML Applications) 技术则允许我们直接将 HTML 页面转换为一个真正的应用程序。从 1997 年的 IE 4.0 开始, Microsoft 为开发者提供了 WebBrowser 控件和其他相关的 COM 接口, 允许程序员在自己的程序中直接嵌入浏览器窗口, 或调用各种浏览器的功能, 如分析或编辑 HTML 页面等。Windows 98 及其后的 Windows 操作系统甚至还利用 WSH (Windows Script Host) 技术将原本只在浏览器中运行的 JavaScript、VBScript 变成了可以在 WIN32 环境下使用的通用脚本语言, 这大概也可算作我们对 Web 客户端开发技术的一种巧妙利用吧。

二、服务端技术的成熟与发展

与客户端技术从静态向动态的演进过程类似，Web 服务端的开发技术也是由静态向动态逐渐发展、完善起来的。

最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。一种名为 SSI (Server Side Includes) 的技术可以让 Web 服务器在返回 HTML 文件前，更新 HTML 文件的某些内容，但其功能非常有限。第一种真正使服务器能根据运行时的具体情况，动态生成 HTML 页面的技术是大名鼎鼎的 CGI (Common Gateway Interface) 技术。1993 年，CGI 1.0 的标准草案由 NCSA (National Center for Supercomputing Applications) 提出，1995 年，NCSA 开始制定 CGI 1.1 标准，1997 年，CGI 1.2 也被纳入了议事日程。CGI 技术允许服务端的应用程序根据客户端的请求，动态生成 HTML 页面，这使客户端和服务端的动态信息交换成为可能。随着 CGI 技术的普及，聊天室、论坛、电子商务、信息查询、全文检索等各式各样的 Web 应用蓬勃兴起，人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。

早期的 CGI 程序大多是编译后的可执行程序，其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程，人们开始探寻用脚本语言实现 CGI 应用的可行方式。在此方面，不能不提的是 Larry Wall 于 1987 年发明的 Perl 语言。Perl 结合了 C 语言的高效以及 sh、awk 等脚本语言的便捷，似乎天生就适用于 CGI 程序的编写。1995 年，第一个用 Perl 写成的 CGI 程序问世。很快，Perl 在 CGI 编程领域的风头就盖过了它的前辈 C 语言。随后，Python 等著名的脚本语言也陆续加入了 CGI 编程语言的行列。

1994 年，Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP (Personal Home Page Tools) 语言。与以往的 CGI 程序不同，PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面，Web 应用的开发者可以用一种更加简便、快捷的方式实现动态 Web 功能。1996 年，Microsoft 借鉴 PHP 的思想，在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 使用的脚本语言是我们熟悉的 VBScript 和 JavaScript。借助 Microsoft Visual Studio 等开发工具在市场上的成功，ASP 迅速成为 Windows 系统下 Web 服务端的主流开发技术。当然，以 Sun 公司为首的 Java 阵营也不会示弱。1997 年，Servlet 技术问世，1998 年，JSP 技术诞生。Servlet 和 JSP 的组合（还可以加上 JavaBean 技术）让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能，此外，Java 的运行时编译技术也大大提高了 Servlet 和 JSP 的执行效率——这也正是 Servlet 和 JSP 被后来的 J2EE 平台吸纳为核心技术的原因之一。

Web 服务端开发技术的完善使开发复杂的 Web 应用成为可能。在此起彼伏的电子商务大潮中,为了适应企业级应用开发的各种复杂需求,为了给最终用户提供更可靠、更完善的信息服 务,两个最重要的企业级开发平台——J2EE 和 .NET 在 2000 年前后分别诞生于 Java 和 Windows 阵营,它们随即就在企业级 Web 开发领域展开了你死我活的拼争。平台之争让整个 Web 世界在最近的几年里不得安宁,但从某种意义上说,也正是这种针锋相对的竞争关系促使了 Web 开发技术以前所未有的速度提高和跃进。

J2EE 是纯粹基于 Java 的解决方案。1998 年, Sun 发布了 EJB 1.0 标准。EJB 为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。至此, J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年, Sun 正式发布了 J2EE 的第一个版本。紧接着,遵循 J2EE 标准,为企业级应用提供支撑平台的各类应用服务软件争先恐后地涌现了出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里最为成功的商业软件平台。随着开源运动的兴起, JBoss 等开源世界里的应用服务新秀也吸引了许多用户的注意力。到 2003 年时, Sun 的 J2EE 版本已经升级到了 1.4 版,其中三个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。至此, J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。

和 J2EE 不同的是, Microsoft 的 .NET 平台是一个强调多语言间交互的通用运行环境。尽管 .NET 的设计者试图以 .NET 平台作为绝大多数 Windows 应用的首选运行环境,但 .NET 首先吸引的却是 Web 开发者的目光。2001 年, ECMA 通过了 Microsoft 提交的 C# 语言和 CLI 标准,这两个技术标准构成了 .NET 平台的基石,它们也于 2003 年成为了 ISO 的国际标准。2002 年, Microsoft 正式发布 .NET Framework 和 Visual Studio .NET 开发环境。早在 .NET 发布之前,就已经有许多 Windows 平台的 Web 开发者迫不及待地利用 Beta 版本开发 Web 应用了。这大概是因为, .NET 平台及相关的开发环境不但为 Web 服务端应用提供了一个支持多种语言的、通用的运行平台,而且还引入了 ASP.NET 这样一种全新的 Web 开发技术。ASP.NET 超越了 ASP 的局限,可以使用 VB.NET、C# 等编译型语言,支持 Web Form、.NET Server Control、ADO.NET 等高级特性。客观地讲, .NET 平台,尤其是 .NET 平台中的 ASP.NET 的确不失为 Web 开发技术在 Windows 平台上的一个集大成者。

如果说 HTML 语言给 Web 世界赋予了无限生机的话,那么, XML 语言的出现大概就可以算是 Web 的一次新生了。按照 Tim Berners-Lee 的说法, Web 是一个“信息空间”。HTML 语言具有较强的表现力,但也存在结构过于灵活、语法不规范的弱点。当信息都以 HTML 语言的面貌出现时, Web 这个信息空间是杂乱无章、没有秩

序的。为了让 Web 世界里的所有信息都有章可循、有法可依，我们需要一种更为规范、更能够体现信息特点的语言。

1996 年，W3C 在 SGML 语言的基础上，提出了 XML (Extensible Markup Language) 语言草案。1998 年，W3C 正式发布了 XML 1.0 标准。XML 语言对信息的格式和表达方法做了最大程度的规范，应用软件可以按照统一的方式处理所有 XML 信息。这样一来，信息在整个 Web 世界里的共享和交换就有了技术上的保障。HTML 语言关心的是信息的表现形式，而 XML 语言关心的是信息本身的格式和数据内容。从这个意义上说，XML 语言不但可以将客户端的信息展现技术提高到一个新的层次，而且可以显著提高服务端的信息获取、生成、发布和共享能力。为了将 XML 信息转换为 HTML 等不同的信息展现形式，1999 年，W3C 制定出了 XSLT 标准。同一年，IE 5.0 增加了对 XML 和 XSLT 的支持。

现在，网站的开发者可以直接使用 XML 语言发布信息了。针对不同的应用领域，人们还制定了许多专门的 XML 规范。例如，2001 年 W3C 发布的 SVG (Scalable Vector Graphics) 1.0 标准就是一种用 XML 语言表达的、全新的二维矢量图形格式。开发者可以用 SVG 格式描述大多数已有的 Flash 动画。与 Flash 格式相比，符合 XML 标准的 SVG 格式显然更有利于信息交换和共享。

Web 本身就是一个最大的分布式应用系统。对于分布式开发而言，XML 技术也大有用武之地。一个明显的事实是，如果能让分布式应用借助 XML 格式交换信息，那么，以往横亘在分布式架构上的信息交换难题也就迎刃而解了。1999 年，W3C 和相关的企业开始讨论设计基于 XML 的通信协议，2000 年，W3C 发布 SOAP (Simple Object Access Protocol) 协议的 1.1 版。人们把利用 SOAP 协议传递 XML 信息的分布式应用模型称为 Web Service。2001 年，W3C 发布了 WSDL (Web Services Description Language) 协议的 1.1 版。SOAP 协议和 WSDL 协议共同构成了 Web Service 的基础。随后，J2EE 和 .NET 这两大企业级开发平台先后实现了 Web Service，并将其视为平台的一项核心功能。

Web Service 对于 Web 开发者的重要意义在于，当我们需要在不同的服务端、不同的客户端乃至不同的应用类型、不同的计算设备之间传递信息的时候，以往的分布式开发技术或者因为适应性不强，或者因为扩展能力不足，都难以满足现代 Web 开发的需要，而 Web Service 正好填补了这一空白。

三、Web 开发框架和应用模型

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