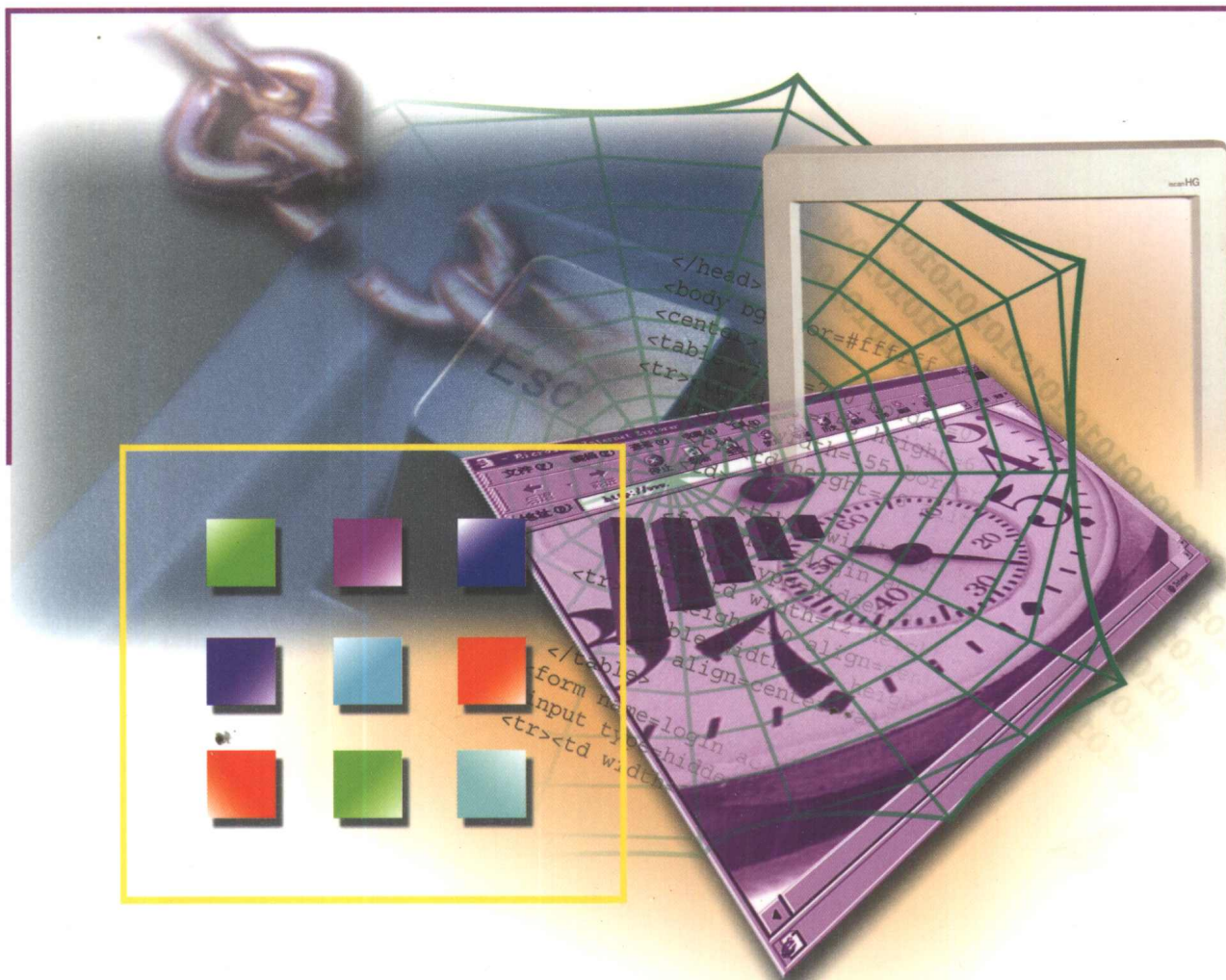




XML & ASP

网页程序设计

陈勤意 陈长念 编著

12XM



清华大学出版社
http://www.tup.tsinghua.edu.cn



网页设计梦之队丛书

XML&ASP 网页程序设计

陈勤意 陈长念 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

清华大学出版社

(京)新登字 158 号

内 容 简 介

XML 即可扩展标记语言, 是一种由 W3C 所推荐的标准, 也是目前网络中最热门的信息技术之一, 而 ASP 是微软为了推广网络应用程序开发和数据库连接而推出的一项交互式应用开发技术。

XML 的数据结构可以让用户根据不同的方式检索数据, ASP 可以让用户与网站交互, 充分发挥 XML 的功能。因此, XML 与 ASP 的结合可以产生惊人的交互效果。

本书适用于中高级网页程序设计人员使用。

本书繁体字版名为《XML&ASP 网页程序设计》由知城数位科技股份有限公司授权出版, 版权属陈勤意、陈长念所有。本书简体字中文版授权清华大学出版社出版, 其专有出版权属清华大学出版社所有, 未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或手段复制或传播本书的部分或全部内容。

北京市版权局著作合同登记号: 图字 01-2000-2257 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: XML&ASP 网页程序设计

作 者: 陈勤意 陈长念

责任编辑: 夏兆彦

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

[http:// www. tup. tsinghua. edu. cn](http://www.tup.tsinghua.edu.cn)

印 刷 者: 北京市丰华印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 22 字数: 532 千字

版 次: 2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

书 号: ISBN 7-900635-33-5

印 数: 0001 ~ 5000

定 价: 42.00 元(含 1 张光盘)

序

现在，XML 的发展已经是势不可挡。在市场的竞争中，快速采用最具潜力和竞争力的标准，将会成为企业成功的关键因素之一，而 XML 正是此时的明日之星。因此，如何将 XML 运用到互联网应用程序中，是每个网络程序员所必须面对的任务。

XML 1.0 标准已在 1998 年底制定，关于 XML 的应用目前正如火如荼地进行；除此之外还有许多与 XML 相关的外围标准正在发展之中，例如，XSL、XLL、XML Schema 等。XML 基本上是一种特别为互联网标准所制定的标记语言，运用它可以编写自己的标记语言，从而制作电子文件，并进一步实现电子数据交换的目的。

所以，XML 主要是用来制定新的标准并定义一种新的标记的语言；其自定的范围包括标记及属性名称，XML 文件可以选择性地包含语法描述数据，以便 XML 应用程序可以检查其完整性。用户可以通过 XML 定义专属的文件格式，程序员也能编写出便于处理文件的应用程序。也就是说，XML 要发挥作用，除了浏览器的支持之外，还需要客户端的程序配合；目前新版本的 Internet Explorer 以及 Netscape 浏览器都已支持 XML，因此主要的问题是客户端的应用程序该如何配合。

让我们来想想下面这些基本的问题：如何操纵 XML 数据？如何才能将 XML 数据与数据库链接？如何才能将 XML 数据与动态网页相结合？以及如何将 XML 数据按照指定的格式显示到客户端浏览器？本书试图说明这些疑问并提供答案。请读者朋友们耐心看下去，并请不吝指教。

陈勤意 陈长念
2000 年 12 月 4 日

目 录

第 1 章 为何要使用 XML	1
1.1 XML 概述	1
1.1.1 什么是 XML	1
1.1.2 开发历史	1
1.1.3 XML 与 SGML、HTML 的关系	2
1.1.4 XML 与 HTML、SGML 的差异	2
1.1.5 什么是 W3C	4
1.1.6 W3C 程序	4
1.1.7 XML 的 7 个概念	5
1.1.8 XML 概况	6
1.2 XML 的特点	8
1.2.1 制定 XML 的主要目的	8
1.2.2 XML 的优点	9
1.2.3 XML 的应用	9
1.3 企业使用 XML 的现状	10
1.4 本章小结	11
1.4.1 XML 主要应用范围	11
1.4.2 XML 的挑战与限制	11
1.4.3 XML 相关标准及其目的	12
1.4.4 小结	12
第 2 章 为何要使用 ASP	13
2.1 客户机/服务器结构介绍	13
2.1.1 客户机/服务器结构基本概念	13
2.1.2 客户机/服务器结构说明	14
2.1.3 2 层的客户机/服务器结构	15
2.1.4 3 层或多层的客户机/服务器结构	15
2.1.5 客户机/服务器结构的优点	16
2.1.6 客户机/服务器结构的缺点	17
2.2 ASP 概述	17
2.2.1 什么是 ASP	17
2.2.2 ASP 的发展	17
2.2.3 脚本语言可分为两大类	18

2.2.4	ASP 与 3 层客户机/服务器结构	19
2.2.5	ASP 的运行原理	19
2.2.6	ASP 的特点	20
2.2.7	ASP 的功能	20
2.2.8	ASP 的应用	21
第 3 章	为何要结合 XML 与 ASP	23
3.1	XML+ASP 网站应用结构	23
3.1.1	XML+ASP 的 3 层结构	24
3.1.2	数据传输的方式	24
3.2	XML 与 ASP 取长补短	24
3.3	网络应用程序遇到的问题	26
3.3.1	范例网站	27
3.3.2	遇到的问题	27
3.4	使用 XML+ASP 解决问题	28
3.5	本章小结	30
第 4 章	XML 文件结构	31
4.1	XML 文件概况	31
4.2	逐步地创建一个 XML 实例	32
4.2.1	步骤 1: 创建蓝图	32
4.2.2	步骤 2: 按图施工	38
4.2.3	步骤 3: 外观装饰	40
4.3	名称空间与模式	41
4.3.1	名称空间	41
4.3.2	XML 模式	43
4.4	本章小结	48
第 5 章	文件对象模型(DOM)介绍	49
5.1	DOM 对象介绍	50
5.1.1	XMLDOMDocument 对象	51
5.1.2	XMLDOMElement 对象	52
5.1.3	XMLDOMNode 对象	52
5.1.4	XMLDOMNodeList 对象	53
5.1.5	XMLDOMNamedNodeMap 对象	54
5.1.6	XMLDOMText 对象	54
5.1.7	XMLDOMAttribute 对象	55
5.1.8	XMLDOMDocumentType 对象	55
5.1.9	XMLDOMProcessingInstruction 对象	56
5.1.10	XMLDOMParseError 对象	56
5.1.11	XMLHttpRequest 对象	57

5.2 DOM 对象属性	57
5.2.1 async	58
5.2.2 documentElement	58
5.2.3 attributes	59
5.2.4 childNodes	60
5.2.5 length	62
5.2.6 text	62
5.2.7 nodeType	64
5.2.8 data	65
5.2.9 nodeValue	66
5.2.10 nodeTypeString	67
5.2.11 nodeName	68
5.2.12 docType	70
5.2.13 firstChild、lastChild、nextSibling 及 previousSibling	70
5.2.14 parentNode	71
5.2.15 tagname	72
5.2.16 value	73
5.2.17 ondataavailable	74
5.3 DOM 对象方法	74
5.3.1 load()	74
5.3.2 loadXML()	75
5.3.3 abort()	76
5.3.4 cloneNode()	76
5.3.5 createNode()	77
5.3.6 appendChild()	78
5.3.7 insertBefore()	79
5.3.8 replaceChild()	79
5.3.9 removeChild()	80
5.3.10 hasChildNodes()	81
5.3.11 getElementsByTagName()	81
5.3.12 nextnode()	82
5.3.13 getAttribute()	83
5.3.14 getAttributeNode()	84
5.3.15 getNamedItem()	85
5.3.16 removeAttribute()	86
5.3.17 removeAttributeNode()	86
5.3.18 removeNamedItem()	87
5.3.19 setAttribute()	88
5.3.20 setAttributeNode()	89

5.3.21	setNamedItem()	90
5.3.22	selectNodes()	91
5.3.23	selectSingleNode()	91
5.3.24	reset()	92
5.3.25	save()	93
5.3.26	open()	94
5.3.27	send()	95
5.4	本章小结	95
第 6 章	开始使用 ASP	97
6.1	ASP 基础	97
6.1.1	ASP 基本概念	97
6.1.2	ASP 基本结构	98
6.1.3	一个完整的 ASP 程序	99
6.1.4	ASP 主要对象	100
6.1.5	ASP 主要组件	100
6.2	第 1 个 ASP 程序	101
6.2.1	基本 ASP 程序	101
6.2.2	以 ASP 处理 XML 数据	107
6.2.3	以 XML 定义 ASP 中的超链接及个性化的设置	111
6.2.4	以 ASP 控制 XML 文件	115
第 7 章	数据岛、XML 与 ASP	121
7.1	RDS、DSO 与数据岛介绍	121
7.1.1	数据绑定	122
7.1.2	DSO	123
7.1.3	RDS	125
7.1.4	数据岛	126
7.2	使用数据岛	128
7.2.1	如何访问 XML 数据岛	128
7.2.2	XML 数据岛会用到的 HTML 元素	129
7.3	数据岛的各种使用案例	129
7.4	本章小结	154
第 8 章	XSL、XML 与 ASP	155
8.1	XSL 介绍	155
8.1.1	什么是 XSL	155
8.1.2	XSL 的功能	155
8.1.3	XSL 的运行过程	156
8.1.4	XSL 的重要性	157
8.2	使用 XSL	158

8.2.1 一个简单的 XSL.....	158
8.2.2 XSL 与 XML 的链接.....	160
8.3 创建 XSL 样式表.....	163
8.3.1 XSL 样式表结构.....	163
8.3.2 模板和样式.....	164
8.3.3 XSL 元素介绍.....	170
8.4 XSL 的各种使用案例.....	176
8.4.1 在 XML 文件中使用 XSL.....	176
8.4.2 在数据岛中使用 XSL(1).....	179
8.4.3 以 XSL 排序.....	182
8.4.4 在数据岛中使用 XSL(2).....	184
8.4.5 利用 XSL 创建超链接.....	187
8.4.6 ASP 与 XSL 配合(1)——没有产生另一个 html 文件.....	188
8.4.7 ASP 与 XSL 配合(2)——产生另一个 html 文件.....	189
8.5 本章小结.....	192
第 9 章 ADO、XML 与 ASP.....	193
9.1 ADO 介绍.....	193
9.1.1 微软 UDA 结构与组件.....	193
9.1.2 OLE DB 与 ADO.....	195
9.1.3 ADO 2.6 对象.....	197
9.1.4 Recordset 对象的指针与数据锁定.....	199
9.2 ADO 与 ASP.....	201
9.2.1 链接到数据源.....	201
9.2.2 通过网页与数据库链接.....	205
9.2.3 逐步练习: ASP 与 ADO.....	206
9.3 ADO 与 XML 整合.....	208
9.3.1 ADO 与 XML 不冲突.....	208
9.3.2 ADO Recordset 与 XML.....	209
9.4 ADO 的各种范例.....	211
9.4.1 将 Recordset 转换成 XML、将 ADO Recordset 存为 XML 文件.....	211
9.4.2 以 Recordset 创建 DSO.....	216
9.4.3 应用 XSL.....	225
9.5 本章小结.....	227
9.5.1 FOR XML RAW.....	228
9.5.2 FOR XML AUTO.....	228
附录 A XML1.0 规范.....	231
附录 B DOM 对象模型.....	271
附录 C 使用 IIS.....	301

附录 D ASP 对象	321
附录 E ADO 2.6 对象模块	329

第 1 章 为何要使用 XML

XML(eXtensible Markup Language, 可扩展标记语言)于 1998 年正式成为 W3C(World Wide Web Consortium)的标准, 因其为网络环境量身定制的特性、其逻辑结构的严谨性及其验证功能, 俨然已成为信息交换格式的主流技术。

甚至连联合国下属的研究与推广电子数据交换(EDI)标准机构 EWG(Electronic Data Interchange for Administration, Commerce and Transport Working Group), 也在会议中声明未来电子交换标准将不只使用 EDI, 也将采用 XML; 其中, XML 将是未来电子交换标准的新方向。因此许多组织、电子商务解决方案的提供者提供程序、以及具前瞻性的厂商都已开始建立 XML 相关的应用和技术。

XML 的特色是使文件间的交换、信息的传递更为便捷, 谈到 XML 则必须同时谈到 SGML(Standard Generalized Markup Language, 标准通用标记语言)及 HTML(HyperText Markup Language, 超文本标记语言)与 XML 和两者间的密切关系。

1.1 XML 概述

1.1.1 什么是 XML

XML 就是 eXtensible Markup Language, 可扩展标记语言。这是一种由 W3C 所推荐的标准, 也是目前互联网中最炙手可热的信息技术(Information Technology, IT)。它属于一种元语言(Meta Language), 可用来定义任何一种新的标记语言, 创造新文件的格式定义。

过去一年内, 台湾地区之外各大科技公司无不竞相开发各种 XML 相关技术, 就连红极一时的电子商务也必须与 XML 应用相结合。因此, 身为一个互联网应用程序员, 当然不能不知道 XML。

1.1.2 开发历史

XML 的前身是 SGML, 这是一种用来描述电子文件结构与内容的标记语言。SGML 于 1986 年推出, 是一种国际标准, 功能强大但是复杂繁琐。由于 SGML 有着复杂性高及费用昂贵的缺点, 因此便有了将 SGML 简化的想法, 而 SGML 简化之后的结果便是 XML。第一个 XML 建议标准于 1998 年 2 月提出, 也就是目前通用的 XML 1.0 版本, 而 XML

规范便是由 W3C XML 工作组维护并负责未来的调整和修改的。

1.1.3 XML 与 SGML、HTML 的关系

XML 与 SGML、HTML 的关系，如下图所示：

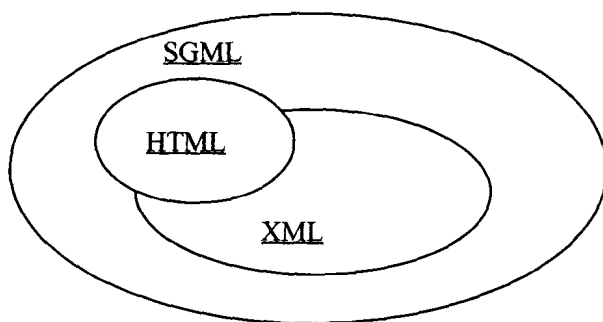


图 1.1 XML 与 SGML、HTML 的关系示意图

基本上 SGML、HTML 及 XML 不会因为其中任何一个的存在，而使其他语言消失。虽然 XML 出现了，但其主要用于长期使用且需结构化的文件；HTML 仍将继续应用于快速发布的简单网页。至于 SGML，虽然无法被 Internet 广泛接受，但是还将继续适用于复杂结构的应用情况。

1.1.4 XML 与 HTML、SGML 的差异

XML 和 HTML 是由 SGML 衍生而来的。SGML 是全世界标准的标记语言，用来定义结构化的电子文件格式，但因 SGML 过于详细和复杂，因此未被广泛采用。

HTML 是 Internet 及万维网(WWW)兴起后，被大量使用于网页制作的简单语言格式。

XML 与 SGML 相比更为简单及灵活。XML 隐藏了 SGML 较复杂的语法结构，因此开发处理 XML 格式文件的应用软件较容易，软件系统间也可互通。XML 拥有 Web 的功能，很适合网络传输。XML 为继承了 SGML 的优点且补充 HTML 的不足。

表 1.1 为 XML、SGML 和 HTML3 的比较。

表 1.1 XML、SGML 及 HTML 的比较

	SGML	HTML	XML
全名	Standard Generalized Markup Language	Hyper Text Markup Language	EXtensible Markup Language
中译名	标准通用标记语言	超文本标记语言	可扩展标记语言
历史沿革	国际标准组织(ISO)于1986年发布的电子文件交换标准	源自 SGML, 1989 年由欧洲原子物理研究中心(CERN)的研究人员 Tim Berners Lee 所创, 是 SGML 的应用	源自 SGML, 1998 年 2 月由 W3C 所开发
用途	规范文件制作及交换标准, 用于不同计算机系统的信息交换	针对网页页面表现方式, 主要是编写简单网页的语言	针对网页内容, 编写结构化文件的语言
文件内容	1.SGML 声明 2.DTD(Document Type Definition) 3.DI(Document Instance)	1.文字 2.圆形 3.超链接	1.XML 2.XML 模式 3.XSL 4.XLL
特色	DTD(Document Type Definition)	互联网使用的超文本文件	内容与呈现格式是分离的, 而不是锁定的。
优点	1.可应各种信息自行定义扩充标准 2.维持数据格式一致 3.重复使用, 资源共享 4.严谨正确的结构化文件 5.不受计算机系统、特殊外围设备及语言的限制 6.可随所需格式输出	1.简单易学 2.支持各种方法制作 HTML 3.简易标准的连结 4.创建表单容易 5.程序编写容易 6.提供开放的平台	1.继承 SGML 的优点 2.改善 HTML 的缺点 3.容易开发应用程序 4.具有可扩展性, 可广泛应用到各专业领域 5.多种内嵌物之功能 6.异质数据库间文件处理 7.采用 Unicode
缺点	1.缺乏主流的浏览器支持, 使向下转换的步骤删除原始 SGML 文件中的智能型机制, 降低信息弹性造成数据重复使用、交换等缺点 2.缺乏标准样式的支持	1.结构限制造成浏览障碍 2.缺乏对表达内容的支持 3.格式固定限制数据交换 4.缺乏一致性, 限制自动化 5.搜索不够精确 6.新功能和新标签不断扩充, 不适于大量管理及维护网页	1.各界使用的标签不一, 难以执行 2.各领域使用自行定义标签, 易产生兼容性之问题 3.数据安全问题有疑问

1.1.5 什么是 W3C

W3C 是在 1994 年 10 月成立的一个国际组织。它成立的宗旨在于想要创建全球信息网的共通标准和规范, 并提供技术标准, 以使用于网络应用程序的互连并主导 WWW 的技术革新。

所以在 W3C 的网站, 可以找到如下的服务:

1. 提供信息库给研发者和客户机参考, 特别是信息网的规格数据。
2. 提供参考程序(代码)以利于植入应用软件与推广标准。
3. 提供各种标准和示范应用程序以展示新技术的应用。

W3C 研发中的信息可供所有会员参考, 而在研究成果对内部公开一个月后, 所有的软件出版物将公开使用。W3C 也有论坛供所有会员讨论全球信息网的相关议题。

1.1.6 W3C 程序

目前 W3C 的制定程序及相关文件, 有下列几种:

- **W3C Recent Recommendation(推荐使用)**
此文件和研究已获得 W3C 一致认同和 W3C 执行负责人的认可。W3C 认定此文件可公开使用并能提高 W3C 影响, 是一份稳定的文件(表示不会经常修改), 可以当作参考数据, 也可作为其他文件的正式参考文献。
例如, XML 1.0 是 Recommendation 版本。
- **W3C Proposed Recommendation(准备推荐使用)**
文件和研究已获得参与制作团体的认同, 且通过 W3C 执行长推荐给审核小组以供检阅。
例如, XHTML 是 Proposed Recommendation 版本。
- **W3C Candidate Recommendations(候选推荐标准)**
文件和研究只获得相关技术产业的认同, 期望可以上呈给 W3C 通过认同, 因而推荐给审核小组进行审查。
例如, Extensible Stylesheet Language(XSL) Version 1.0 便是 Candidate Recommendations 版本。
- **W3C Recent Working Draft(工作草案)**
进行中的研究报告, 不代表已被参与团体或 W3C 认同, 但代表着 W3C 的研究方向。
例如, Document Object Model(DOM) Level3 Views and Formatting Specification 便是 Working Draft 版本。
- **W3C Recent 注意**
有记载日期的公开文件, 含有概论、评价及技术报告。

1.1.7 XML 的 7 个概念

1. XML 是一个把“结构化的数据”放到文件中的方法

所谓结构化的数据，就是指表格、电话簿、金融事务记录、工程图文件之类的数据。一般说来，用来产生这类数据的程序多半会将数据以文本形式或是二进制形式存储在磁盘中，若是这些数据是以文本的形式存储，那么数据的读取自然就相当便利，而不需要再借助到原本用来产生该数据的程序。

XML 便是一系列的守则、规则与常规，凡是通过使用 XML 来设计文字编排(也就是设计数据的结构)，便能容易产生且易于读取的文件，而这些文件当然还会清晰且具有扩充性。

2. XML 有点像 HTML，但不是 HTML

如同 HTML 一样，XML 也有标记和属性。差别在于 HTML 已经指定了每个标记及属性的意义，甚至连介于标记和属性间的文本在浏览器上如何显示都确定下来；但是 XML 只使用标记来分数据的区块并且将如何阐释数据全部交由应用程序(例如，浏览器)来决定。换句话说，如果你在 XML 文件中看到“<p>”，并不代表接着就会是一段文字；而要视根据个 XML 文件的情况而定，有可能是表示价钱、参数、人、……。

因为 XML 与 HTML 之间这些相同与不同的地方，所以说“XML 有点像 HTML 但不是 HTML”的说法。

3. XML 是文字，但并不是用来读的

XML 文件是文本文件(这在第一点中已经说到)，但是和 HTML 大不相同的地方在于，XML 文件并不是设计来被人类读取。XML 与 HTML 是文本文件，因此它们也允许程序员能够更容易地在应用程序中除错，而且在比较迫切的情况下，可以使用简单的文本编辑器去修改 XML 文件中的漏洞。但是 XML 文件的规则比 HTML 更为严格。要是 XML 文件中，属性之前没有加上引号或是少了一个标记，那么整个 XML 文件便不能被使用；但是在 HTML 中这些情况通常是默许的或可容忍的。常见的 XML 规格书中提到：应用程序不被允许猜测 XML 文件有缺失的地方，如果文件有遗漏，就必须马上停止执行，并且发布错误信息。

4. XML 代表的是系列的信息技术

目前 XML 1.0 规格书中，主要定义了什么是 XML 中的标记及属性。但是源自于 XML 1.0，又有许多正在开发的可扩展技术，例如，为了某些特定目的而制定的一系列标记及属性，或是可扩展出来的相关规格书。

以目前的相关技术而言，比较著名的是：Xlink、XPointer、XFragments、CSS、XSL、XML、DOM、Namespace(名称空间)、RDF、模式等，这些名词将在本章陆续为读者介绍。目前读者仅需知道已经有许多由 XML 扩展出来的技术正在开发，所以才说 XML 代表的是一系列的信息技术。

除了这些之外还有更多的相关技术，或是一系列为了特定用途而制订的标记属性

模块正在开发中。就让我们拭目以待,看看 W3C 还会推出什么最新的技术出来。

5. XML 的确是有点啰嗦,但那不是问题

既然 XML 是文本格式,且通过标记的使用来界定数据的范围,因此 XML 文件通常比相同的 2 进制文件要大得多,这是 XML 开发小组经过审慎思量后所决定的。优点是文本格式的文件比较明确(参照第 3 点),缺点当然就是所占的存储空间比较大。但是随着硬件的开发,硬盘越来越便宜;而各种平台都有支持的压缩程序(例如,zip),能快速且有效地压缩文件大小,且往往是免费的。此外,像通讯协议(如数据机的协议或是 HTTP/1.1)也可以直接压缩文件,因此 XML 格式就和 2 进制格式一样不会占带宽。

所以虽然 XML 文件内容繁琐且文件较大,但是并不会有什么大问题。

6. XML 是一个新东西,但也不全新

XML 开发小组是从 1996 年开始,并且在 1998 年 2 月成为 W3C 所推出的标准之一。距离现在不是多久,因此你可能会怀疑 XML 是一个不成熟的技术。但是事实上,XML 并不是一个新技术。之前在介绍 XML 开发历史时曾说到 XML 的前身是 SGML,而 SGML 在 80 年代就开始开发了,在 1986 还成为 ISO 标准并曾被广泛应用在大型的文件管理中;而 HTML 则是在 1990 年开始开发。XML 开发小组便是将 SGML 去芜存菁后,并配合 HTML 的开发经验来制定 XML。因此 XML 虽然是一个新的技术,但也不是完全没过去的开发历史。

7. XML 是一个不用授权、跨平台且被完全支持的语言

因为 XML 已经是 W3C 开放的标准,所以不需经过授权人人可以直接使用,因此客户机可以自行开发自己的软件,而不需要付 XML 的使用费。

此外,因为 XML 并不是某一家公司的特定出品,而是项标准技术,所以在任何平台上都能使用与执行。且由于现在已经有大量的 XML 相关支持及开发工具,所以 XML 可说是备受肯定及充分支持。

对于跨平台信息交换而言,XML 本身为文本,无论任何平台都可应用,此外 XML 将文件内容与显示方式分离,有助于信息于不同平台的再利用;如 WML 即为移动电话上的应用。

因为 XML 具有不需授权、跨平台及充分被支持 3 个特点,选择 XML 当作网站或是专项开发的基础虽然并不一定是最好的方案,但绝对是值得考虑的方法。

1.1.8 XML 概况

- DTD

文件大都具有相当程度的结构,只不过没有经过特别的归纳,但是 XML 允许客户自定义 DTD(Document Type Definition, 文档类型定义),在 DTD 中可以制定文件的结构性,可以根据此 DTD 中所制定的结构来创建文件。

所以,可以说 DTD 便是用来制定 XML 文件类型上的定义,可以加强对 XML 文件在结构上的要求。一份 XML 文件可以参照 0 或 1 个以上的 DTD,并不是每

份 XML 文件都要有 DTD，客户机可以自由选择要不要有 DTD。

- 合式

当一份 XML 文件满足 XML 规范中的相关法则，且该份文件中不使用任何 DTD 时，就可以称这份 XML 文件是合式的。

- 有效

当一份 XML 文件满足下列所有条件，就可以称该份文件是有效的。

1. 该份 XML 文件是合式的。
2. 该份 XML 文件使用了 DTD。
3. DTD 中的语法是正确的。
4. 在该 XML 文件中正确地使用了 DTD。

- 名称空间

在一份 XML 文件中可能同时参照多个 DTD，所以标记的名称就有可能重复。

例如，<NAME>这个标记在第一份 DTD 中指的是员工的名称，但在第 2 份 DTD 中也以<NAME>作为公司名称的标记。所以若是一份 XML 文件同时用到这两份 DTD 便会分不清<NAME>是代表员工名称还是公司名称。制定名称空间标准的主要目的是为解决这种问题，通过名称空间可以指定某个特定的标记是来自哪份 DTD。

XML 的名称空间并没有收录在 XML 1.0 的标准中，而是通过后来一份单项的标准来作增补，这份后来才定的标准叫做“XML 中的名称空间”。

- 模式

模式用来定义文件结构。读者看到这里一定会很疑惑：既然已有 DTD 在定义文件结构，为何 W3C 还要制订另一套定义文件结构的标准呢？原因在于 DTD 其实是 SGML 时代的遗留物，本身仍有许多地方有待加强，不能满足实际使用的需求。

DTD 有两个主要的缺点：

- ◆ DTD 未使用 XML 语法
- ◆ DTD 描述数据类型的能力薄弱：举例来说，DTD 无法描述时间、固定整数、文本等等。

基于上述问题，W3C 的模式工作小组参考行业模式相关的提案，如文档内容描述(Document Content Description, DCD)、资源描述框架(Resource Description Framework, RDF)、面向对象的 XML 模式 (Schema for Object-oriented XML, SOX) 以及 XML Data，制定模式标准，提出相应解决方案。目前该标准草案已进入最后一次建议阶段，如顺利通过，将可进入候选推荐标准 (Candidate Recommendation)，离正式标准的颁布也不远了。

- CSS

CSS 代表 Cascading Style Sheet，即级联样式表。它最主要的特色是排版的效果，以及层级的应用；主要是利用特定的标记来控制网页里的字体和排版。在以往是用来支持 HTML，但是现在又有 CSS2(Cascading Style Sheets Level 2，第 2 级级联样式表)，用来进一步支持 XML 数据的显示。