

e 时代网络学科推荐教程
E-time Network Discipline Recommend Textbooks



精通 XML 与网页设计高级教程

何东隆 李美真 / 编著

XML 语法规则的剖析
XML、HTML、CSS 的整合应用
网络数据库的原理与方法
个性化的 Script 程序
构建 BEHAVIOR 的应用范例
附赠光盘内含书中范例



中国青年出版社



文魁资讯股份有限公司

e 时代网络学科推荐教程
E-time Network Discipline Recommend Textbooks



精通 XML 与网页设计高级教程

何东隆 李美真 / 编著



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书由文魁资讯股份有限公司授权中国青年出版社独家出版。未经出版者书面许可,任何单位和个人均不得以任何形式复制或传播本书的部分或全部。

版权贸易合同登记号: 01-2000-4244

策 划: 胡守文

王修文

郭 光

责任编辑: 江 颖

何 琼

责任校对: 肖新民

书 名:《精通 XML 与网页设计高级教程》

编 著: 何东隆 李美真

出版发行: 中国青年出版社

地址: 北京市东四十二条 21 号 邮政编码: 100708

电话: (010) 64069368 传真: (010) 64053266

印 刷: 北京新丰印刷厂

开 本: 16 开

版 次: 2001 年 1 月北京第 1 版

印 次: 2001 年 1 月第 1 次印刷

印 数: 1-5000

标准书号: ISBN 7-5006-4173-7/TP · 108

定 价: 48.00 元 (附赠 1CD)

序

XML与**HTML**师出同门，都是从**SGML**（**Standard Generalized Markup Language**）延伸而出的标记语言。**HTML**着重数据的显示，制定的主要目标就是网页设计，自从1993年推出后，随着互联网的蓬勃发展，俨然已成为**WEB**网页的唯一母语。

HTML由于出道较早，而且一开始就被局限在显示数据的范畴中，虽然经过数次的修改与扩充，并且得到**CSS**的强力支持，终究也只能提高网页的视觉效果，对日趋迫切的网络数据处理能力渐渐显得捉襟见肘。

XML的制定，则以数据的建立及管理为主要目标。在**XML**文件中，可以自行定义数据的类型，可以建立结构化的数据库，还可以通过筛选条件进行数据排序与查询。此外，**XML**文件是一种通用性的数据库，可以用不同的程序语言来解读，当然也可以用**HTML**来处理。所以在电子商务的网络时代，用**XML**来建立数据，再由**HTML**来显示，将是网页设计的新方向。

XML还具有延伸性，用户可以依据不同的需求，自行设定标签的格式。因此，自从1998年通过**W3C**的审查后，迅速地改变了网页设计的观念与方法，**IE5.0**所支持的**HTC**、**Behavior**、**HTA**等等，都是典型的例子。

Scriptlet、**HTC**、**Behavior**都是元件化的**Script**程序，可以在不同的网页里重复使用，不但大幅提高开发的效率，还可以让不懂**Script**程序者享受更多设计网页的乐趣。**HTA**则是软件开发的新概念，直接将**HTML**文件存储成**HTA**的格式，就是一个独立的应用软件，与**VB**、**C++**等程序语言所设计的软件没什么差别。换言之，会设计网页就可以开发应用软件，只要通过浏览器就可以执行应用软件，操作系统似乎不重要了！**IE5.0**，一套形同免费的软件，微软却不惜为它大打官司，你认为是为了什么？

这本书的主题，除了**XML**的设计方法之外，主要着重**XML**与**HTML**的整合应用。对于**Scriptlet**、**HTC**、**HTA**等也有详细的范例及解说。希望能提供给读者一个辽阔的“网页设计的新视野”。

何东隆 李美真

目 录

第1章 XML概述

1.1	认识XML	2
1.1.1	何谓XML	2
1.1.2	XML的结构	2
1.1.3	IE5.0的支持	3
1.1.4	在XML中使用中文	5
1.2	XML文件的设计	6
1.2.1	文件的结构	6
1.2.2	声明区	7
1.2.3	定义区	7
1.2.4	文件的主体	8
1.2.5	嵌套的数据结构	10
1.2.6	注解	12
1.3	XML的语法规则	15
1.3.1	空标签	15
1.3.2	属性的设定	16
1.3.3	英文大小写	17
1.3.4	特殊符号	18
1.3.5	<CDATA>标签的应用	19
1.3.6	根元素的文字数据	20
1.4	XML与CSS样式表	21
1.4.1	样式表的套用方法	21
1.4.2	显示的内容	23
1.4.3	空标签的应用	23
1.4.4	样式的继承性	24
1.4.5	CSS的限制	26
1.5	在XML文件中使用HTML标签	29
1.5.1	HTML标签的套用方法	29
1.5.2	空标签的使用方法	30
1.5.3	标签的样式	32
1.5.4	CSS样式文件的设定	34
1.6	HTML与XML的组合	35
1.6.1	HTML的自定义标签	35
1.6.2	自定义标签的样式	38
1.6.3	Style的声明	40
1.6.4	自定义标签的对象模型	42

1.6.5	Data Island的设定方法	45
1.6.6	Data Island的应用	47
第2章	数据结构的定义(DTD)	
2.1	DTD的声明及引用	50
2.1.1	DTD的声明方法	50
2.1.2	Validating文件	53
2.1.3	外部DTD	54
2.1.4	内部DTD与外部DTD的整合	56
2.1.5	改变标签的使用方法	58
2.2	标签的类型	59
2.2.1	#PCDATA	59
2.2.2	EMPTY	60
2.2.3	ANY	60
2.2.4	嵌套结构	61
2.2.5	选择性的嵌套结构	63
2.2.6	混合类型	64
2.3	实体的声明与使用	66
2.3.1	内部实体	66
2.3.2	外部实体	68
2.3.3	在声明中使用实体	70
2.3.4	参数型的实体	72
2.3.5	实体的优先顺序	75
2.4	标签的属性	77
2.4.1	何谓属性	77
2.4.2	属性的设定方法	78
2.4.3	属性的内定值	81
2.4.4	选择性的内定值	85
2.4.5	XML:Space	87
2.5	属性的类型	88
2.5.1	CDATA	88
2.5.2	NMTOKEN及NMTOKENS	88
2.5.3	ID、IDREF、IDREFS	90
2.5.4	ENTITY、ENTITIES	92
第3章	数据的来源(XML DSO)	
3.1	XML的对象模型	96
3.1.1	对象及属性	96

3.1.2	根元素的名称及性质	99
3.1.3	对象的阶层架构	103
3.1.4	标签的数目	105
3.1.5	显示文件内容	106
3.2	XML的数据类型	108
3.2.1	数据类型的设定方法	108
3.2.2	常用的类型	111
3.2.3	NodeTypedValue	113
3.2.4	数据类型的检查	115
3.2.5	DataType属性	118
3.3	显示数据	120
3.3.1	DSO的设定方法	120
3.3.2	移动数据库的指标	125
3.3.3	指标的位置	127
3.3.4	数据库的长度	131
3.4	表格的应用	135
3.4.1	显示数据库	135
3.4.2	栏位名称的设定	139
3.4.3	显示一条数据	141
3.4.4	嵌套表格的应用	144
第4章	XSL的设计	
4.1	认识XSL	150
4.1.1	XSL的声明与套用	150
4.1.2	套用XSL样式	151
4.1.3	XSL与HTML	152
4.1.4	完整的HTML文件	155
4.1.5	CSS样式的套用	158
4.2	数据的安排	161
4.2.1	XSL:value.of	161
4.2.2	数据显示的顺序	163
4.2.3	属性的应用	165
4.2.4	xsl:for.each	167
4.2.5	路径的管理	170
4.2.6	数据排序	172
4.2.7	标签的属性	175
4.3	样版的设计	178

4.3.1	XSL:Apply.Templates	178
4.3.2	指定显示的标签	182
4.3.3	嵌套数据的处理	185
4.3.4	多重数据的处理	189
4.3.5	多重数据的样版	193
4.4	XSL的其他元素	195
4.4.1	XSL: COPY	195
4.4.2	XSL: COMMENT	199
4.4.3	XSL: Node.Name	201
4.4.4	XSL: Element	203
4.4.5	XSL: Attribute	204
4.4.6	XSL: Eval	207
第5章	XSL与数据处理	
5.1	标签及属性的指定方法	210
5.1.1	阶层与路径	210
5.1.2	“*” 的用法	212
5.1.3	“//” 的用法	215
5.1.4	属性的指定方法	216
5.1.5	标签的集合	218
5.2	数据的筛选条件	221
5.2.1	检查标签名称	221
5.2.2	检查标签的内容	225
5.2.3	检查属性值	228
5.2.4	布林运算	230
5.2.5	比较运算	234
5.2.6	组合条件	236
5.2.7	集合与运算	238
5.3	筛选条件的应用	242
5.3.1	在样版中使用筛选条件	242
5.3.2	XSL: If	245
5.3.3	XSL: CHOOSE	250
5.4	XSL的内建函数	255
5.4.1	FormatIndex	255
5.4.2	ChildNumber	257
5.4.3	AbsoluteChildNumber	261
5.4.4	Depth	264

5.4.5	UniqueID	267
第6章 ScriptLet		
6.1	认识ScriptLet	270
6.1.1	何谓ScriptLet	270
6.1.2	ScriptLet的套用方法	272
6.1.3	嵌套的ScriptLet	275
6.1.4	ScriptLet的外观	278
6.1.5	ScriptLet的名称	281
6.2	ScriptLet的设计原理	284
6.2.1	内部变量	284
6.2.2	外部变量	287
6.2.3	子程序的设计	291
6.2.4	SetTimeout () 的用法	295
6.2.5	改变ScriptLet的元件	299
6.3	事件的处理与传递	304
6.3.1	事件的传递	304
6.3.2	事件的检测	307
6.3.3	EVENT的应用	313
6.3.4	自定义的事件	317
6.3.5	ScriptLet的地址	323
6.3.6	ScriptLet的载入状态	326
6.4	ScriptLet的其他属性	328
6.4.1	ScrollBar	328
6.4.2	滚动条的形式	333
6.4.3	Frozen	336
6.4.4	SelectableContent	339
6.4.5	Version	344
第7章 Behavior与HTC		
7.1	认识Behaviore	348
7.1.1	何谓Behavior	348
7.1.2	HTC的文件结构	350
7.1.3	元件的样式	355
7.1.4	事件的处理顺序	359
7.1.5	HTC的应用	364
7.2	Behavior的设定方法	369
7.2.1	整页套用的设定方法	369

7.2.2	个别套用的设定方法	372
7.2.3	自定义样式	375
7.2.4	局部设定	376
7.2.5	在程序中套用Behavior	380
7.2.6	自定义标签的Behavior	381
7.3	HTC的进阶设计	383
7.3.1	ATTACH的应用	383
7.3.2	自定义的属性	386
7.3.3	属性的应用	390
7.3.4	自定义的METHOD	393
7.4	内建Behavior的应用	394
7.4.1	ClientCaps	394
7.4.2	DownLoad	397
7.4.3	HomePage	399
7.4.4	SaveFavorite	402
7.4.5	SaveHistory	404
第8章 HTML应用程序		
8.1	认识HTA	410
8.1.1	何谓HTA	410
8.1.2	HTA的操作	413
8.1.3	HTA的设计	414
8.1.4	HTA的标题	415
8.1.5	HTA的应用	416
8.2	HTA的外观	418
8.2.1	〈HTA: Application〉	418
8.2.2	标题栏的设定方法	421
8.2.3	HTA的图标	425
8.2.4	HTA的边框	428
8.2.5	边框的形式	432
8.2.6	视窗的大小	433
8.3	HTA的其他属性	435
8.3.1	CommandLine	435
8.3.2	Version	436
8.3.3	SingleInstance	437
8.3.4	ShowInTaskBar	438



XML 概述

- 1.1 认识XML
 - 1.1.1 何谓XML
 - 1.1.2 XML的结构
 - 1.1.3 IE5.0的支持
 - 1.1.4 在XML中使用中文
- 1.2 XML文件的设计
 - 1.2.1 文件的结构
 - 1.2.2 声明区
 - 1.2.3 定义区
 - 1.2.4 文件的主体
 - 1.2.5 嵌套的数据结构
 - 1.2.6 注解
- 1.3 XML的语法规则
 - 1.3.1 空标签
 - 1.3.2 属性的设定
 - 1.3.3 英文大小写
 - 1.3.4 特殊符号
 - 1.3.5 <CDATA>标签的应用
 - 1.3.6 根元素的文字数据
- 1.4 XML与CSS样式表
 - 1.4.1 样式表的套用方法
 - 1.4.2 显示的内容
 - 1.4.3 空标签的应用
 - 1.4.4 样式的继承性
 - 1.4.5 CSS的限制
- 1.5 在XML文件中使用HTML标签
 - 1.5.1 HTML标签的套用方法
 - 1.5.2 空标签的使用方法
 - 1.5.3 标签的样式
 - 1.5.4 CSS样式文件的设定
- 1.6 HTML与XML的组合
 - 1.6.1 HTML的自定义标签
 - 1.6.2 自定义标签的样式
 - 1.6.3 Style的声明
 - 1.6.4 自定义标签的对象模型
 - 1.6.5 Data Island的设定方法
 - 1.6.6 Data Island的应用

1.1 认识XML

1.1.1 何谓XML

XML的全名是eXtensible Markup Language（可以延伸或扩展的标记语言），它的语法类似HTML，都是用标签来描述数据。HTML的标签是固定的，我们只能使用、不能修改；XML则不同，它没有预先定义好的标签可以使用，而是依据设计上的需要，自行定义标签。例如：

```
<?xml version="1.0" encoding="BIG5" ?>
  <宋词>
    <鹧鸪天>
      <作者>晏几道</作者>
      <内容>
        彩袖殷勤捧玉钟，当年拼却醉颜红。舞低杨柳楼心月，歌
        尽桃花扇底风。从别后、忆相逢，几回魂梦与君同。今宵
        剩把银缸照，犹恐相逢是梦中。
      </内容>
    </鹧鸪天>
  </宋词>
```

范例中，〈宋词〉、〈鹧鸪天〉、〈内容〉等都是自定义的标签。用这种标签来建立文件，当然比HTML的<H1>、<H2>、<P>等标签更清楚、更容易看懂。所以在电子商务的网络时代，用XML来建立数据，再由HTML来显示，将是设计网页的新方向。

1.1.2 XML的结构

XML用根元素及树状结构来定义数据，例如：

```
<?xml version="1.0" encoding="BIG5" ?>
  <唐诗>
    <五言绝句>
      <作者> 王维  </作者>
      <标题> 鹿柴  </标题>
      <内容> 空山不见人，但闻人语响。
              返景入深林，复照青苔上。
    </内容>
```

```

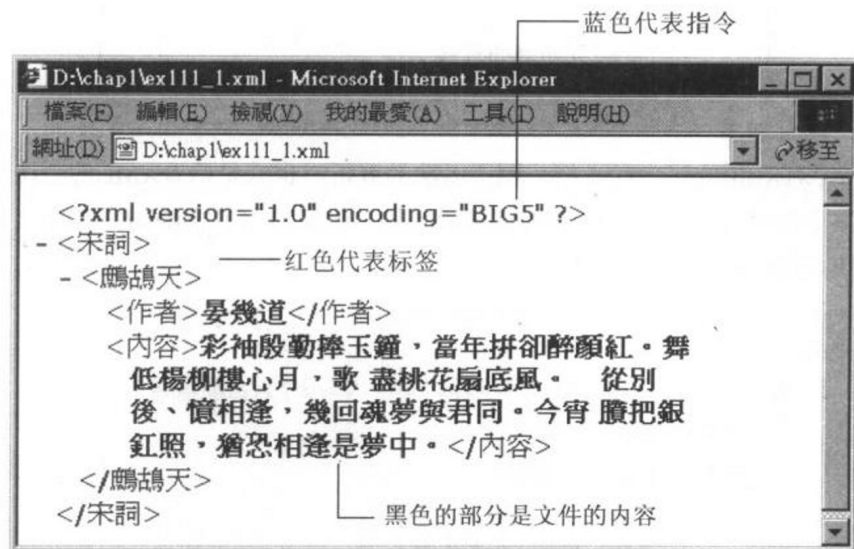
</五言绝句>
<五言绝句>
  <作者> 李白 </作者>
  <标题> 夜思 </标题>
  <内容> 床前明月光，疑是地上霜。
        举头望明月，低头思故乡。
  </内容>
</五言绝句>
</唐诗>

```

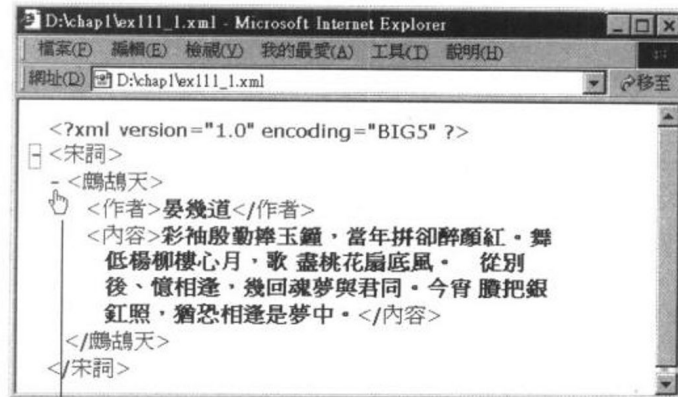
范例中，〈唐诗〉称为根元素，每一个〈五言绝句〉……〈/五言绝句〉段落，代表一条数据，〈作者〉、〈标题〉、〈内容〉这三个标签，则是每一条数据的项目。一份XML文件只能有一个根元素，数据的条数以及每一条数据中的项目，则没有任何限制。

1.1.3 IE5.0的支持

微软的IE5.0已经支持XML的规格，用IE5.0直接打开“*.xml”文件，画面上显示的是树状的数据结构，包括所有的指令、标签及数据内容。



IE5显示的数据结构，类似可折叠大纲，在〈-〉号上面按一下鼠标左键，可以隐藏内部的数据，标记符号则变成〈+〉。

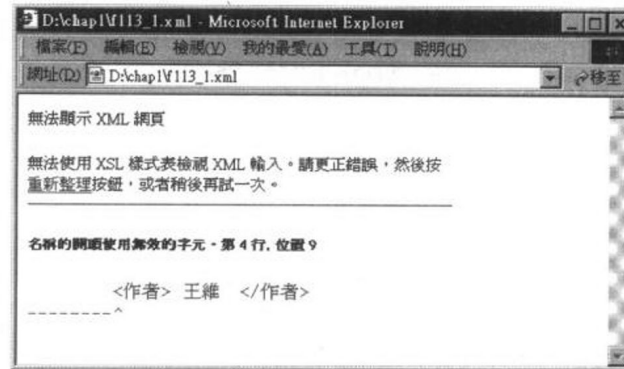


按一下左鍵



这是IE 5针对XML的第一种支持，属于语法的分析，也就是检查XML的语法是否正确，如果发现语法上有错误，会显示错误的信息。例如，将标签〈五言绝句〉改为〈五言绝句，就无法通过IE 5的检查。

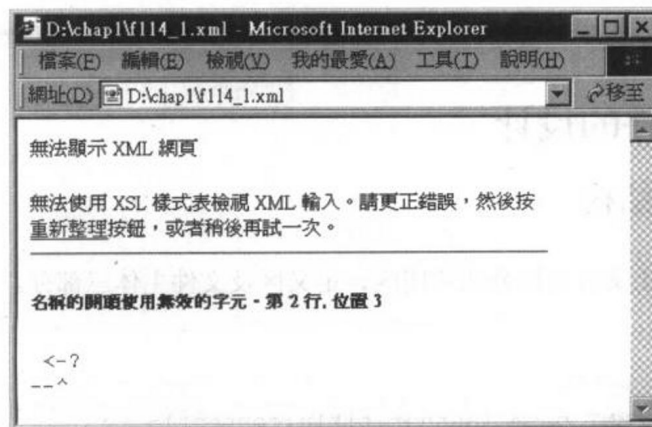
```
<?xml version="1.0" encoding="BIG5" ?>
<唐诗>
  <五言绝句    <-----少了右边的括号
    <作者> 王维  </作者>
    <标题> 鹿柴  </标题>
    <内容> 空山不见人，但闻人语响。
           返景入深林，复照青苔上。
    </内容>
  </五言绝句>
</唐诗>
```



1.1.4 在XML中使用中文

XML以Unicode作为内定的字符集（Character Set），所以不能直接在文件中显示中文字。

```
<?xml version="1.0" ?> <-----使用内定的字符集
<唐诗>
  <五言绝句>
    <作者> 王维 </作者>
    <标题> 鹿柴 </标题>
    <内容> 空山不见人，但闻人语响。
           返景入深林，复照青苔上。
    </内容>
  </五言绝句>
</唐诗>
```



如果要显示中文资料，或者要定义中文的标签，必须将字符集改成BIG5，例如：

```
<?xml version="1.0" encoding="BIG5" ?>
```

```
<唐诗>
```

```
<五言绝句>
```

```
<作者> 王维 </作者>
```

```
<标题> 鹿柴 </标题>
```

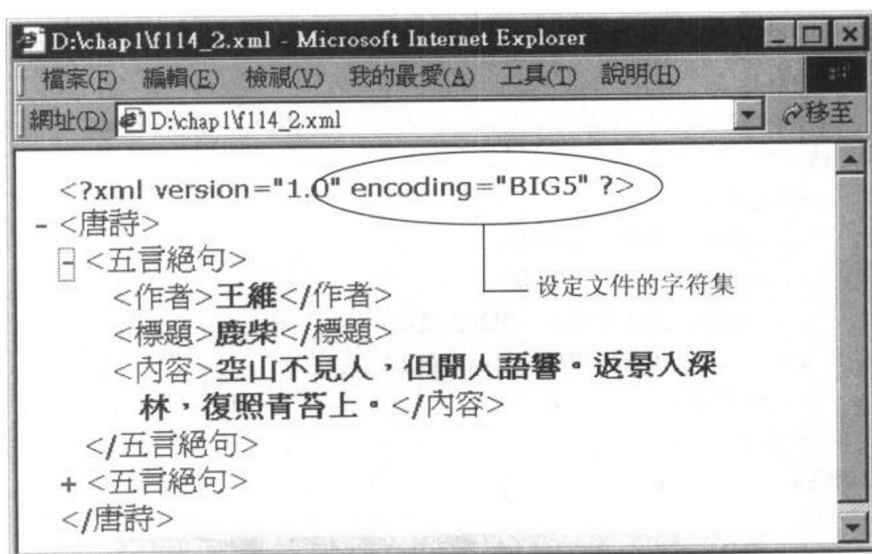
```
<内容> 空山不见人，但闻人语响。
```

```
返景入深林，复照青苔上。
```

```
</内容>
```

```
</五言绝句>
```

```
</唐诗>
```



1.2 XML文件的设计

1.2.1 文件的结构

一份完整的XML文件可以分成声明区、定义区及文件主体三部分。

```
<?xml version="1.0" encoding="BIG5" ?> <----- 声明区
```

```
<!DOCTYPE poem[ <----- 定义区
```

```
<!ELEMENT poem (author,title,content)>
```

```

<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT content (#PCDATA)>
]>

<poem> <-----文件主体区
  <author> 王维 </author>
  <title> 鹿柴 </title>
  <content>
    空山不见人，但闻人语响。
    返景入深林，复照青苔上。
  </content>
</poem>

```

1.2.2 声明区

XML文件的第一行一定是：

```
<?xml version="1.0" encoding="BIG5" ?>
```

如果要使用内定的字符集可以省略encoding的设定，变成：

```
<?xml version="1.0" ?>
```

XML文件如果需要引用外部的文件，例如设定样式的 " *. CSS "，也要在声明区中指定，例如：

```

<?xml version="1.0" encoding="BIG5" ?> <-----声明区
<?xml-stylesheet type="text/css"
    href="文件名称" > <-----引用其它文件

```

1.2.3 定义区

定义区用来设定文件的格式或自定义标签的相关性质，也叫做DTD (Document Type Definition)。定义区必须包含在<!DOCTYPE[.....]>段落中，例如：

```

<!DOCTYPE poem[ <-----定义区
  <!ELEMENT poem (author,title,content)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT content (#PCDATA)>

```