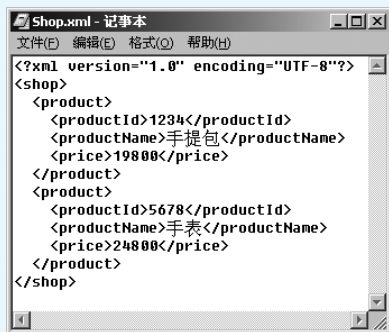


站中所需的信息。对比刚才的 CSV 文件，诸位有什么发现吗？只是瞥一眼，就能够看出来在 XML 文件中，因为标签为信息赋予了意义，所以分析起来更方便。但是，另一方面，文件的尺寸也变大了。刚才的 CSV 文件的大小不过 50 字节，而这个 XML 文件的大小是 280 字节，竟比 CSV 文件的 5 倍还多。文件尺寸增大，就意味着会占用更多的存储空间、需要更长的传输及处理时间。



```
<?xml version="1.0" encoding="UTF-8"?>
<shop>
  <product>
    <productId>1234</productId>
    <productName>手提包</productName>
    <price>19800</price>
  </product>
  <product>
    <productId>5678</productId>
    <productName>手表</productName>
    <price>24800</price>
  </product>
</shop>
```

图 11.11 购物网站的 XML 文件

另外在诸位平时所使用的应用程序中，不仅可以把文件保存成私有的数据格式，还可以把文件保存成通用的数据格式。以 Microsoft Excel 为例，在旧版本的 Microsoft Excel 2000 中，采用了 CSV 作为通用的数据格式。而在写作本书时发行的最新版本 Microsoft Excel 2002 中，就采用了 CSV 和 XML 两种格式（如图 11.12 所示）。这也算是一个今后还会继续同时使用 CSV 和 XML 的证据吧。

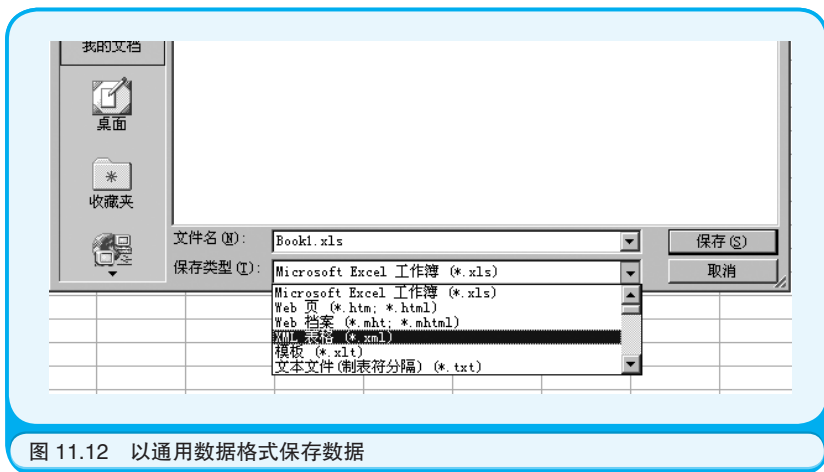


图 11.12 以通用数据格式保存数据

11.6 可以为 XML 标签设定命名空间

XML 文档并非互联网专用，但是 XML 确实是一种主要通过互联网在全世界的计算机之间交换数据时使用的数据格式。这样的话就有可能遇到一个问题：虽然标签的名字相同，但是标记语言的创造者们却为它们赋予了各种不同的含义。例如 `<cat>` 这个标签，有人用它来表示猫（CAT），也有人会用它来表示连接（conCATenate）（如图 11.13 所示）^①。

^① cat 除了表示猫，还是一个 Unix 命令的名称，该命令用于将多个文件连接在一起。在计算机行业，应该也有不少人更倾向于由 cat 这个词联想到连接，而不是猫。

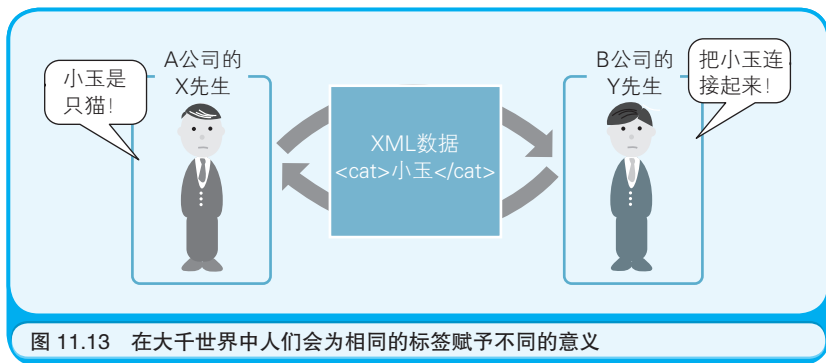


图 11.13 在大千世界中人们会为相同的标签赋予不同的意义

于是就诞生了一个 W3C 推荐标准——XML 命名空间 (Namespace in XML)，旨在防止这种同形异义带来的混乱。所谓命名空间，通常是一个能代表企业或个人的字符串，用于修饰限定标签的名字。在 XML 文档中，通过把 “xmlns=” 命名空间的名字 ” 作为标签的一个属性记述，就可以为标签设定命名空间。xmlns 即 XML Namespace (命名空间) 的缩写。通常用全世界唯一的标识符作为命名空间的名称。说到互联网世界中的唯一标识符，公司的 URI 就再好不过了。例如，在 XML 文件中，GrapeCity 公司的矢泽创建的标签 `<cat>` 就可以写成如下这种格式。

```
<cat xmlns="http://www.grapecity.com/yazawa">小玉</cat>
```

这样的话，就可以与使用了其他命名空间的 `<cat>` 标签相区分了。

在本例中，作为 `<cat>` 标签的命名空间设置的 `http://www.grapecity.com/yazawa`，仅作为一个全世界唯一的标识符来使用。就算把这个 URI 输入到 Web 浏览器的地址栏中，也并不会显示出相应的网页^①。

① 如果试着在浏览器中访问这个 URI，实际上会跳转到这个页面：<http://www.grapecity.com/jp/404.htm>。——译者注

11.7 可以严格地定义 XML 的文档结构

除了之前讲解过的“格式良好的 XML 文档”，还有一个词叫作“有效的 XML 文档”（Valid XML document）。所谓有效的 XML 文档是指在 XML 文档中写有 DTD（Document Type Definition，文档类型描述）信息。前面笔者没有说明，其实完整的 XML 文档包括 XML 声明、XML 实例和 DTD 三个部分。所谓 XML 声明，就是写在 XML 文档开头的、形如 `<?xml version="1.0" encoding="Shift_JIS"?>` 的部分。XML 实例是文档中通过标签被标记的部分。而 DTD 的作用是定义 XML 实例的结构。虽然也可以省略 DTD，但是通过 DTD 可以严格地检查 XML 实例的内容是否有效。

图 11.14 展示了一个写有 DTD 的 XML 文档。请把它想成是一个描述公司名称、地址和员工数量的 XML 文档。用“`<!DOCTYPE>`”和“`]>`”括起来的部分就是 DTD。DTD 定义了，在 `<mydata>` 标签中可以有一个以上的 `<company>` 标签；在 `<company>` 标签中可以包含 `<name>`、`<address>` 和 `<employee>` 标签。只要定义了这样的 DTD，当遇到那些虽然记录了公司名称和地址，但还没有记录员工数量的数据时，就可以判断出这不是一个有效的 XML 实例。

与 DTD 相同，还有一个名为 XML Schema 的技术也可用于定义 XML 实例的结构。在 XML 中，DTD 借用了可称得上是标记语言始祖的 SGML（Standard Generalized Markup Language，标准通用标记语言）语言的语法。而 XML Schema 是为了 XML 新近研发的技术，因此它可以对 XML 文档执行更严格地检查，例如检查数据类型或数字位数等。DTD 是 1996 年发布的 W3C 推荐标准，而 XML Schema 发布于 2001 年。今后将成为主流的是崭新的 XML Schema，而不是古老的 DTD。

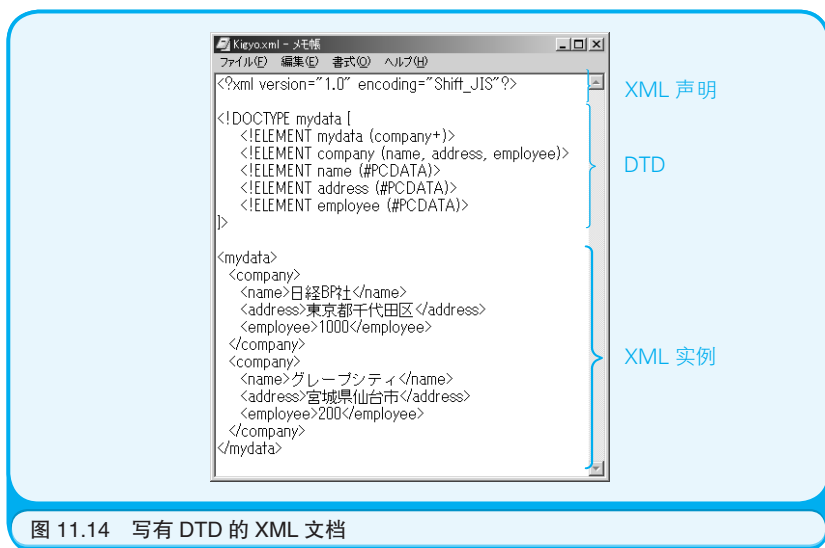


图 11.14 写有 DTD 的 XML 文档

11.8 用于解析 XML 的组件

前面介绍过，如果用 XML 文档记录信息，计算机就可以自动地进行处理。那么，编写处理 XML 文档的程序时应该怎么做呢？

也许会有人想：因为 XML 文档是纯文本文件，所以无论是用 BASIC 还是 C 语言，只要用某种编程语言编写一个能够读写文件的程序就可以了……这当然没有错！但是，如果要亲手从零开始编写这样的程序，就太麻烦了。像是切分标签之类的处理，即便 XML 文档的内容不同，其步骤也大致相同。要是有人能提供现成的这部分处理的代码就好了——这样想的人应该不止笔者一个吧。

的确存在着用于处理 XML 文档的程序组件。比如已成为 W3C 标准的 DOM（Document Object Model，文档对象模型）以及由 XML-dev 社区开发的 SAX（Simple API for XML）。其实无论是 DOM 还是 SAX，

都只是组件的规范，实际的组件是由某个厂商或社区提供的。

如果使用的是 Windows，那么就on应该已经安装了一个由微软提供的、遵循了 DOM 规范的组件（一个名为 msxml3.dll 的 DLL 文件）。下面我们就使用 VBScript 编程语言，试着编写一个实验程序吧。用记事本编写出如代码清单 11.1 所示的程序，保存到名为 TestProg.vbs 的文件中，这个文件要和之前所编写的 MyPet.xml 放置在同一个文件夹中。双击 TestProg.vbs 的图标即可运行该程序（如图 11.15 所示）。这个程序的功能是读取 MyPet.xml 文件的内容，显示出每种宠物的名字。诸位没有必要去详细了解这个程序的逻辑，知道有简单的方法可以处理 XML 文档就足够了。

代码清单 11.1 使用了 DOM 的程序

```
Set obj = CreateObject("Microsoft.XMLDOM")
obj.async = False
obj.Load "MyPet.xml"
s = ""
For i = 1 To obj.documentElement.childNodes.length
    s = s & obj.documentElement.childNodes.Item(i - 1).nodeName
    s = s & "... "
    s = s & obj.documentElement.childNodes.Item(i - 1).Text
    s = s & vbCrLf
Next
MsgBox s
```



图 11.15 代码清单 11.1 的执行结果

11.9 XML 可用于各种各样的领域

通过使用 XML，诞生了各种各样的标记语言（如表 11.2 所示）。以往的软件厂商在存储数学算式、多媒体数据等数据时，使用的都是自家应用程序的私有格式。然而在未来，作为世界标准的 XML 格式的标记语言将成为主流。即使是现在，也已经涌现出了一批成为 W3C 建议标准的标记语言。

表 11.2 用 XML 定义的标记语言示例

名称	用途	有关的企业或组织
XSL	为 XML 中的信息提供显示格式	W3C
MathML	描述数学算式	W3C
SMIL	把多媒体数据嵌入到网页中	W3C
MML	描述电子病历	电子病历研究会
SVG	用向量表示图形数据	W3C
JepaX	表示电子书	日本电子出版协会等
WML	表示移动终端上的内容	WAP Forum
CHTML	表示手机上的内容	Acces 等 6 家公司
XHTML	用 XML 定义 HTML4.0	W3C
SOAP	实现分布式计算	W3C

为了实现各自的目的，每一种标记语言中都定义了各种各样的标签。例如，在描述数学算式的 MathML（Mathematical Markup Language，数学标记语言）中，就定义了表示根号、乘方或分数等数学元素的标签。

$$aX^2 + bX + c = 0$$

比如上面的这个方程，如果用 MathML 描述的话，结果就会如图 11.16 所示。



```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
<mi>a</mi>
<msup>
<mi>x</mi>
<mn>2</mn>
</msup>
<mo>+</mo>
<mi>b</mi>
<mi>x</mi>
<mo>+</mo>
<mi>c</mi>
<mo>=</mo>
<mn>0</mn>
</mrow>
</math>
```

图 11.16 用 MathML 描述的算式

SOAP (Simple Object Access Protocol, 简单对象访问协议) 可用于分布式计算。所谓分布式计算, 就是把程序分散部署在用网络连接起来的多台计算机上, 使这些计算机相互协作, 充分发挥计算机整体的计算能力。简单地说, SOAP 就是使运行在 A 公司计算机中的 A 程序, 可以调用运行在 B 公司计算机中的 B 程序。

SOAP 的出现使过去的分布式计算技术变得更容易使用, 也更通用。无论是调用程序时所需的参数信息, 还是程序执行后的返回结果, 都可以用通用的数据格式 XML 表示 (如图 11.17 所示)。另一方面, SOAP 收发数据时所使用的传输协议并不固定, 凡是能够收发 XML 数据的协议均可使用。一般情况下使用的是 HTTP 或 SMTP 协议。可以说 SOAP 的诞生使得人们可以更加轻松地构建分布式计算环境了。

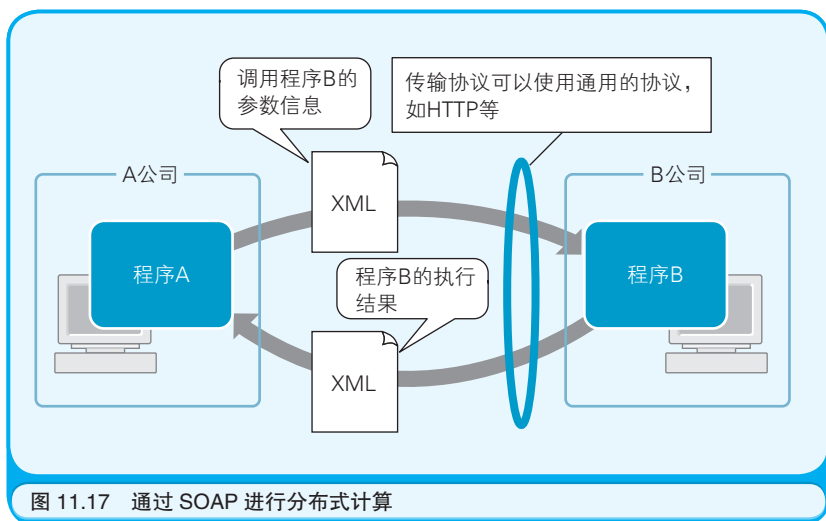


图 11.17 通过 SOAP 进行分布式计算

☆ ☆ ☆

XML 受到了众人的瞩目，在各种各样的场景中都可以见到它的身影，这已经是不折不扣的事实了，而且还会继续诞生新的 XML 的使用方法。但是请不要认为这等同于“今后所有的数据都应该是 XML 格式的”。因为 XML 只有在充当通用数据格式时才有价值。也就是说，只有在像互联网那样的环境中，运行在不同机器中的不同应用程序相互联结，XML 才会大有作为。只有一台独立的计算机，或者只在一家公司内部的话，使用 XML 格式存储数据反而体现不出优势，仅仅是文件的尺寸变大从而浪费存储空间罢了。

同样地，在分布式计算中，如果是由不同种类的机器互联组成的系统，那么使用基于 XML 的 SOAP 才是有意义的。反之如果环境中的机器和应用程序全部来自同一厂商，那么使用厂商自己定制的格式而并非基于 XML 的格式，反而可以更加快捷地处理信息。XML 是通用

的，但它不是万能的。笔者会把 XML 中的 X 看作是 eXchangable（可交换的）而并非是 eXtensible（可扩展的），诸位赞同这种看法吗？

下一章是本书的最后一章，笔者将讲解由各种技术组合而成的计算机系统。敬请期待！