

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/Tr/html4/loose.dtd">
3 <html lang="ja" xmlns:og="http://ogp.me/ns#" xmlns:mixi="http://mixi-platform.com/ns#"
4   xmlns:fb="http://www.facebook.com/2008/fbml">
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
7 <meta http-equiv="Content-Script-Type" content="text/javascript">
8 <meta http-equiv="Content-Style-Type" content="text/css">
9 <meta http-equiv="X-UA-Compatible" content="IE=edge">
10 <meta name="keywords" content="情報技術, IT, 白根研所, itpro, IT Pro, ニュース, 解説, コラム, 情報システム, イン
    ネット, セキュリティ, ネットワーク">
11 <meta name="description" content="白根研所が運営する、IT（情報技術）にかかわるプロフェッショナルに向けた総合
    情報サイト。ニュースだけでなく、詳細な解説／コラムやネットの双方向性を利用したコンテンツを提供。">
12 <meta name="viewport" content="width=device-width">
13 <meta name="robots" content="NOODP">
14 <meta name="googlebot" content="NOODP">
15 <meta name="robots" content="NOYDIIE">
16 <link rel="alternate" type="application/rss+xml" title="RSS"
17   href="http://itpro.nikkeibp.co.jp/rss/develop.rdf" />
18 <title>白根ソフトウェア | Itpro</title>
19 <link rel="SHORTCUT ICON" href="/images/s/itpro/2013/favicom.ico">
20 <link href="/css/s/itpro/2012/common.css" rel="stylesheet" type="text/css">
21 <link href="/css/s/gnavi/201406.css" rel="stylesheet" type="text/css">
22 <link rel="stylesheet" href="/css/s/itpro/2012/theme/new.css" type="text/css">
23 <link href="/css/s/itpro/2012/ad.css" rel="stylesheet" type="text/css">
24 <script type="text/javascript">
25   var now_theme = 'develop';
26   var now_subtheme = 'new';
27   var rankingType = 'subtheme';
28   var ranking filter a = 'atpene-new';

```

图 11.2 在记事本中显示图 11.1 所示网页的 HTML 源代码

## 11.2 XML 是可扩展的语言

正如其名，XML 是一种标记语言。XML 文件的扩展名一般是 .xml（使用别的也可以）。下面请诸位从 Windows 的“开始”菜单中打开“搜索”功能，找找各自的计算机中有没有 XML 文件。笔者就在自己的计算机中找到了一个名为 iuhist.xml 的 XML 文件，该文件位于文件夹 C:\Program Files\WindowsUpdate\V4 中。接下来就试着用记事本打开这个文件（也请诸位试着打开自己找到的 XML 文件）（如图 11.3 所示）。



```

<?xml version="1.0"?>
<Items xmlns="x-schema:http://schemas.windowsupdate.com/u/resultschema.xml"><item Status
xmlns="" timestamp="2002-11-01T11:15:39"><identity
itemID="win2k.windows2000.ver_platform_win32_nt.5.0.x64.ja...2195...com_microsoft.q327696_is5_
0_sp4.5757.1_10_101_0
name="G327696_IS5_0_SP4_5757"><publisherName>com_microsoft</publisherName></identity><d
escription priority="3" hidden="0"><size>2757744</size><descriptionText><title>G327696: セキュ
リティ問題の修正プログラム</title><url href="/msdownload/update/v3/static/bui/ja/eula.htm"/>
この修正プログラムによって、IIS 5.0 の脆弱性が修正されます。今すぐダウンロードして、
この脆弱性を解決してください。この修正プログラムに関する詳細は、近期中にこのページで
公開されます。お使いのコンピュータの安全を一刻も早く保護するため、この修正プログラム
は詳細な情報の公開よりも先に提供しています。</descriptionText></description><platform
name="ver_platform_win32_nt"><processorArchitecture>x86</processorArchitecture><version
major="5" minor="0" build="2195" servicePackMajor=""
servicePackMinor=""></platform><installStatus value="COMPLETE"
needsReboot="1"/></item></Items></item Status xmlns=""
timestamp="2002-11-01T11:14:18"></identity>

```

图 11.3 打开了 XML 文件 iuhist.xml，可以看到里面使用了标签

可以看到 XML 文件也使用了标签。在 `iuhist.xml` 中就有 `<publisherName>` 和 `<processorArchitecture>` 等标签，而且很有可能这两个标签表示的就是“发行者的名字”和“处理器的架构”。

那么是 XML 规定了这些标签吗？答案是否定的。XML 本身并不会限定标签的种类，反倒是允许 XML 的使用者随心所欲地创建标签。也就是说，在“<”和“>”中的单词可以是任意的。这就是所谓的“可扩展”。在 HTML 中，我们只能使用由 HTML 定义出的那若干种标签，因此 HTML 是固定的标记语言。与此相对，XML 是可扩展的标记语言。也许诸位会感到有些混乱，但是只要回顾之前的讲解，就应该能清楚地区分 HTML 和 XML 了。

### 11.3 XML 是元语言

XML 并没有限定标签的使用方式，使用什么样的标签都可以。可以说 XML 仅仅限定了进行标记时标签的书写格式（书写风格）。也就是说通过定义要使用的标签种类，就可以创造出一门新的标记语言。通常把这种用于创造语言的语言称作“元语言”。例如，我们可以使用 `<dog>` 和 `<cat>` 等标签，创造一种属于自己的标记语言——宠物语言。不过，就算新语言是自己创造的，也毕竟属于 XML 格式的标记语言，所以不遵循一定的规范是不行的。如果只是在文档中胡乱地堆积标签，则无法称之为符合 XML 格式的语言。表 11.1 中列出了作为元语言的 XML 中的约束。因为这些约束都很简单，所以请诸位先来粗略地浏览一下。

表 11.1 XML 中的主要约束

约束	示例
XML 文档的开头要写有 XML 声明, 表明使用的 XML 版本和字符编码	<code>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</code>
信息要用形如“< 标签名 >”的开始标签和形如“</ 标签名 >”的结束标签括起来	<code>&lt;cat&gt; 小玉 &lt;/cat&gt;</code>
标签名不能以数字开头, 中间也不能含有空格	不能用 <code>&lt;5cat&gt;</code> 或 <code>&lt;my cat&gt;</code> 作标签名
由于半角空格、换行符、制表符 (TAB) 都会被视为空白字符, 所以在文档中可以任意地换行或缩进书写	( 请参考图 11.4 )
对于没有内容的元素, 不但可以写成“< 标签名 ></ 标签名 >;”, 还可以写成“< 标签名 />”	<code>&lt;cat&gt;&lt;/cat&gt;</code> 和 <code>&lt;cat/&gt;</code> 是等价的
标签名区分大小写	<code>&lt;cat&gt;</code> 、 <code>&lt;CAT&gt;</code> 和 <code>&lt;Cat&gt;</code> 互不相同
标签中可以再嵌套标签以表示层级结构, 但不能交叉嵌套	<code>&lt;pet&gt;&lt;cat&gt; 小 玉 &lt;/cat&gt;&lt;/pet&gt;</code> 正确, <code>&lt;cat&gt;&lt;pet&gt; 小玉 &lt;/cat&gt;&lt;/pet&gt;</code> 错误
在 XML 声明的后面, 必须有且只有一个“根元素”, 该标签包含了所有其他的标签	<code>&lt;pet&gt;……其他的标签……&lt;/pet&gt;</code>
在开始标签中, 可以以“属性名 = 属性值”的形式, 加入任意的属性	<code>&lt;cat type="三色猫"&gt; 小玉 &lt;/cat&gt;</code>
如果要在内容中使用“<”“>”“&”“”和“'”这 5 个特殊符号, 要把它们写成“&lt;”“&gt;”“&amp;”“&quot;”和“&apos;”	<code>&lt;cat&gt; 小玉 &amp;amp; 小老虎 &lt;/cat&gt;</code>
只要用“<![CDATA[”和“]]>”把内容括起来, 就可以在里面直接使用“<”“>”“&”“”和“'”这 5 个特殊符号了。这种写法适用于要书写大量特殊符号的场景	<code>&lt;cat&gt;&lt;![CDATA[ 小 玉 &amp; 小 老 虎 &amp; 咪 咪 &amp; 小 哆 啦 ]]&gt;&lt;/cat&gt;</code>
注释的写法是用“<!--”和“-->”把注释的内容括起来	<code>&lt;!-- 这是注释 --&gt;</code>

XML 的数据是纯文本格式的, 也就是说只包含字符。通常把遵循了 XML 的约束编写出的文档称为“XML 文档”; 把保存着 XML 文档的文件称为“XML 文件”。可以使用记事本等文本编辑器编写 XML 文件。

图 11.4 展示了一个用描述宠物的标记语言编写的 XML 文件示例。其中使用了 3 种标签：<pet>、<cat> 和 <dog>。虽然标签的名字是由笔者自己决定的，但是在标签排列和 XML 声明等方面遵循了 XML 的约束，所以是一个良好的 XML 文件。

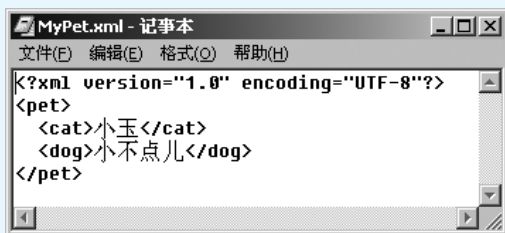


图 11.4 描述宠物的标记语言

我们把图 11.4 所示的文件命名为 MyPet.xml 并保存，然后再用 Web 浏览器打开该文件看看。当然，由于它不是 HTML 文件，所以不会显示成网页。但是现在的 Web 浏览器都集成了 XML 解析器，可以用这个功能来检查 XML 文件的书写格式。如果用 Internet Explorer Web 浏览器打开 MyPet.xml，就可以看到为了便于理解，里面的关键词、标签以及其他信息都用不同的颜色区分了出来。虽然图 11.5 是黑白的，但实际在屏幕上最开始的 1 行是蓝色的。在 <pet> 等标签中，表示标签开始和结束的符号“<”“</”和“>”也都是蓝色的，而 pet 和 cat 等标签的名字是褐色的。用标签括起来“小玉”和“小不点儿”则是黑色的。

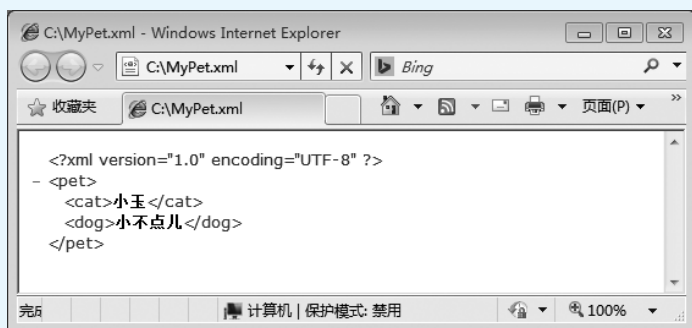


图 11.5 用 Internet Explorer Web 浏览器打开图 11.4 所示的 XML 文件

通常把遵循 XML 约束、正确标记了的文档称作“格式良好的 XML 文档”（Well-formed XML Document）。换言之，只要能通过 XML 解析器的解析，就是格式良好的 XML 文档。下面我们做一个实验，将 MyPet.xml 中的 `</cat>` 删除，保存后用 Web 浏览器再次加载该文件。因为 XML 约束中规定，标签必须以 `< 标签名 >`、`</ 标签名 >` 的形式成对儿出现，所以如果删除了 `</cat>` 而只留下 `<cat>` 的话，就不再是格式良好的 XML 文档了。这导致 XML 解析器不能正确解析，在 Web 浏览器上自然也就无法正确显示了（如图 11.6 所示）。诸位在自己手动创建 XML 文档的时候，也可以利用 Web 浏览器带有的 XML 解析器，检查 XML 文档的格式是否正确。

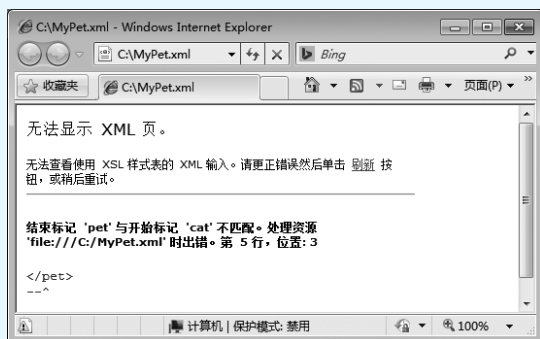


图 11.6 打开了不符合 XML 规范的 XML 文档

## 11.4 XML 可以为信息赋予意义

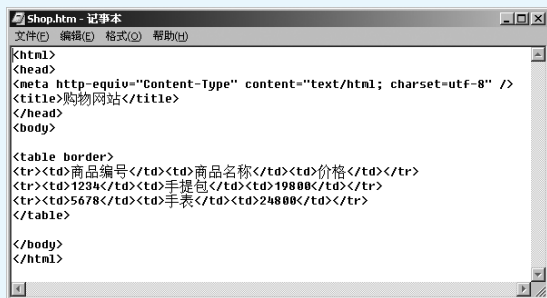
现在，诸位已经充分理解为什么说 XML 是可扩展的标记语言了吧？但是随之又产生出了一个新的疑问——XML 到底有什么用呢？要了解 XML 的用途，就要先了解 XML 的诞生过程。

众所周知，网页的出现使互联网得到了普及。网页是指使用 HTML 规定好的标签，将字符串和图片显示在 Web 浏览器上的页面。毫无疑问的是浏览网页的是计算机的用户，也就是人。例如一个购物网站，浏览网站中页面的是人，确认商品价格的是人，最后下单订购商品的还是人。

既然是用计算机来购物又学会编程了，就会想编写这样一个程序让购物变得更轻松：能够自动检查多个购物网站上的商品价格，然后自动在报价最低的网站上下单。但是如果网站只提供了 HTML，那么这个程序几乎不可能完成。因为 HTML 中规定的各种标签只能用来指定

信息的呈现样式，而不能表示信息的含义。

请看图 11.7 所示的 HTML 文件。如果把这个 HTML 文件显示在 Web 浏览器上（如图 11.8 所示），那么对人来说，商品编号、商品名称和价格是可以区分出来的。例如，虽然 1234 和 19800 都是数字，但是人们还是知道 1234 是商品编号，而 19800 是价格。但是，在 HTML 的标签中，并没有可以区分商品编号、商品名称和价格的标签。<table>、<tr> 和 <td> 只表示会以表格的形式呈现信息。作为程序要处理的数据格式，从图 11.7 所示的 HTML 文件中提取出商品编号、商品名称和价格的过程将非常繁琐。那么像下面这样做如何呢？首先定义出 <productId>、<productName>、<price> 等标签，然后用它们表示商品编号、商品名称、价格等信息。程序加载了带有这些标签的文件后，就能够轻松地识别出商品编号、商品名称和价格了，因为信息的含义已经用这些标签标记出来了。



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>购物网站</title>
</head>
<body>

<table border>
<tr><td>商品编号</td><td>商品名称</td><td>价格</td></tr>
<tr><td>1234</td><td>手提包</td><td>19800</td></tr>
<tr><td>5678</td><td>手表</td><td>24800</td></tr>
</table>

</body>
</html>
```

图 11.7 购物网站的 HTML 文件示例



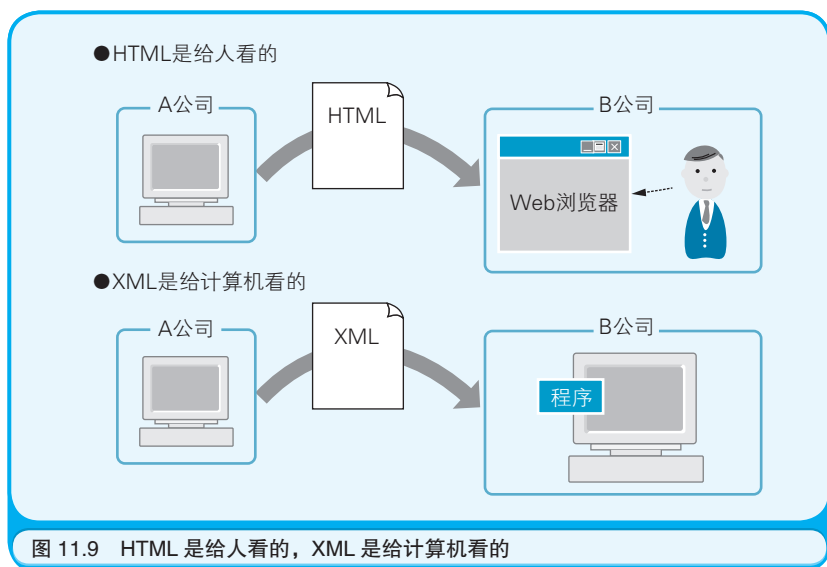
图 11.8 人们倒是可以区分出商品编号、商品名称和价格，但是……

在商业领域中存在着不计其数的信息，蕴涵着各种各样的意义。行业不同，信息的类型也就不同。并且随着时代的发展，新兴行业还在不断地涌现。如果要适用于所有行业，那么就算 HTML 的标签再多也还是不够用。于是就发明出了 XML 这种元语言，而 HTML 的用途就仅限于信息的可视化了，自始至终都用于展现网页。这也就是要告诉大家：今后请使用更加灵活的 XML 为各个行业、各个特殊用途创建标记语言。也就是说，XML 的主要用途是为在互联网上交换的信息赋予意义（如图 11.9 所示）。当然，在互联网以外的场景也可以使用 XML。只不过在 XML 诞生的过程中互联网一直伴随其左右。

在互联网的世界中，有一个叫作 W3C（World Wide Web Consortium，万维网联盟）的机构。该机构以“W3C 推荐标准”的形式制定了一系列标准。XML 于 1996 年成为了 W3C 的推荐标准（XML 1.0）。这之后，人们使用 XML 这种元语言，又定义出了新的网页标记语言 XHTML（Extensible Hypertext Markup Language，可扩展超文本标记语言），该语言也于 2000 年成为了 W3C 推荐标准。早晚有一天，



XHTML 会取代现行的 HTML (HTML 4.0), 成为编写网页的主流标记语言<sup>①</sup>。



## 11.5 XML 是通用的数据交换格式

W3C 的推荐标准是不依赖于特定厂商的通用规范。因此可以认为成为 W3C 推荐标准的 XML 是一种通用的数据交换格式。也就是说, 如果某家厂商的某个应用程序把数据保存到了 XML 文件中, 那么其他厂商的另一个应用程序就应该可以通过加载这个 XML 文件来使用数据。除此之外, XML 也可以在同一个厂商的不同应用程序之间交换数据。

XML 并不是第一个跨越了厂商或应用程序差异的通用数据交换格

<sup>①</sup> 原书于 2003 年出版, 那时还没有 HTML5。——译者注

式。在计算机行业，长久以来一直把 CSV（Comma Separated Value，逗号分隔值）作为通用数据交换格式沿用至今。下面就试着对比一下 XML 和 CSV 吧。

与 XML 一样，CSV 也是仅由字符构成的纯文本文件。一般情况下，CSV 文件的扩展名为 .csv。正如其名，在 CSV 文件中，记录的是经过“,”（半角逗号）分割后的信息。例如，上一节提到的购物网站中的商品信息如果用 CSV 表示的话，就如图 11.10 所示。其中，字符串要用“”（半角双引号）括起来，而数字则直接书写。每一件商品的记录（有一定意义的信息的集合）占一行。



图 11.10 购物网站的 CSV 文件

在 CSV 中，只记录了信息本身，而并没有为各个信息赋予意义。可以说在这一点上，还是 XML 更胜一筹。既然这样的话，是不是说今后 CSV 将被淘汰，只剩下 XML 还在使用呢？答案是否定的。CSV 和 XML 都会继续存在下去，因为它们各有千秋。不仅是计算机行业，其他行业亦是如此，如果有多个方法可以达到相同的目的，那么这些方法就自然会各有优劣。

请浏览一下图 11.11 所示的 XML 文件，里面使用了 <shop>、<product>、<productId>、<productName> 和 <price> 标签来描述购物网