

第17章 PHP与XML

- XML已经成为当今计算机处理数据方面不可或缺的重要部分，这使得XML成为计算机技术某个单独的领域，这个领域正是因为XML标准而且应用广泛而得以形成。
- 尽管对于基本PHP应用来说，跟XML打交道的时候不是很多，但因为XML本身的重要性，以及一个可预见未来的情况，介绍XML以及使用PHP处理XML是很必要的。尽管XML应用已经如此广泛，但对于PHP来讲，它仍然属于一个比较高级别的技术及应用层次，有很多概念和细节需要读者慢慢体会与理解，并且从更专门的文献中获取更多的参考信息。本章从最基础的方面向读者介绍XML，然后介绍如何使用PHP处理XML。



17.1 什么是XML

- XML是eXtensible Markup Language的缩写，其中文含义是扩展标记语言。就像本书最开始介绍的 HTML一样，XML也是一种标记语言。但它不仅仅是一种标记语言，更深层面地讲，它是一种存储数据的格式。XML使用一系列简单的标记描述数据，而这些标记可以方便地建立。虽然 XML占用的空间比二进制数据要占用更多的空间，但 XML极其简单并且易于掌握和使用。XML没有定义保存什么数据，也没有定义数据的格式，它只是定义了标记和这些标记的属性。
- 说明：HTML一样，XML（可扩展标识语言）是通用标识语言标准（SGML）的一个子集，它是用来描述网络上数据内容和结构的标准。尽管如此，XML不象HTML仅仅提供了在页面上显示信息的通用方法，XML则对数据赋予上下文相关的功能和动态功能，它继承了 SGML的大部分功能，却使用了不太复杂的技术。



17.2 建立一个简单的XML文件

- 介绍一些理论知识之后，下面来做一个实践。在本节中，读者可以使用最上手的文本编辑器，编辑一个完整的XML文档。这个XML文档用来描述图书信息的一些数据，如书名、作者、价格和出版社等。使用XML来描述图书的这些数据，将会使这些信息更加清晰易读。就是所要建立的XML文档的完整代码，读者可以在文本编辑器里键入如下所示的代码，然后按后缀名.xml保存，



17.3 深入认识XML文档

- 和所有标记语言一样，XML也有其特定的语法格式。本节讲深入XML文档内部，向读者介绍XML文档的细节部分。这些内容将包括，XML的声明、XML元素、XML标签及其属性、XML处理指令、XML文档中的注释以及文档定义类型等基本概念和用法。XML是一个比较简单的描述数据的方法，但这并不意味着XML很简单，实际上，关于XML的内容是非常丰富的，本节将从基本和实用两个方面出发，为读者尽可能多地介绍XML文档的各个方面。



17.3.1 XML声明

- XML的作用就是告诉XML处理程序：这个文件是按照XML文件的标准对数据进行描述的。当创建一个XML文件时，通常要以一个XML声明作为开始。是因为XML声明在文件中是可选内容，可加可不加，但W3C推荐加入这声明。所以通常把XML声明作为XML文件的第一行。一个最简单的XML声明如下所示。
- `<?xml version = "1.0"?>`
- XML的声明由“<?”开始，以“?>”结束。在“<?”后面紧跟着处理指示的名称，在这里是“xml”。XML声明中要求必须指定“version”的属性值，这个值定义了XML的版本。另外，声明中还有两个可选属性，分别是“standalone”和“encoding”。因此，一个完整的XML声明如下所示。
- `<?xml version="1.0" standalone="no" encoding="UTF-8"?>`



17.3.2 XML元素

- XML元素是XML文档的精髓，这小节就为读者介绍XML元素的有关知识。元素是XML文件内容的基本单元。从语法上讲，一个元素包含一个起始标记、一个结束标记以及标记之间的数据内容。其形式是：<标记>数据内容</标记>。例如下面是一个关于“国籍”的元素例子。
- <国籍>中国</国籍>
- 在17.2节的例子中有关于图书名称（title）的元素，如下所示。
- <title>Learning XML</title>



17.3.3 标记和属性

- XML的标记和HTML的标记在形式上大体相同，符号“<”和符号“>”之间的内容都称为标记。其基本形式如下所示。
- <标记名 属性名=”属性取值”>
- XML对于标记的语法规则比HTML要严格得多。首先，标记是必不可少的。任何一个形式良好的XML文件中至少要有一个元素。
- 注意：和HTML不同，XML对标签对大小写敏感的。
<Word>和<word>在XML中是两个不同的标签。



17.3.4 注释

- XML文档中除了XML声明和XML元素这些主要内容之外，还有其他几项内容，这小节到17.3.6小节将向读者介绍这些内容。
- 注释可以出现在XML文档的任何位置，甚至可以出现在根元素的前面或后面。XML文档注释以 `<!--` 开始，以 `-->` 结束，这个用法和HTML一致。例如下面的XML代码片段中使用了注释。
- `<!-- 这些是图书的有关信息 -->`
- `<book>`
- `<title>Learning PHP5</title>` `<!-- 图书的名称 -->`
- `<author>David</author>`
- `<publisher>White Water Press</publisher>`
- `<price>29.90</price>`
- `</book>`
- `<!-- 图书信息结束 -->`



17.3.5 处理指令

- 处理指令是用来给处理XML文档的应用程序提供信息的。也就是说，XML分析器可能对它并不感兴趣，而把这些信息原封不动地传给XML应用程序。这个应用程序应该会来解释这个指示，按照它所提供的信息进行处理，或者再把它原封不动地传给下一个应用程序。正如前面所看到的，XML声明就是一个处理指令。
- XML有很多处理指令，它们都遵循如下所示的格式。
- `<?处理指示名 处理指示信息?>`



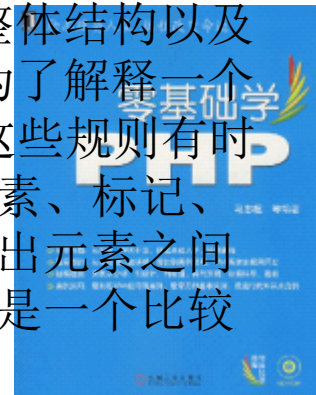
17.3.6 实体简介

- 在前面介绍标记属性时，提到了实体这一概念，本小节对实体再做一些简要的说明。可以这样理解实体，实体是在文档用来替换一些特殊标记符号的字符串。XML中或HTML中常见的实体如下所示。
- 实体< 代表小于符号<，lt就表示less than。
- 实体> 代表大于符号>，gt就表示greater than。
- 实体" 代表一个双引号”。
- 实体' 代表一个单引号（或撇号），
- 实体& 代表一个“与”符号&。



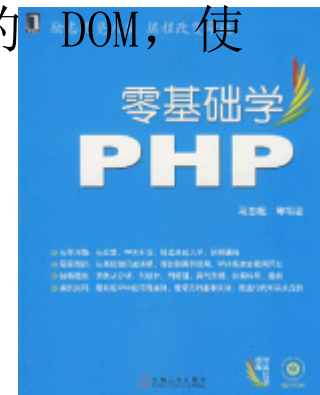
17.3.7 文件类型定义（DTD）

- 至此为止，已经向读者介绍了大部分关于XML的知识。通过对前面内容的学习，读者应该了解到，XML的精髓是基于信息描述的、能够体现数据信息之间逻辑关系的、可以确保文件的易读性和易搜索性的自定义标记。一个完全意义上的XML文件不仅应该是“形式良好的”，而且还应该是使用了自定义标记的XML文件。定义用来表示数据的标记，最常用的方法就是实用文档定义类型（Document Type Definition），即所谓的DTD。
- DTD指定在XML文档中出现的元素、这些元素出现的次序、元素相互嵌套的关系以及XML文档结构的其它详细信息，DTD是XML规范的一部分。一个XML文件必须遵守文件类型描述DTD（Document Type Definition）中定义的种种规定。DTD定义了文件的整体结构以及文件的语法。简而言之，DTD规定了一个语法分析器为了解释一个“有效的”XML文件所需要知道的所有规则的细节。这些规则有时很简单，有时会很复杂。比如说，列出所有有效的元素、标记、属性和实体，这就是一个比较简单的规则。又如，指出元素之间的内在联系，说明某个元素必须包含的其它元素，就是一个比较复杂的规则。下面就来学习DTD的基本用法。



17.4 在PHP中建立对XML的支持

- 虽然PHP在以前的版本中就已经提供了对 XML的支持，但是随着PHP5的出现，这种支持大大增强了。由于 PHP4对XML的支持比较有限，比如默认情况下只提供基于 SAX的解析器、PHP4 DOM没有实现W3C标准等。
- PHP5 包括彻底重新编写的和新增加的扩展，如 SAX 解析器、DOM、SimpleXML、XMLReader、XMLWriter 和 XSLT 处理程序。所有这些扩展都以 libxml2 为基础。除了自 PHP4 改进的 SAX 支持以外，PHP5 还同时支持符合 W3C 标准的 DOM 和 SimpleXML 扩展。默认情况下同时支持 SAX、DOM 和 SimpleXML。如果熟悉其他语言中的 DOM，使用 PHP 实现类似的功能会更简单。



17.5 用PHP函数处理XML文档

- Expat是PHP脚本语言的XML解析器（亦称为XML处理器），它提供了一系列函数可以使程序访问XML文档的结构和内容。
- Expat是一种基于事件的解析器。XML解析器有两种基本类型：
- 一是基于树型的解析器：将XML文档转换成树型结构。这类解析器分析整篇文章，同时提供一个API来访问所产生树的每个元素。其通用的标准为DOM，即文档对象模式，本章最后将会介绍。
- 二是基于事件的解析器，将XML文档视为一系列的事件。当一个特殊事件发生时，解析器将调用开发者提供的函数来处理。基于事件的解析器有一个XML文档的数据集中视图，也就是说它集中在XML文档的数据部分，而不是其结构。这些解析器从头到尾处理文档，并将类似于元素的开始、元素的结尾、特征数据的开始等等事件通过回调（callback）函数报告给应用程序。



17.5.1 XML解析器的建立和释放

- 函数xml_parser_create ()用来初始化一个XML解析器，即建立XML解析器的实例，该实例将用于以后的所有函数。这一点非常类似于PHP中MySQL函数的连接。该函数语法如下所示。
- resource xml_parser_create([string \$encoding])
- 该函数建立一个新的XML解析器并返回可被其它XML函数使用的资源句柄。该函数有一个可选参数，用来指定要被解析的XML输入的字符编码方式。PHP5开始，可以自动侦测输入的XML编码，因此\$encoding参数仅用来指定解析后输出数据的编码。默认输出的字符编码是 ISO-8859-1，而PHP5.0.2及以上版本是UTF-8。
- 函数xml_parser_free ()用来释放指定的XML解析器，该函数语法如下所示。
- bool xml_parser_free (resource \$parser)



17.5.2 处理XML元素的函数

- 函数xml_set_element_handler()用来处理建立起始和终止元素处理器，该函数用法如下所示。
- `bool xml_set_element_handler (resource $parser, callback $start_elem_handler, callback $end_elem_handler)`
- 函数xml_set_element_handler()的第1个参数\$parser是指向要调用处理器的XML解析器的指针参数 第2个参数\$start_elem_handler和第3个参数\$end_elem_handler是表示函数名称的字符串。如果处理器被成功的建立，该函数将返回TRUE。如果参数\$parser指向的不是合法的解析器，该函数将返回FALSE。
- 第2个参数\$start_elem_handler所指定的函数将在一个元素开始的时候调用，也就是说，当XML分析器遇到元素开始事件时，就会调用参数\$start_elem_handler所指定的函数。同理，遇到元素结束事件时，该函数就会调用参数\$end_elem_handler所指定的函数。



17.5.3 处理XML字符数据的函数

- 函数xml_set_character_data_handler()用来建立字符数据处理器，该函数的用法如下所示。
- `bool xml_set_character_data_handler (resource $parser, callback $handler)`
- 该函数是参数\$parser指向的XML 解析器指定字符数据处理函数。参数\$handler是一个表示函数名称的字符串。由参数\$handler所指定的函数，将在分析器遇到一个字符数据事件时调用。如果处理器被成功的建立，该函数将返回TRUE。如果参数\$parser指向的不是合法的解析器，该函数将返回 FALSE。
- 和函数xml_set_element_handler()极为类似，由参数\$handler所指定的函数必须含有两个参数，各参数解释如下所示。
- 第1个参数parser，是指向要调用XML 解析器的指针。
- 第2个参数data，是包含有字符数据的字符串。



17.5.4 解析一个XML文档

- 函数xml_parse()用来解析一个XML文档，该函数的语法如下所示。
- `int xml_parse (resource $parser, string $data [, bool $is_final])`
- 函数xml_parse()的第1个参数\$parser是一个指向XML解析器的指针，第2个参数是需要解析的数据集。可以多次对新的数据调用xml_parse()函数来分段解析一个文档，这时，只要在解析最后一段数据时将参数\$is_final的值设置为TRUE即可。第3个参数表示，如果被设置为TRUE，则data为当前解析中最后一段数据。



17.5.5 完整实例分析

- 前面4小节介绍了分析XML的主要函数，这小节将通过一个具体实例使读者了解这些函数在PHP程序中的具体用法。这小节使用的XML文档就是在17.2小节创建的17-1.xml。
- 这节的实例程序将首先在PHP程序中初始化XML分析器，然后为不同的XML事件定义不同的处理器，最后分析XML文档。



使用 PHP 函数处理 XML 文档的输出结果



17.6 使用SimpleXML处理XML

- 在PHP5中读取、操作、编写XML，如果处理的XML文档比较简单或比较小，那么通常会使用SimpleXML，必要的时候再加上DOM。SimpleXML和DOM是PHP中两种用来处理XML的常用的API，它们易于使用而且功能完善，而且SimpleXML和DOM可以相互操作。本节先为读者介绍SimpleXML的使用，下一节介绍DOM。



17.6.1 SimpleXML简介

- SimpleXML正像其名字所蕴含的含义一样，使用起来非常简单。如果XML文档不是很复杂、嵌套不太深、又没有混合内容的话，使用SimpleXML要比使用DOM简单得多。SimpleXMLAPI需要在PHP5的环境下使用，它包括和DOM的互操作性，以便编写XML文件和其它的一些支持。SimpleXML最适合简单的、类似记录的数据，比如从同一个应用程序其他部分传递来的XML文档或字符串。
- SimpleXML将XML文档转换成一个对象，下面是SimpleXML转换XML文档各个组成部分的情况分析。
- XML元素被转换成SimpleXMLElement对象的一个单一属性。如果同一层上有多个元素，那么这些元素会被存放在数组中。
- XML元素的属性可以使用关联数组来访问，数组的索引就是XML元素的属性名。XML元素属性的值就是关联数组的元素值。
- XML元素的字符数据将被转换成字符串。



17.6.2 载入XML文件的函数

- SimpleXML使用函数`simplexml_load_file()`载入一个XML文件，并将该XML文档解释为一个XML对象。该函数的原型如下所示。
- `object simplexml_load_file (string $filename [, string $class_name [, int $options]])`
- 函数`simplexml_load_file()`的第1个参数是要载入的XML文件名，第2个参数是可选的，如果指定该参数，函数`simplexml_load_file()`返回这个指定类的对象。第3个参数`$option`也是可选的，这个参数在实际应用中使用的情况很少，这里不再赘述。



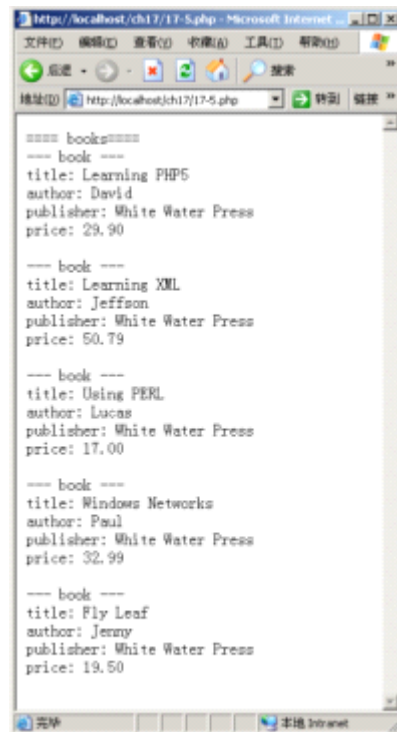
17.6.3. 载入XML字符串的函数

- Simple除了提供从文件载入XML的函数 `simplexml_load_file()` 之外， 还提供 一个载入XML字符串的函数——`simplexml_load_string()`， 该函数的语法如下所示。
- `object simplexml_load_string (string $data [, string $class_name [, int $options]])`
- 函数`simplexml_load_string()`的第1个参数是XML字符串， 第2个参数和第3个参数的意义同函数`simplexml_load_file()`， 这里不再赘述。



17.6.4 使用SimpleXML的实例

- 这节通过一较完整的实例，从整体上来了解SimpleXML的用法。本实例所载入的XML文档仍是17-1.xml。



```
===== books =====  
--- book ---  
title: Learning PHP5  
author: David  
publisher: White Water Press  
price: 29.90  
  
--- book ---  
title: Learning XML  
author: Jefferson  
publisher: White Water Press  
price: 50.79  
  
--- book ---  
title: Using PERL  
author: Lucas  
publisher: White Water Press  
price: 17.00  
  
--- book ---  
title: Windows Networks  
author: Paul  
publisher: White Water Press  
price: 32.99  
  
--- book ---  
title: Fly Leaf  
author: Jerry  
publisher: White Water Press  
price: 19.50
```

使用 SimpleXML 处理 XML 文档的输出结果



17.7 使用PHP的DOM库处理XML文档

- DOM是Document Object Mode的缩写。它是在浏览器中使用的、用JavaScript操作的W3C DOM规范。使用DOM可以很容易地将XML文档分析成一个树形结构。PHP中的DOM扩展内容比较庞大，限于篇幅本书不可能一一介绍，本节从宏观和总体角度介绍PHP的DOM扩展，同时挑选一些重要并且常用的函数向读者加以介绍。下面就为读者介绍如何通过DOM在PHP中处理XML文档。
- （1）在PHP中使用DOM，首先需要创建DOM的一个实例，即DOMDocument，如下代码所示。
- `$dom = new DOMDocument;`
- （2）建立好DOM的实例之后，就可以载入XML到该实例中。有两个方法可以实现载入XML，一是从一个字符串载入，使用函数loadXML()，代码如下所示。
- `$dom->loadXML('<string>books</string>');`
- 另一种是从文件载入，使用函数load()，代码如下所示。
- `$dom->load('17-1.xml');`
- （3）使用DOM对象的documentElement属性可以访问XML文档的根元素。演示了使用这个属性获取17-1.xml文档的根元素。



17.8 小结

- 本章向读者介绍了XML的基础知识，以及如何在PHP程序中完成对XML的处理。关于XML的讲述，主要涉及以下基础内容。
- 什么是XML。
- 如何建立一个XML文档。
- XML的声明。
- XML元素。
- XML的标记和属性。
- XML文档中的注释。
- XML处理指令。



17.8 小结

- 文档类型定义DTD。
- 关于如何在PHP程序中处理XML，本章主要简述如下内容。
- 使用PHP函数处理XML文档（Expat），包括XML解析器的建立与释放、处理元素的函数介绍、处理字符数据的函数介绍、解析XML文档的函数介绍及实例代码分析等。
- 使用SimpleXML处理XML，包括SimpleXML的简介、常用函数介绍和实例代码分析等。
- 使用PHP的DOM库处理XML文档，包括重要、常用函数介绍及其示例代码。

