**PHP與XML**

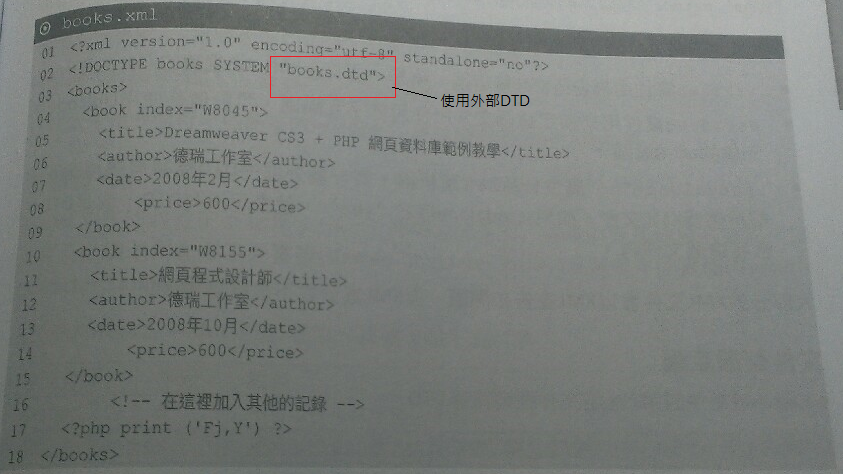
XML是e**X**tensible **M**arkup **L**anguage的縮寫，定義人類與電腦都能夠閱讀的結構化文件的語法。網頁標準語言是html，藉由標記與屬性讓瀏覽器識別。Xml是一種能夠在電腦間做資料交換的文件格式。

**XML基本組成**

1. 使用XML語言標示的文件
2. 可有可無的DTD或XML Schema，用來定義元素與屬性的用法。
3. 用來展示元素的樣式表
4. 解析XML文件的解析器

**XML結構**

基本架構

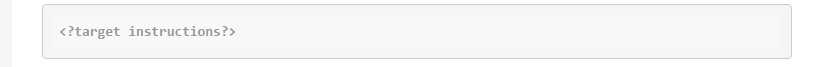
****

**處理指令**

處理指令是將XML文件的資訊傳送給讀取這個文件的應用程式的方式，可以包含程式或script本身

上圖中的<?php print (‘Fj,Y’)?>就是處理指令，此指令為顯示目前日期時間。

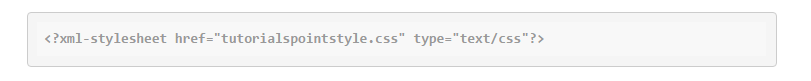
下面是處理指令的語法:



其中：

* target – 標示指令指向哪個應用程序。
* instruction - 字符，描述了應用程序要處理的訊息。

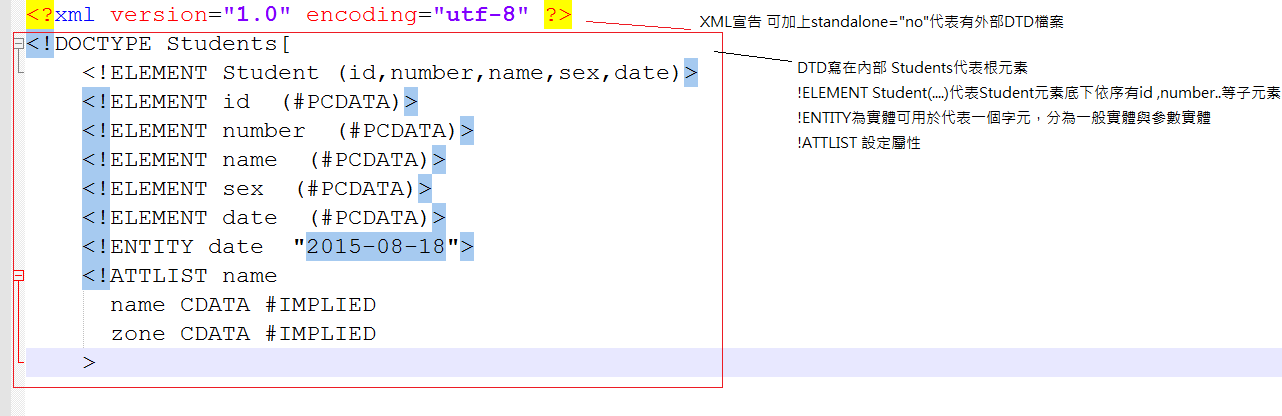
處理指令很少被使用。主要用於連結 XML 文檔到樣式表。下面是一个例子：



*target* 就是 xml-stylesheet。

*href="tutorialpointstyle.css"* 和 *type="text/css"* 就是數據或者目標應用程序用將要用來處理给定 XML 文件的*指令*。

使用內部DTD



CDATA是屬性型態，表示此屬性是一個字元或字元字串，如IMG元素的src屬性

#IMPLIED是屬性的預設或關鍵字，代表此屬性可有可無

(#REQUIRED:代表這個屬性是必要的

#FIXED:代表這個屬性是必要的且不能更改)

#PCDATA表示此元素內容可包含字元實體(<,>,”)

**DTD優缺點**

DTD是XML的附屬檔案，用來定義讀取文件的應用程式如何解釋文件內的標記，XML文件並不一定需要DTD文件，下列為DTD之優缺點。

DTD優點

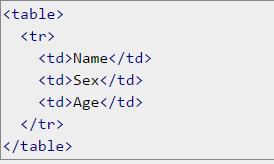
1. 當需要將標記應用到大量文件時，DTD將很有用處。
2. DTD保證指定的資料欄位會存在，或以指定格式傳送。
3. 當需要和其他作者或文件來源協調時。
4. 有DTD比較容易發現文件內的錯誤。

DTD缺點

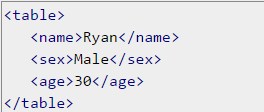
1. 需花時間製作，維護不易。
2. 減慢XML文件處理速度，在用戶端會有很大限制。
3. DTD與命名空間機制不合。

**XML的命名空間**

當一個XML文件內可能使用來自其他XML應用程式或XML詞彙的標記，例如一份XML裏頭有描述HTML的資料:



這時我們也自定義了XML的元素:



這時便會有所謂的命名衝突的問題，因為兩個程式片段中都包含了<table>這個元素，所以XML解析器會不清楚要如何分辨以及處理該文件。

因此我們就需要手動去做宣告的動作，將上面的程式片段宣告為HTML，而下面的文件宣告為自定義的文件。

透過 xmlns 這一個屬性來做宣告 Name Space 的動作：



上圖URI看起來都是網址，但是XML解析器並不是到這個網址擷取關於命名空間的資訊，它只是一組獨一無二的識別字，只是有些人會順便把命名空間相關的資訊放到此網址上面，好讓使用者可以輕易了解文件資訊。

**正確形式的XML**

如果XML文件語法有錯，XML規範嚴格禁止XML解析器試圖讀取或顯示文件。符合語法規則的XML文件，叫做正確形式(Well-Formed)的XML文件。

正確形式的文件，必須遵守下列規則:

1. 如果有指定XML宣告(可不用宣告)，宣告前不能空白或換行。
2. 除了空元素，每個元素必須有開始與結束標記。<number></number>
3. 空元素標記結尾必須包含 “ / ”符號。<br/>
4. 開始與結束標記巢狀位置必須正確，不可重疊。<tel><person></person></tel>
5. 開始標記<符號與元素名稱間不可空白。<tel>
6. 所有元素屬性值都必須使用單引號或雙引號包起來。<tel id=”1 ”>
7. 同一元素不可有相同名稱屬性存在。
8. 標記內不可有註解文字與處理指令。
9. 元素或屬性不可有未經脫逸的<或$符號。
10. XML文件只能有一個根元素。

**PHP的XML處理**

PHP處理xml方式(DOM.php有三種方法實作)

1. **DOM:**

適用於完整的xml文件，完全建立在記憶體內，dom物件將XML物件視為樹狀結構的節點，每個元素都可以用節點方式新增、修改或刪除。

所以愈大的xml文件需耗用較大量的資源。

1. **SAX-Simple API for XML:**

是一種事件驅動的PHP應用程式，SAX解析器會收到事件通知才呼叫指定的函數。

將xml視為字串資料串流，適合特定型態的所有元素執行重複工作，耗用記憶體較少。

1. **SimpleXML:**

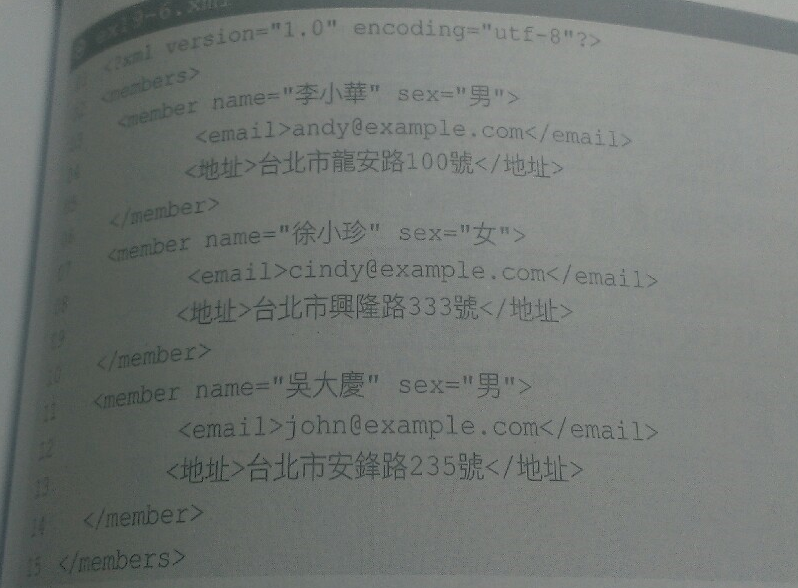
可快速開啟xml文件，將xml元素轉換成php資料型態，不像DOM建立整個XML文件，而是將XML元素儲存成PHP變數。

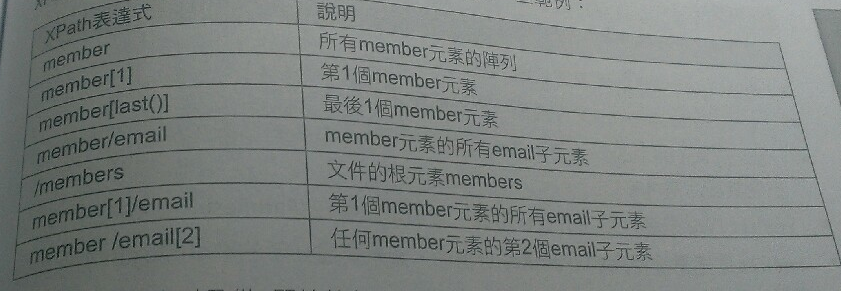
SimpleXML是存取xml資料最快速的方法。

**XPath**

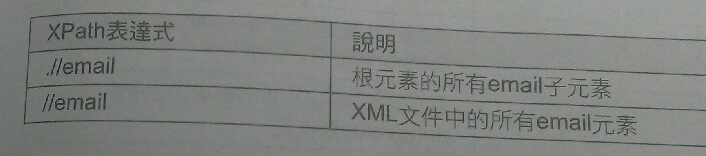
XPath是一個簡單的語言，類似檔案系統的路徑命名方式。不過XPath 是一種模式(Pattern)，可以選出 XML檔案中，路徑符合某個模式的所有節點出來。

XPath表達式

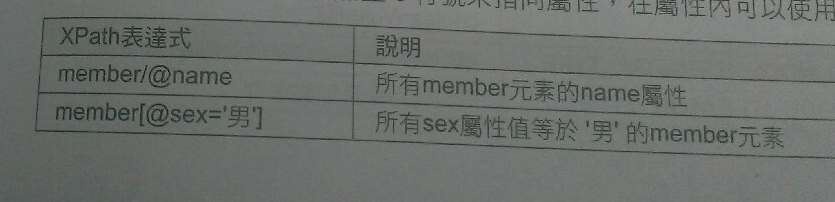




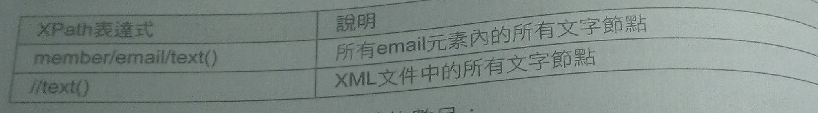
點號.代表XML文件的根元素，//符號代表所有子元素



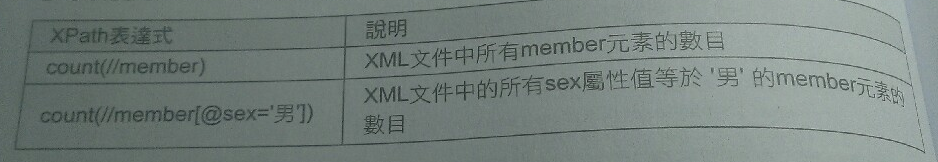
元素屬性名稱加上@指向屬性



要讀取元素內文字時使用text()方法:

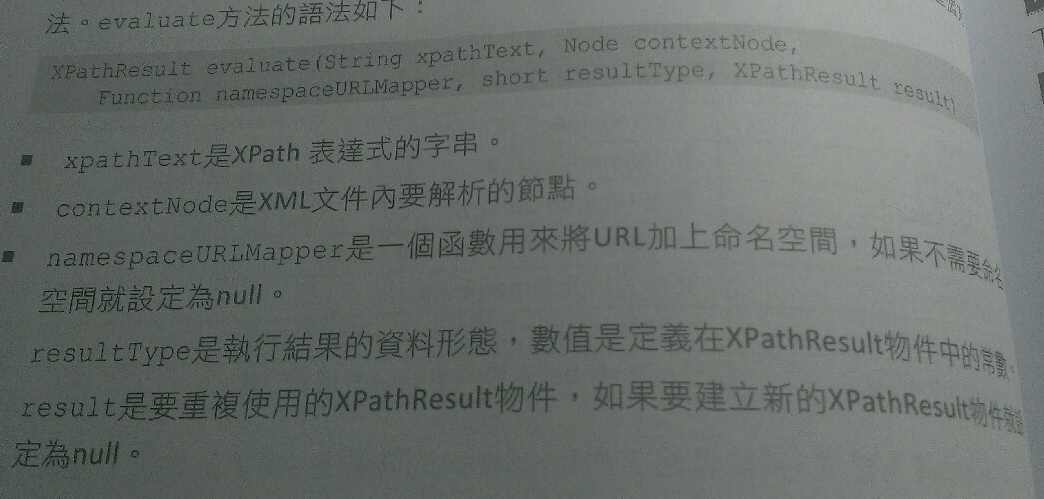


使用count()函數回傳節點數目:



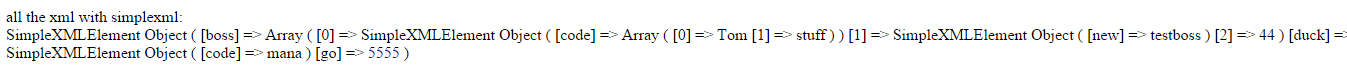
**解析XPath**

W3c使用Document物件的evaluate方法解析XPath表達式

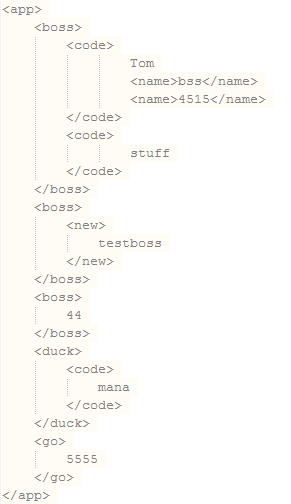


**補充**

Simplexml是將XML元素儲存成PHP變數，輸出資料如下



XML文件:



其中以SimpleXMLElement Object 陣列為節點將值放進陣列中，如果節點底下有子元素則再以SimpleXMLElement Object陣列(儲存子元素)為值。如果同一層有相同名稱的節點則索引變為數字索引，不然為關聯索引。

使用simplexml的XPath():$xml->xpath('boss/code');

傳回兩個符合條件的節點，以SimpleXMLElement Object 陣列傳回。

