

Lecture 1: 使用檔名：myXmlNameSpace0010.xml & myXmlNameSpace0011.xml

一、先 edit 一個 myXmlNameSpace0010.xml & myXmlNameSpace0011.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>
```

```
<inventory>
  <book >
    <title>通訊原理與應用</title>
    <author>藍國桐</author>
    <page> maybe &gt; 288 頁</page>
    <price>NT$ 350</price>
  </book>

  <book >
    <title>行動通訊系統</title>
    <author>李大明</author>
    <page>450 頁</page>
    <price>I think &lt; NT$ 400</price>
  </book>

  <book>
    <title>網路程式設計</title>
    <author>王李吉</author>
    <page>550 頁</page>
    <price>NT$ 450</price>
  </book>
</inventory>
```

```
<?xml version="1.0" encoding="Big5" ?>
```

```
<inventory>
  <CD >
    <title>傷心酒店</title>
    <author>江惠</author>
    <price>250</price>
  </CD>

  <CD >
    <title>等無人</title>
    <author>尤雅</author>
    <price>230</price>
  </CD>

  <CD>
    <title>妹力四射</title>
    <author>江惠妹</author>
    <price>350</price>
  </CD>
</inventory>
```

說明一、Now suppose you want to combine these documents into a single XML document.

The problem is that in the pooled document the new application would not be able to differentiate book title from CD title, book author from CD author, and book price from CD price.

說明二、The XML namespace mechanism provides an easier way to differentiate two or more elements, or two or more attributes, that have the same name by assigning each to a separate namespace.

二、edit 一個 myXmlNameSpace0012.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>
```

```
<inventory
```

```
  xmlns:BName="http://www.ntou.edu.tw/EE/Allen/books"
```

```
  xmlns:CName="http://www.ntou.edu.tw/EE/Allen/CDs" >
```

```
  <BName:book >
```

```
    <BName:title>通訊原理與應用</BName:title>
```

```
    <BName:author>藍國桐</BName:author>
```

```
    <BName:page> maybe &gt; 288 頁
```

```
      </BName:page>
```

```
    <BName:price>NT$ 350</BName:price>
```

```
  </BName:book>
```

```
  <BName:book >
```

```
    <BName:title>行動通訊系統</BName:title>
```

```
    <BName:author>李大明</BName:author>
```

```
    <BName:page>450 頁</BName:page>
```

```
    <BName:price>I think &lt; NT$ 400
```

```
      </BName:price>
```

```
  </BName:book>
```

```
  <BName:book>
```

```
    <BName:title>網路程式設計</BName:title>
```

```
    <BName:author>王李吉</BName:author>
```

```
  <BName:page>550 頁</BName:page>
```

```
  <BName:price>NT$ 450</BName:price>
```

```
</BName:book>
```

```
<CName:CD >
```

```
  <CName:title>傷心酒店</CName:title>
```

```
  <CName:author>江惠</CName:author>
```

```
  <CName:price>250</CName:price>
```

```
</CName:CD>
```

```
<CName:CD >
```

```
  <CName:title>等無人</CName:title>
```

```
  <CName:author>尤雅</CName:author>
```

```
  <CName:price>230</CName:price>
```

```
</CName:CD>
```

```
<CName:CD>
```

```
  <CName:title>妹力四射</CName:title>
```

```
  <CName:author>江惠妹</CName:author>
```

```
  <CName:price>350</CName:price>
```

```
</CName:CD>
```

```
</inventory>
```

Reserved
by W3C

Namespace name

說明三、`xmlns:BName="http://www.ntou.edu.tw/EE/Allen/books"`

說明四、The namespace name is the actual identifier of the namespace and should be a URI.

A namespace is identified using a URI, not because the XML processor or application needs to actually access any information at that URI, but rather **simply because URIs are globally unique**. When you create a namespace yourself (rather than use a namespace provided by another party), you should choose a permanent URI that you control, to avoid conflicting with someone else's namespace.

說明五、The term URI (Uniform Resource Identifier) is all-inclusive, referring to Internet resource-addressing strings that use any of the present or future addressing schemes. URIs currently include the following :

URLs (Uniform Resource Locator), which use traditional addressing schemes such as http (for example, <http://www.ntou.edu.tw>), ftp (for example, <ftp://ftp.microsoft.com/>), and mailto (for example, <mailto:someone@microsoft.com>).

URNs (Uniform Resource Names), which use the newer URN addressing scheme. This scheme is intended to address Internet resources in a way that is location-independent and is stable over time (unlike URLs, which often change). The following are several examples of URNs drawn from the W3C namespace specification: <urn:loc.gov:books>, <urn:ISBN:0-395-36341-6>, and <urn:w3-org-ns:HTML>. For complete information on URN syntax, see <http://www.ietf.org/rfc/rfc2141.txt>. (for URIs, see <http://www.w3.org/Addressing/>)

說明六、namespace 的命名原則：

1. The **xmlns** in the namespace declaration is a *predefined prefix* (you don't have to declare it) that is used specifically for defining namespaces.
2. **BName**, **CName** are the *namespace prefix*, which is a shorthand notation for the full namespace name.
3. A valid namespace prefix must begin with a **letter** or **underscore (_)**, followed by zero or more **letters**, **digits**, **periods (.)**, **hyphens (-)**, or **underscores (_)**.
4. Beginning a prefix with the letters **xml** (in any case combination) is **reserved** for prefixes defined by XML-related specifications.

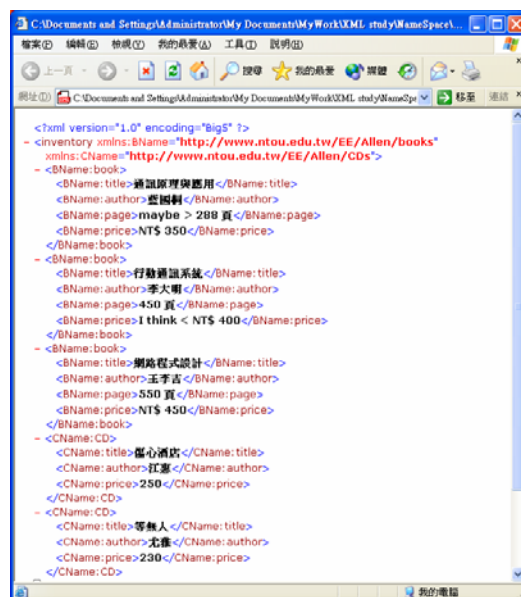
說明七、A parser might consider

<xmlns:BName> to be similar to
{<http://www.ntou.edu.tw/EE/Allen/books>}:BName

說明八、However, remember that the prefixes themselves mean nothing to the XML parser, they get replaced with the URI internally.

Namespace is only used as an identifier.

三、利用 IE 觀看結果：



CAUTION

Because URIs need not be valid, XML namespaces treats them as a string. In particular, comparisons are done character-by-character. According to this definition, the following two URIs are **not identical**, even though they point to the same document.

`http://www.marchal.com`

`http://marchal.com`

Even differences in case are interpreted as different URIs. For XML namespaces, the following two URIs are **different**:

`http://www.marchal.com`

`http://www.MARCHAL.com`

Lecture 2: 使用檔名：myXmlNameSpace0020.xml

一、先 edit 一個 myXmlNameSpace0020.xml 檔。

(目標：用 default namespace 節省 prefix 的使用)

```
<?xml version="1.0" encoding="Big5" ?>
```

```
<inventory
```

```
xmlns="http://www.ntou.edu.tw/EE/Allen/books"
```

```
xmlns:CName="http://www.ntou.edu.tw/EE/Allen/CDs"
```

```
>
```

```
<book >
```

```
<title>通訊原理與應用</title>
```

```
<author>藍國桐</author>
```

```
<page> maybe &gt; 288 頁</page>
```

```
<price>NT$ 350</price>
```

```
</book>
```

```
<book>
```

```
<title>網路程式設計</title>
```

```
<author>王李吉</author>
```

```
<page>550 頁</page>
```

```
<price>NT$ 450</price>
```

```
</book>
```

Default
namespace

```
<CName:CD >
```

```
<CName:title>傷心酒店</CName:title>
```

```
<CName:author>江惠</CName:author>
```

```
<CName:price>250</CName:price>
```

```
</CName:CD>
```

```
<CName:CD >
```

```
<CName:title>等無人</CName:title>
```

```
<CName:author>尤雅</CName:author>
```

```
<CName:price>230</CName:price>
```

```
</CName:CD>
```

```
<CName:CD>
```

```
<CName:title>妹力四射</CName:title>
```

```
<CName:author>江惠妹</CName:author>
```

```
<CName:price>350</CName:price>
```

```
</CName:CD>
```

```
</inventory>
```

說明一、 We declare a default namespace by assigning the namespace name to the reserved *xmlns* attribute.

說明二、 As a result, the book element and all nested elements within it that don't have prefixes belong to the namespace named <http://www.ntou.edu.tw/EE/Allen/books>.

說明三、 You can declare more than one namespace for an element, but only one can be the default.

二、利用 IE 觀看結果：(demo)

Lecture 2-1: 使用檔名：myXmlNameSpace0021.xml

三、利用 myXmlNameSpace0021.xml 檔。

(目標：在 PI 中宣告 NameSpace)

```
<?xml version="1.0" encoding="Big5" ?>
<?xml:namespace
ns="http://www.ntou.edu.tw/EE/Allen/books"
xmlns:CName="http://www.ntou.edu.tw/EE/Allen/CDs"?>

<inventory >

  <book >
    <title>通訊原理與應用</title>
    <author>藍國桐</author>
    <page> maybe &gt; 288 頁</page>
    <price>NT$ 350</price>
  </book>

  <book>
    <title>網路程式設計</title>
    <author>王李吉</author>
    <page>550 頁</page>
    <price>NT$ 450</price>
  </book>

  <CName:CD >
    <CName:title>傷心酒店</CName:title>
    <CName:author>江惠</CName:author>
    <CName:price>250</CName:price>
  </CName:CD>

  <CName:CD >
    <CName:title>等無人</CName:title>
    <CName:author>尤雅</CName:author>
    <CName:price>230</CName:price>
  </CName:CD>

  <CName:CD>
    <CName:title>妹力四射</CName:title>
    <CName:author>江惠妹</CName:author>
    <CName:price>350</CName:price>
  </CName:CD>

</inventory>
```

說明一：

用 IE 來作 well-form check 會認為 這個 XML file 並非 well-formed, 但利用 XML-Spy 卻認為本 XML file 是 well-formed。

說明二：

將 QName 的設定改成和 default NameSpace 相同

xmlns:CName="http://www.ntou.edu.tw/EE/Allen/books" 依然是 well-form, 所以 <price> 和 <CName:price> 是否屬於同一個 NameSpace ??

Lecture 3: 使用檔名：myXmlNamespace0030.xml

一、先 edit 一個 myXmlNamespace0030.xml 檔。

(目標：namespace 不一定要在 document (root) element 上宣告)

```
<?xml version="1.0" encoding="Big5" ?>

<inventory
xmlns="http://www.ntou.edu.tw/EE/Allen/books">

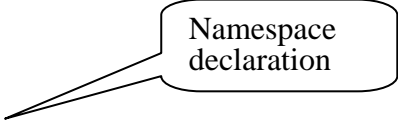
<book >
  <title>通訊原理與應用</title>
  <author>藍國桐</author>
  <page> maybe &gt; 288 頁</page>
  <price>NT$ 350</price>
</book>

<book >
  <title>行動通訊系統</title>
  <author>李大明</author>
  <page>450 頁</page>
  <price>I think &lt; NT$ 400</price>
</book>
```

```
<book>
  <title>網路程式設計</title>
  <author>王李吉</author>
  <page>550 頁</page>
  <price>NT$ 450</price>
</book>

<CName:CD
xmlns:CName="http://www.ntou.edu.tw/EE/Allen/CDs">
  <CName:title>傷心酒店</CName:title>
  <CName:author>江惠</CName:author>
  <CName:price>250</CName:price>
</CName:CD>

</inventory>
```



說明一、We don't have to declare all of our namespace prefixes on the root element; in fact, a namespace prefix can be declared on any element in the document.

說明二、A **qualified name (QName)** is composed of **a prefix and a local part**. The prefix provides the namespace prefix part of the qualified name, and must be associated with a namespace Uniform Resource Identifier (URI).

說明三、Note that these elements that don't use a prefix are no longer called QName, even though they are still universally unique (指的就是 default NameSpace).

二、利用 IE 觀看結果：(demo)

Lecture 4: 使用檔名：myXmlNameSpace0040.xml

一、先 edit 一個 myXmlNameSpace0040.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>

<inventory
xmlns="http://www.ntou.edu.tw/EE/Allen/books">

<book >
  <title>通訊原理與應用</title>
  <author>藍國桐</author>
  <page> maybe &gt; 288 頁</page>
  <price>NT$ 350</price>
</book>

<book >
  <title>行動通訊系統</title>
  <author>李大明</author>
  <page>450 頁</page>
  <price>I think &lt; NT$ 400</price>
```

```
</book>

<book>
  <title>網路程式設計</title>
  <author>王李吉</author>
  <page>550 頁</page>
  <price>NT$ 450</price>
</book>
```

Default Namespace
changed

```
<CD xmlns="http://www.ntou.edu.tw/EE/Allen/CDs">
  <title>傷心酒店</title>
  <author>江惠</author>
  <price>250</price>
</CD>
```

```
</inventory>
```

說明一、We can declare the CD namespace to be the default namespace for **<CD> element** and **its descendents**.

二、利用 IE 觀看結果：(demo)

Lecture 5: 使用檔名：myXmlNameSpace0050.xml

一、先 edit 一個 myXmlNameSpace0050.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>

<inventory
xmlns="http://www.ntou.edu.tw/EE/Allen/books">

<book >
  <title>通訊原理與應用</title>
  <author>藍國桐</author>
  <page> maybe &gt; 288 頁</page>
  <price>NT$ 350</price>
</book>

<book >
  <title>行動通訊系統</title>
  <author>李大明</author>
  <page>450 頁</page>
  <price>I think &lt; NT$ 400</price>
</book>
```

```
<CD xmlns="http://www.ntou.edu.tw/EE/Allen/CDs">
  <title>傷心酒店</title>
  <author>江惠</author>
  <price>250</price>
  購買人： <author xmlns="">藍國桐</author> 先生
</CD>

<book>
  <title>網路程式設計</title>
  <author>王李吉</author>
  <page>550 頁</page>
  <price>NT$ 450</price>
</book>

</inventory>
```

說明一、Sometime not all of the elements belong to a namespace. We can **override** the default namespace within a nested element by assigning a different value to *xmlns* within that element. If we assign an empty string to *xmlns*, all non-prefixed elements within the scope of the assignment are considered not to belong to a namespace.

二、利用 IE 觀看結果：(demo)

Lecture 6: 使用檔名：myXmlNameSpace0060.xml

一、先 edit 一個 myXmlNameSpace0060.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>

<inventory
xmlns="http://www.ntou.edu.tw/EE/Allen/books">

<book >
  <title>通訊原理與應用</title>
  <author id="comm-122">藍國桐</author>
  <page> maybe &gt; 288 頁</page>
  <price>NT$ 350</price>
</book>
```

```
<EBOOK:book
xmlns:EBOOK="http://www.ntou.edu.tw/EE/Allen/ebooks">
  <EBOOK:title>網路程式設計</EBOOK:title>
  <EBOOK:author id="comp-132">王李吉</EBOOK:author>
  <EBOOK:page>550 頁</EBOOK:page>
  <EBOOK:price>NT$ 450</EBOOK:price>
</EBOOK:book>

</inventory>
```

說明一、In fact, attributes usually don't have namespaces. They are just “associated” with the elements to which they belong. (我懷疑這種說法?)

二、再先 edit 一個 myXmlNameSpace0061.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>

<inventory
xmlns="http://www.ntou.edu.tw/EE/Allen/books">

<book >
  <title>通訊原理與應用</title>
  <author id="comm-122">藍國桐</author>
  <page> maybe &gt; 288 頁</page>
  <price>NT$ 350</price>
</book>
```

```
<EBOOK:book
xmlns:EBOOK="http://www.ntou.edu.tw/EE/Allen/ebooks">
  <EBOOK:title>網路程式設計</EBOOK:title>
  <EBOOK:author EBOOK:id="comp-132">王李吉
  </EBOOK:author>
  <EBOOK:page>550 頁</EBOOK:page>
  <EBOOK:price>NT$ 450</EBOOK:price>
</EBOOK:book>

</inventory>
```

說明二、兩者的結果是一樣的。Most applications will treat the two situations identically. (我懷疑?)

三、利用 IE 觀看結果：(demo)

Lecture 7: 使用檔名：myXmlNameSpace0062.xml

一、先 edit 一個 myXmlNameSpace0062.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>
```

```
<inventory xmlns="http://www.ntou.edu.tw/EE/Allen/books">
```

```
  <book xmlns:EBOOK="http://www.ntou.edu.tw/EE/Allen/ebooks"
        EBOOK:id="3" id="#55">
    <title>通訊原理與應用</title>
    <author id="#12">藍國桐</author>
    <EBOOK:author id="#122">王李吉</EBOOK:author>
    <page> maybe &gt; 288 頁</page>
    <price>NT$ 350</price>
  </book>
```

```
</inventory>
```

說明一、Note that the default namespace does not apply to attributes within its scope.

Rather, you always need to designate the namespace of an attribute by explicitly adding a namespace prefix.

說明二、As shown here, you can have two attributes with the same name (事實上，這兩個 attributes 是不同名字的) in a start-tag if you assign a unique namespace to one of them.

二、利用 IE 觀看結果：(demo)

Lecture 7-1: 使用檔名：myXmlNameSpace0063.xml

一、先 edit 一個 myXmlNameSpace0063.xml 檔。

```
<?xml version="1.0" encoding="Big5" ?>  
  
<inventory xmlns="http://www.ntou.edu.tw/EE/Allen/CDs">  
  
  <book xmlns:EBOOK="http://www.ntou.edu.tw/EE/Allen/books"  
    EBOOK:id="3" id="#55">  
    <title>通訊原理與應用</title>  
    <author EBOOK:id="#122" id="#55">藍國桐</author>  
    <EBOOK:author EBOOK:id="#122" id="#55">王李吉  
    </EBOOK:author>  
    <page> maybe &gt; 288 頁</page>  
    <price>NT$ 350</price>  
  </book>
```

說明、default NameSpace 無法 apply 到 attribute，所以上述的 attribute **id="#55"** 是沒有 NameSpace 的。

三、利用 IE 觀看結果：(demo)

二、範例：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<person xmlns="http://sernaferna.com/pers">
  <name id="1">
    <title>Sir</title>
    <first>John</first>
    <middle>Fitzgerald Johansen</middle>
    <last>Doe</last>
  </name>
  <position>Vice President of Marketing</position>
  <résumé>
    <html:html xmlns:html="http://www.w3.org/1999/xhtml">
      <html:head>
        <html:title>Resume of John Doe</html:title>
      </html:head>
      <html:body>
        <html:h1>John Doe</html:h1>
        <html:p html:style="FONT-FAMILY: Arial"> John's a great guy, you know? </html:p>
      </html:body>
    </html:html>
  </résumé>
</person>
```

1. Attributes that are specifically declared to be in a namespace are called **global attributes**.
2. Because we want the *style* attribute to be specifically in the XHTML namespace, we have gone back to using prefixes on our XHTML elements, instead of a default namespace.
3. The *id* attributes that we added is associated with the <name> element, but it doesn't actually have a namespace.
4. Similarly, the *style* attribute is associated with the <p> element, but in this case the attribute is specifically in the XHTML namespace.
5. Application may or may not treat both of these the same, and consider them to be in the same namespace as the elements to which they are attached. All applications will treat the *style* attribute as being in the XHTML namespace, because we have specifically said so, but some will think *id* is in the same namespace as <name>, and some won't.

Lecture 8:

- The namespace mechanism makes XML more modular.
- Also, XML standards for enhancing documents often require you to use specific elements and attributes so that the applications that handle the documents can recognize them.
- The following are examples of such standards:
 1. The XML specification itself includes the reserved *xml:space* and *xml:lang* attribute. The *xml:lang* attribute don't affect the behavior of the XML processor. The processor merely passes the attribute specification on to the application, which can use the value as appropriate.
(demo: c:/ktlan/XMLStudy/ TEST001.xml)
 2. As you'll learn in the future, you create an **XML schema** by using a standard set of elements (for example, **schema, element**) that belong to the namespace named *http://www.w3.org/2001/XMLSchema*.
(demo: 範例一、二)
 3. If you display a document using a CSS (Cascading Style Sheets), you can insert HTML elements into your document by using XML elements that represent the HTML elements and that are prefaced with the *html* namespace prefix. (想在一個 XML document 檔案之中，利用 CSS 的特性插入一個 html 的 tag 來顯示網頁，**html** 這個 prefix 絕不能任意改變，而且不能採用 default namespace 的方式處理。)
(demo: 範例三)
 4. When you create an XSLT style sheet, you use a standard set of elements (for example, **stylesheet, value-of**) that belong to the namespace named *http://www.w3.org/1999/XSL/Transform*.
(demo: 範例四)

範例一、使用檔名： InstanceXML003.xml 。

(folder: c:/ktlan/XMLstudy/Schema/)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- File Name: InstanceXML003.xml -->
<bookname xmlns="http://www.tncc.tnit.edu.tw/ktlan"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://www.tncc.tnit.edu.tw/ktlan SchemaXML003.xsd">
  <code>F8915</code>
  <title>ASP Home Page</title>
  <author>ktlan</author>
</bookname>
```

範例二、使用檔名： SchemaXML003.xsd 。

(folder: c:/ktlan/XMLstudy/Schema/)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- File Name: SchemaXML003.xsd -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.tncc.tnit.edu.tw/ktlan"
            xmlns:target="http://www.tncc.tnit.edu.tw/ktlan"
            elementFormDefault="qualified">
  <xsd:element name="bookname">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="code" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="author" type="xsd:string"/>
      </xsd:sequence>
      <xsd:attribute name="title" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

說明：

```
<xsd:element xsd:name="title" type="xsd:string"/>
```

對屬性 name 加上 NameSpace 會造成 not valid (但仍是 well-form)

範例三、使用檔名： myXmlCSS_0111.xml 。

(**folder:** c:/kltan/XMLstudy/XML_CSS/)

```
<?xml version="1.0" encoding="Big5" ?>  
<!-- my first Xml file -->  
<?xml-stylesheet type="text/css" href="myXmlCSS_0010.css" ?>
```

```
<inventory>  
  <book>  
    <title>通訊原理與應用</title>  
    <author>藍國桐</author>  
    <page>288 頁</page>  
    <price>NT$ 350</price>  
  </book>  
  
  <book>  
    <title>  
      <html:A xmlns:html="http://www.w3c.org/TR/REC-html40/"  
        href="http://www.ntou.edu.tw">  
        網路程式設計  
      </html:A>  
    </title>  
    <author>王李吉</author>  
    <page>550 頁</page>  
    <price>NT$ 450</price>  
  </book>  
</inventory>
```

說明：

```
<?xml-stylesheet type="text/css" href="myXmlCSS_0010.css" ?>
```

將 href="..." 刪掉，所以 xml 將找不到 這個 .css 檔，所以直接顯示 xml 中的資料。但是 <html:A ...> 這個 tag 仍會被 IE 所認同。

範例四、使用檔名： myCD-XSL-0050.xsl 。

(**folder:** c:/kltan/XMLstudy/XSLT/)

```
<?xml version="1.0" encoding="UTF-16"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:template match="/">
    <html>
      <head>
        <title>This is my first xsl pratice!</title>
      </head>
      <body>
        <H2>Hello World!</H2>
        <xsl:value-of select="CD/song" /><BR/>
        <xsl:value-of select="CD/song" /><BR/>
        <xsl:value-of select="CD/song" /><BR/>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

說明：

```
<xsl:value-of xsl:select="..." />
```

對屬性 **select** 加上 NameSpace 反而會造成作 Translate 時的錯誤!

Lecture 9: Homework II:

(students major in CS)

Study the specification of the namespaces from the W3C web site and clarify the concepts behind namespaces. You will give me a report to supplement the subject of the namespace that is not mentioned on the lectures or to present your experimental result about namespace.

(other students)

就 Homework I 中的第一題，CD 可以分成 三類，這三類都有相同的 tags，例如 <single />、<price /> 等等。利用 不同的 Namespace 區分它們。

Reference:

1. The XML Namespaces specification is located at <http://www.w3.org/TR/REC-xml-names/>
2. Describing the syntax for URIs at the IETF (Internet Engineering Task Force) web site, located at <http://www.ietf.org/rfc/rfc2396.txt>
3. Describing the syntax for URNs, located at <http://www.ietf.org/rfc/rfc2141.txt>
4. Describing the syntax for URLs, located at <http://www.ietf.org/rfc/rfc1738.txt>
5. (for URIs, see <http://www.w3.org/Addressing/>)
6. “Using Namespaces in Documents” in Microsoft XML SDK 4.0 help file. Or, <http://msdn.microsoft.com/library>.
7. For more information on these character sets and the specific characters you can use in XML, see the section “2.2 Characters” in the XML specification at <http://www.w3.org/TR/REC-xml>.
8. For a description of the official language identifiers you can assign to *xml:lang*, see the section “2.12 Language Identification” in the XML specification at <http://www.w3.org/TR/REC-xml>.
- 9.