



XML與詮釋資料

報告人：余顯強
世新大學 資訊傳播學系教授
yjc@cc.shu.edu.tw



大綱

■ XML概述

- 文法規則(Well-formed)
- 型別宣告(結構定義)
 - ◆ (Document Type Definition, DTD)
 - ◆ XML Schema
- 其他延伸技術的應用

■ Metadata簡介

Unit. 1

XML概述



何謂標示

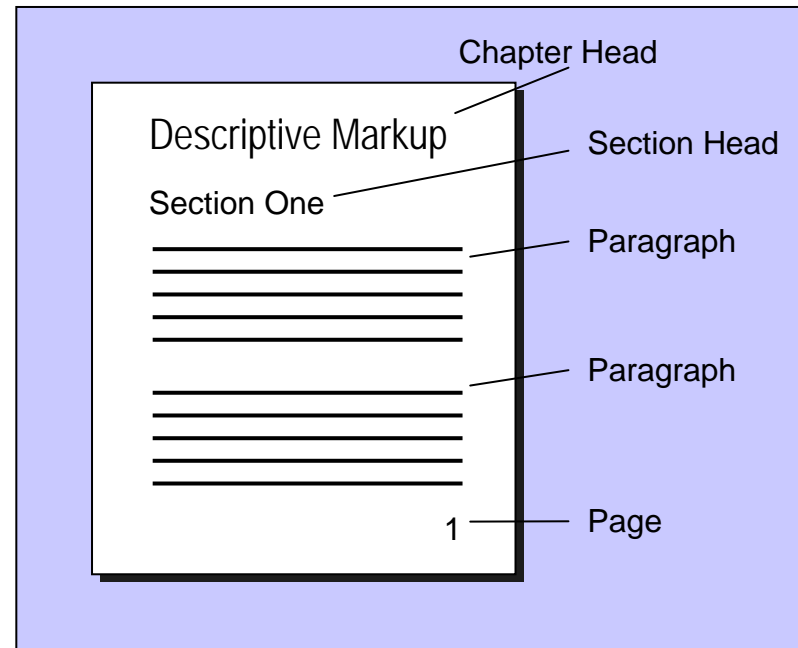
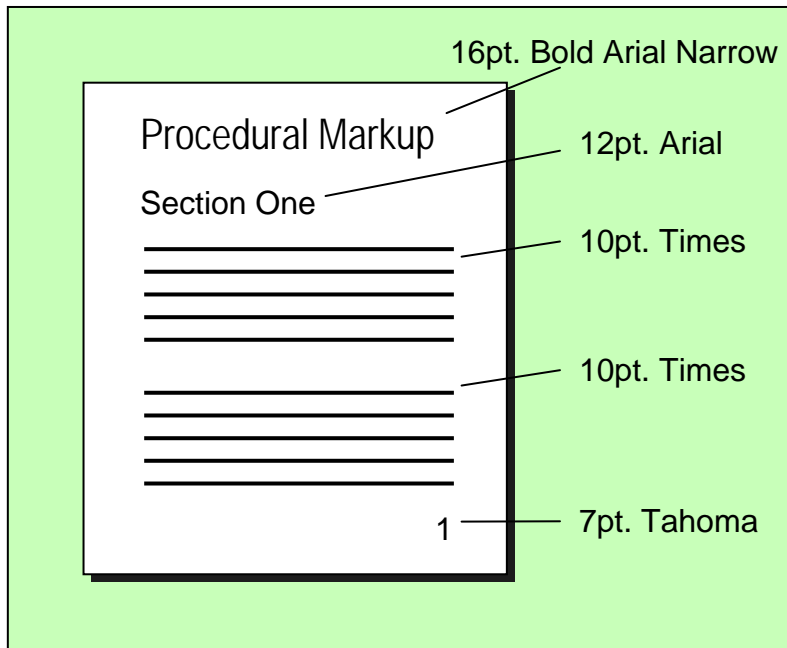
■ 標示: Markup

- 可以視為一個識別符號(code)或標誌(token)
- 指示如何詮釋資料的性質
 - ◆ 所屬欄位、資料型態、顯示格式
- 描述文件中的資料應如何被詮釋

何謂標示

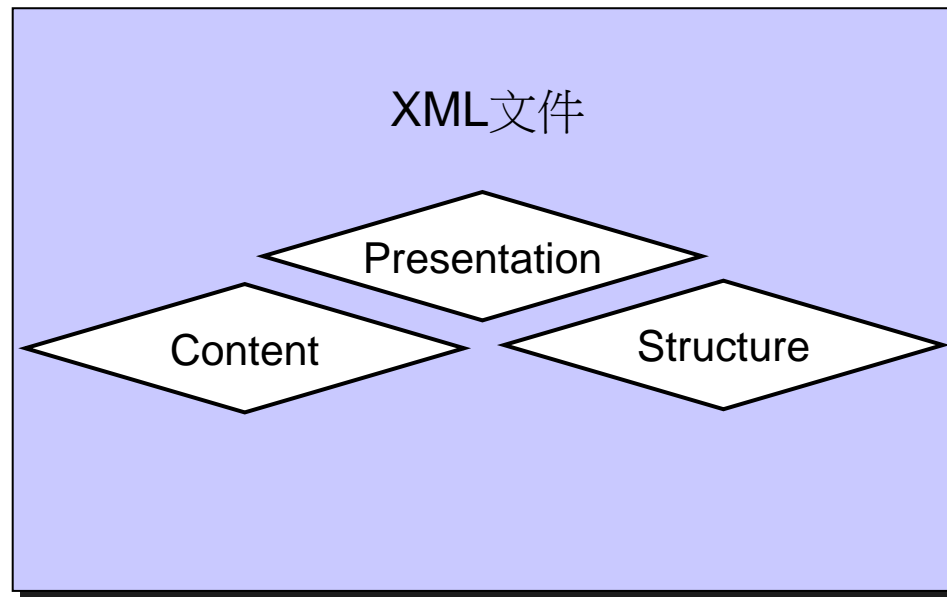
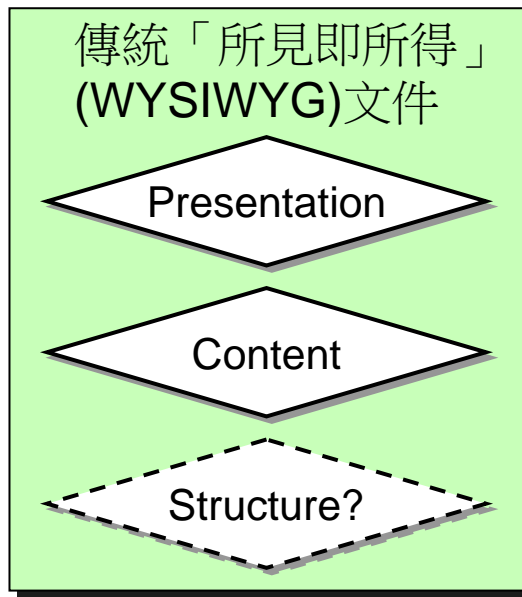
■ 標示: Markup

- 程序式標示 (Procedural Markup)
- 描述式標示 (Descriptive Markup)



描述性標示的特色

- 將文件內容與呈現格式區分開來
- 針對文件的語意結構進行標識



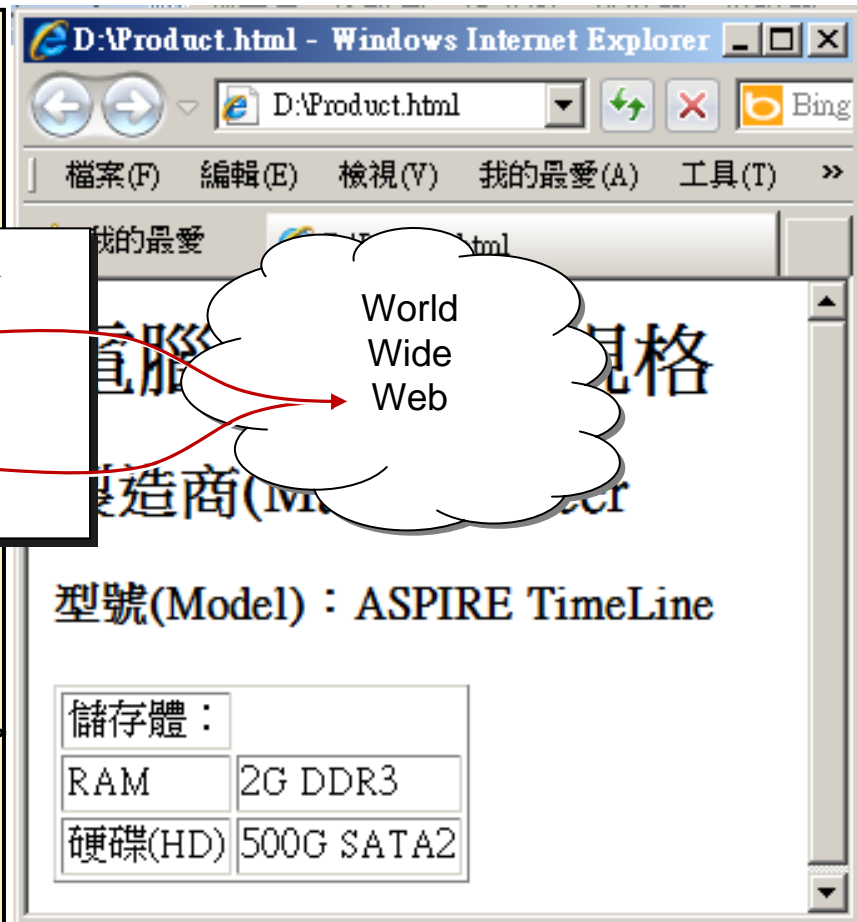
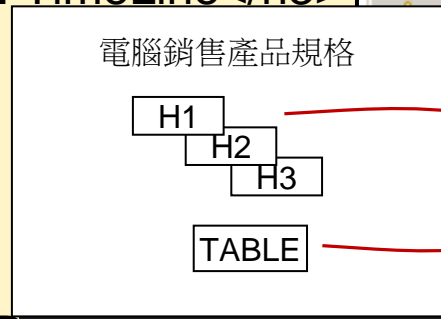
範例：

製造商(Maker)	Acer	
型號(Model)	ASPIRE TimeLine	
儲存體(Storage)	RAM	2G DDR3
	硬碟(HD)	500G SATA2

```

<html>
<h1>電腦銷售產品規格</h1>
<h2>製造商(Maker)：Acer</h2>
<h3>型號(Model)：ASPIRE TimeLine</h3>
<table border=1>
<tr>
<td>儲存體：</td>
</tr>
<tr>
<td>RAM</td><td>2G DDR3</td>
</tr>
<tr>
<td>硬碟(HD)</td><td>500G SATA2</td>
</tr>
</table>
</html>

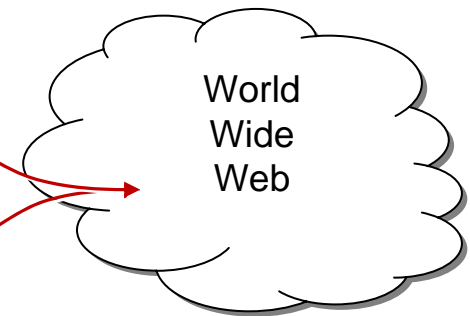
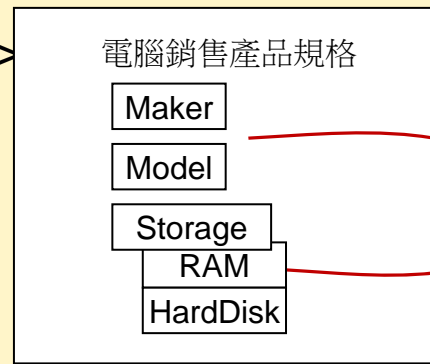
```



範例：

製造商(Maker)	Acer	
型號(Model)	Travel Mate	
儲存體 (Storage)	RAM	256M
	硬碟(HD)	40G

```
<?xml version="1.0" encoding="big5" ?>
<NBforSale>
  <title>電腦銷售產品規格</title>
  <maker>Acer</maker>
  <model>ASPIRE TimeLine</model>
  <storage>
    <RAM unit="GB" type="DDR3">2</RAM>
    <HD unit="GB" Type="SATA2">500</HD>
  </storage>
</NBforSale>
```





三個重要的標示語言

- **SGML**
- **HTML**
- **XML**



1. SGML

- **Standard Generalized Markup Language**
- **1986年 ISO 所制定的標準 – ISO 8879**
- 屬於描述性標示
- 是一種 **Meta-language**



SGML文件的組成

- **宣告(declaration)**
 - 指定文件所使用的字集及特定的選項功能
- **文件型別定義(Document Type Definition, DTD)**
 - 定義文獻所包含的各種元素(elements)
 - 定義元素的內容與屬性(attributes)
 - ...
- **文件實體(Document Instance, DI)**
 - 加上標示處理後的文件



SGML文件實體

```
<!DOCTYPE Email SYSTEM "c:\temp\email.dtd">
```

```
<Email>
```

```
  <From>St.John Yu</From>
```

```
  <Date>2010-1-1 PM 18:30</Date>
```

```
  <To>Shien-chiang</To>
```

```
  <To>Seljuk</To>
```

```
  <Subject>Present XML </Subject>
```

```
  <Body>XML與詮釋資料
```

```
    ...
```

```
    ...
```

```
  </Body>
```

```
</Email>
```



SGML 特色

- 有彈性 (**flexibility**)
 - 能描述任何的資訊結構與任何複雜的文件
- 非專屬性 (**non-proprietary**)
 - 無繳付任何權利的使用費用
- 平台獨立性 (**platform-independence**)
 - 能夠運作於任何作業系統
- 系統獨立性 (**system-independence**)
 - 利於文件的交換與長期保存
- 資訊再利用性 (**re-usability**)
 - 資料易於分解、重組、合併、轉置



SGML發展的限制

- **SGML**應用程式不易開發
- **SGML**文件不易在**Web**上傳佈
- 缺乏廠商的支援



2. HTML

- **Hypertext Markup Language**
- **Tim Berners-Lee於1989.4在CERN的發明**
 - 1989年在CERN內部建立一套資訊網系統
 - 1990年完成了HTTP，電腦可以利用HTTP在Internet上傳輸超連結文件
 - 1990年末，完成了類似瀏覽器的程式，並取名為World Wide Web
 - 1994年5月24日在CERN舉行第一屆WWW研討會，於會中正式將Web定名為World Wide Web
- 是 **SGML** 的應用
- 目前 **Web** 上寫作網頁的標準資料格式
- 簡單易學
 - 具可攜性 (portable)
 - Text + Link + Multimedia

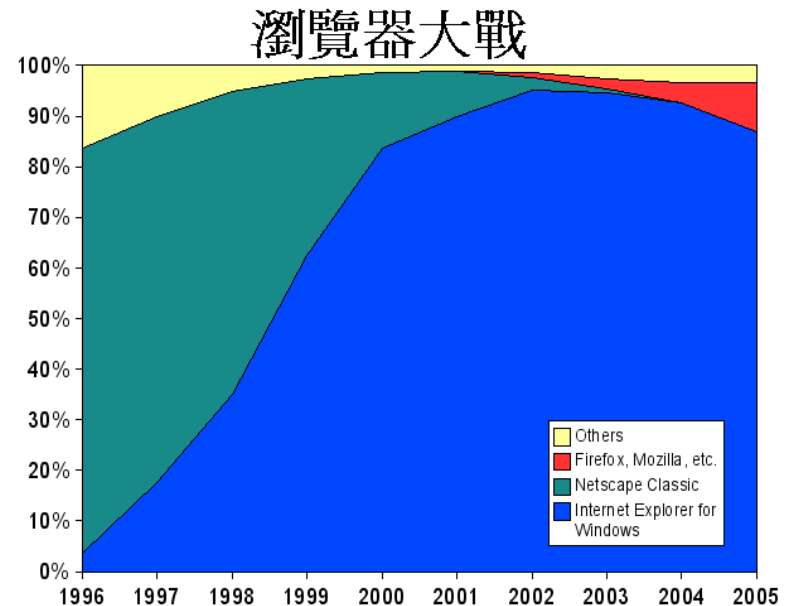


HTML特性

- **HTML DTD** 的設計主要是為了滿足線上顯示的需求
- **HTML**有內建的樣式 (**style**)
- **HTML**引用**SGML**的標示最簡化特徵 (**markup minimization feature**)
- **HTML**沒有採用 **SGML** 的超連結機制

HTML的限制

- 結構上的限制
- 資訊再利用的限制
- 資料交換的限制
- 自動文件處理的限制
- 無法支援較精確的查詢
- 各家廠商推出的 **HTML Extension** 不相容

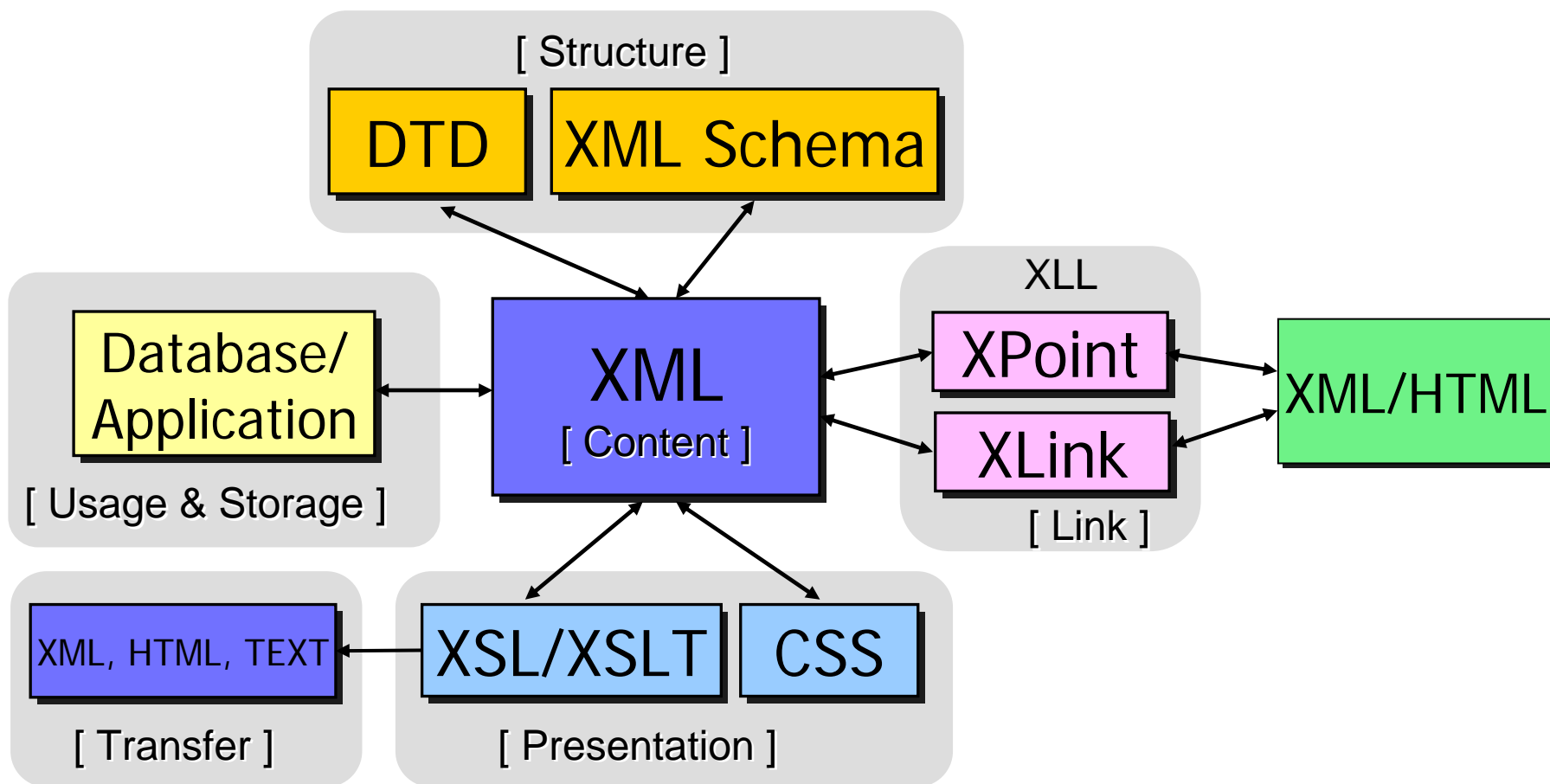




3. XML

- **Extensible Markup Language**
- **W3C Recommendation**
 - 1.0- 1998.02.10
 - 1.1- 2004.02.04
- 大廠支持：**Microsoft**、**Netscape**、**Sun...**
- 取 **SGML**之長，補 **HTML**之短
 - 允許使用者依據需求，自行定義 標籤(tags)
 - 能在 Web 上傳遞與剖析

XML相關技術





技術目的

- **XML**只是用來單純的描述資料
- 單獨瀏覽**XML**文件的內容並沒有任何意義
- 一個完整的**XML**架構需要搭配相關技術，才能將需要的資訊作適當的處理



XML結構

- 文件宣告(Prolog)
 - 定義XML版本、字碼
- 根元素(Root element)
 - 文件整體的根節點
- 子元素(Element)
 - 內容項目
 - ◆ 元素大小寫有別
 - ◆ 單一的元素可以包含任何數目的巢狀階層元素

XML Sample 1
(世界日報資料範例)

XML Sample 2 (CSS)

XML Sample 3 (XSL)



標示類型

- 起始標籤(**start tags**)
 - `<article>`, `<title>`
- 結束標籤(**end tags**)
 - `</article>`, `</title>`
- 空元素(**empty element**)
 - `<graphic/>`
- 根元素(**root element, document element**)
- 父元素和子元素(**parent and child element**)
- 組合內容(**mixed content**)

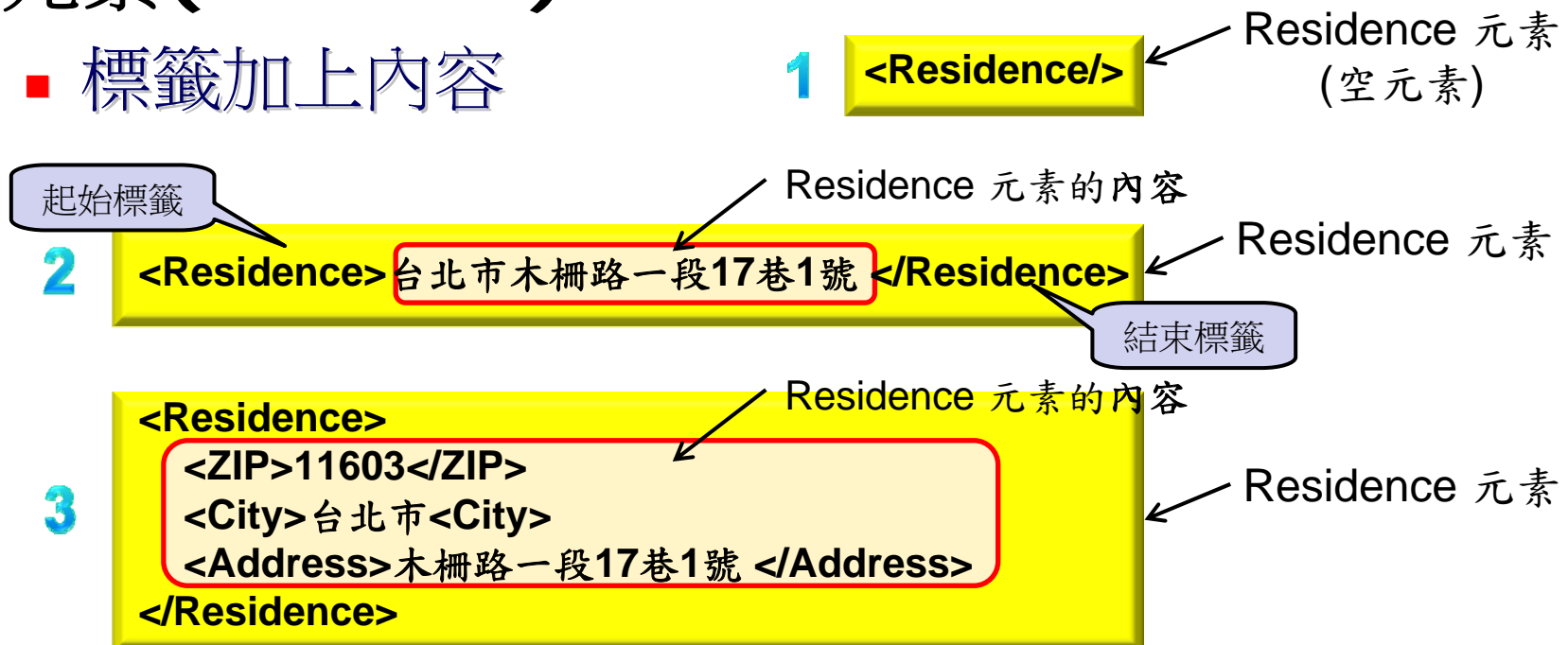
標籤 VS. 元素

■ 標籤(Tag)

- 文件內容的標示符號，分為起始標籤、結束標籤

■ 元素(Element)

- 標籤加上內容



XML文件前言 (Prolog)

- **XML宣告(declaration)**

- *version, encoding, standalone*

- `<?xml version="1.0" encoding="big5" standalone="yes"?>`

- **註解(comments)**

- `<!-- 這是註解 -->`

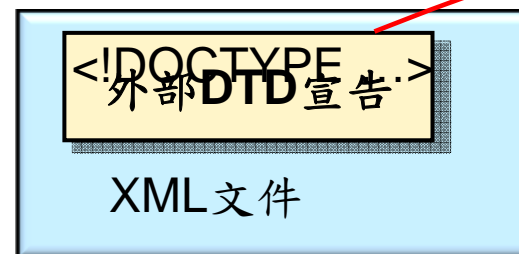
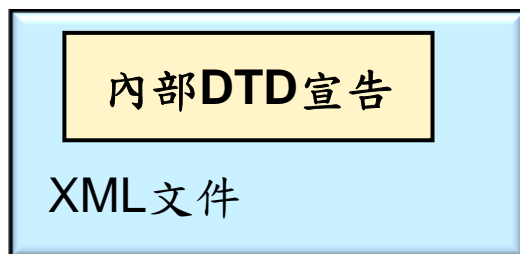
- **處理指示(processing instructions)**

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

- **文件型別宣告(document type declarations)**

- `<!DOCTYPE Booklist SYSTEM "D:\Edit\Present\XML\實作練習\Book.dtd">`

- `<Booklist xmlns="http://www.seljuk.com/myNamespace" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.seljuk.com/myNamespace D:\Edit\Present\XML\實作練習\Book.xsd">`



Well-formed規則

- 恰有一個根元素，而其他的元素必須在此根元素內
- 每個元素必須有起始標籤(**start-tag**)與結束標籤(**end-tag**)
- 所有的標籤必須呈現適當的巢狀 (**nested**) 結構
 - ex: `<I>bold and italicitalic</I>` 是不允許的
- 空元素必須遵守特殊的XML語法
 - ex ``
- 所有的屬性值必須括上單引號或雙引號
 - ex : ``
- 元素內的屬性不能重複
- 標籤與屬性名稱大小寫有分
 - Ex: `<Name>張三</Name>`與`<name>張三</name>` 是不同的元素



如何設計XML內涵

- 檢視**Element**
 - Data or metadata (information about another data element)?
- 檢視**Attribute**
 - Describe how to interpret? Or present data element?
 - Apply to all data elements in its scope?
- 檢視**Tag**
 - Help describe what all data elements in its scope?
- 檢視**Grouping**
 - All members related in the parent nodes describe?
 - Relationship between siblings unambiguous?



範例：

```
<Empty>
  <Name>
    <Last>張</Last>
    <First>三</First>
  </Name>
  <Address>
    <City>台北市</City>
    <Zip>11603</Zip>
    <Street>木柵路一段17巷1號</Street>
  </Address>
  <Phone type="Office">
    <Area_Code>02</Area_Code>
    <No>22368225</No>
  </Phone>
  <Phone type="Mobile">
    <No>0988123456</No>
  </Phone>
</Empty>
```

DTD

(Document Type Definition)



DTD

- 沿用**SGML**的結構定義，其語法與**XML**不同
- 目地在於文件管理，將內容均視為文字類型
- 用法：
 - internal DTD：XML文件內包含DTD的宣告
 - external DTD：XML文件內宣告參照外部DTD檔案
 - mixed DTD：混合上述兩種的方法



DTD 宣告規則

- 在 **XML** 文件宣告

- 標明外部DTD來源：`<!DOCTYPE booklist SYSTEM "d:\book.dtd" >`
- 標明內部DTD定義：`<!DOCTYPE document [.....]>`

- 在**DTD**文件宣告

- 定義**Element** 和 **Attribute**

- ◆ `<!ELEMENT element_name content_spec >`

- element_name 元素名稱
- content_spec 元素內容的類型

- ◆ `<!ATTLIST element_name attribute_name type default >`

- element_name 所屬元素名稱
- attribute_name 屬性的名稱
- type 類型
- default 預設值

- 定義**Entity**

- ◆ `<!ENTITY entity_name content >`

- entity_name 實體名稱
- content 實體內容

DTD 宣告語法

■ 類型

- 元素宣告：!ELEMENT
- 屬性宣告：!ATTLIST
- 實體宣告：!ENTITY

■ 元素關係

- 逗號(,)表示依序出現次序
- 刪節號(|)表示眾欄位選一

■ 元素性質

- | | | | |
|-----|--------|--------|------|
| ■ | 最少1最多1 | -- 必備 | 不可重複 |
| ■ + | 最少1最多* | -- 必備 | 可重複 |
| ■ * | 最少0最多* | -- 非必備 | 可重複 |
| ■ ? | 最少0最多1 | -- 非必備 | 不可重複 |

■ 資料型態

- #PCDATA
- CDATA
- ANY, Empty

DTD 範例

最少1最多1- 必備 不可重複
+最少1最多*- 必備 可重複
最少0最多- 非必備 可重複
?最少0最多1- 非必備 不可重複

- `<?xml version="1.0" encoding="UTF-8" ?>`
- `<!ELEMENT booklist (#PCDATA,book+)>`
- `<!ELEMENT book (title, authorlist, pub_date?)>`
- `<!ELEMENT title (#PCDATA)>`
- `<!ELEMENT authorlist (author+)>`
- `<!ELEMENT author (#PCDATA)>`
- `<!ELEMENT pub_date (#PCDATA)>`
- `<!ATTLIST book code CDATA #REQUIRED>`
- `<!ATTLIST book price CDATA #REQUIRED>`
- `<!ENTITY book "Shih-hsin University">`

實體宣告(不屬於任何元素)

book元素的屬性

符合定義規範的XML

- `<?xml version="1.0" encoding="Big5"?>`
- `<!DOCTYPE booklist SYSTEM " d:\book.dtd " >` — 外部型別定義檔案位置
- `<!--XML 標準與技術說明-->`
- `<booklist>` — 根元素標籤
- `<book code="NE20118" price="390">`
- `<title>XML理論與實作導引</title>`
- `<authorlist>`
- `<author>余顯強</author>`
- `</authorlist>`
- `</book>`
- `<book code="PG20030" price="590">`
- `<title>Delphi4系統開發與應用</title>`
- `<authorlist>`
- `<author>余顯強</author>`
- `</authorlist>`
- `</book>`
- `</booklist>`

內部 DTD 範例

```
<?xml version="1.0" encoding="Big5"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product, id, role)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT role (#PCDATA)>
]>
<document>
  <employee>
    <name>
      <lastname>張</lastname>
      <firstname>山峰</firstname>
    </name>
    <hiredate>3/1/2005</hiredate>
    <projects>
      <project>
        <product>數位典藏專案計畫</product>
        <id>NSC 94-2422-H-128 -001</id>
        <role>助理</role>
      </project>
      <project>
        <product>世界日報網頁建置計畫</product>
        <id>SHU-001</id>
        <role>專任助理</role>
      </project>
    </projects>
  </employee>
</document>
```

外部DTD範例

```
<?xml version="1.0" encoding="Big5"?>
<!DOCTYPE document SYSTEM "D:\Project.dtd">
<document>
  <employee>
    <name>
      <lastname>張</lastname>
      <firstname>山峰</firstname>
    </name>
    <hiredate>3/1/2005</hiredate>
    <projects>
      <project>
        <product>數位典藏專案計畫</product>
        <NSC 94-2422-H-128 -001</id>
        <role>助理</role>
      </project>
      <project>
        <product>世界日報網頁建置計畫</product>
        <id>SHU-001</id>
        <role>專任助理</role>
      </project>
    </projects>
  </employee>
</document>
```

D:\Project.dtd

```
<?xml version="1.0" encoding="Big5"?>
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product, id, role)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT role (#PCDATA)>
```

XML Schema



DTD的限制

- 無法指定元素內容或屬性資料型態
 - Ex. 只有PCDATA, CDATA宣告，不利資訊傳播與處理
 - Ex. `<SNO>123</SNO>`是整數還是字串?
- 不易描述元素出現次數
 - Ex. 數集關係只具備0,1和「多個」的條件
- **DTD並未遵守XML的語法規則**
 - Ex. `<!ELEMENT book (title, authorlist, pub_date?)>`
 - Ex. `<!ATTLIST book code CDATA #REQUIRED>`
- 無法使用名稱空間(**namespace**)
 - 無法合併多種文件宣告
- 缺乏元素與屬性的群組命名設定，無法重複使用先前宣告之元素或屬性






XML Schema

- **XML Schema Definition (XSD)**
- **W3C**依據**XML 1.0**所制定的綱要定義語言
- 最早由**Microsoft**提出
- **2001.05.02**正式成為**W3C**的建議書

- 目的：
 - 解決**DTD**不符資訊傳播與處理的問題
 - 增加文件型別定義的彈性與功能

Schema 建議書內涵

- 整個W3C Schema標準的說明文件一共分為三個部分
 - 入門指引 (Primer) 
 - 架構(Structure) 
 - ◆ 描述如何以XML Schema的句法來定義及規範XML文件架構
 - 資料類型(Datatypes) 
 - ◆ 提供一套定義XML資料類型的機制



XML Schema 的特性

- 本身就是一份**well-formed**的 XML 文件
- 支援更多XML元素的資料型態XML
- **Schema** 支援名稱空間(Namespace)



指定文件型別

- 當XML文件要參考一個DTD時：

```
<!DOCTYPE booklist SYSTEM "D:\Edit\Present\XML\book.dtd">
```

- 當XML文件要參考一個XML Schema時：

```
<booklist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:noNamespaceSchemaLocation="D:\Edit\Present\XML\book.xsd">  
    ...  
</booklist>
```

DTD與XML Schema應用比較

XML文件範例(片斷)

```
<InvoiceNo>123456789</InvoiceNo>  
<ProductID>J123456</ProductID>
```

以DTD描述的类型別定義:

```
<!ELEMENT InvoiceNo (#PCDATA)>  
<!ELEMENT ProductID (#PCDATA)>
```

以XML Schema描述的类型別定義:

```
<element name='InvoiceNo' type='positive-integer'/>  
<element name='ProductID' type='ProductCode'/>  
<simpleType name='ProductCode' base='string'>  
  <pattern value='[A-Z]{1}d{6}'/>  
</simpleType>
```



名稱空間

- 允許在一個XML文件使用多組XML標籤
- 使得引用或製作的 元素、屬性... 等標記定義能更清楚與避免混淆

- **Ex.**

`xmlns:xs="http://www.w3.org/2001/XMLSchema"`

`xmlns:aa="http://www.seljuk.com.tw/ns/books"`

`<aa:element name="title">`

←非W3C XML Schema元素宣告

`<xs:element name="title">`

←是W3C XML Schema元素宣告



名稱空間使用原則

- 名稱空間會指定一個相對應的字首，而不用冗長的名稱
- 一個XML的名稱空間是由**URI**來辨識，代表一群名稱的集合，而這些名稱是用於XML文件的元素和屬性
- 名稱空間會以合格名稱(**qualified name**)方式來表示
 - 合格名稱=字首：區域名稱
- 一個名稱空間的可見範圍為包含該名稱空間的元素與該元素內所屬的子元素



名稱空間使用範例

```
<?xml version="1.0"?>
<bk:list xmlns:bk="www.seljuk.com.tw/ns/books"
          xmlns:au="www.seljuk.com.tw/ns/creator" >
  <bk:title>XML理論與實作導引</bk:title>
  <au:author>余顯強 譯</au:author>
</bk:list>
```

亦可省略命名空間的字首，則該元素及元素內所屬的子元素均會自動被使用

```
<?xml version="1.0"?>
<list xmlns="www.seljuk.com.tw/ns/books"
       xmlns:au="www.seljuk.com.tw/ns/creator" >
  <title>XML理論與實作導引</title>
  <au:author>余顯強 譯</au:author>
</list>
```



註解說明

- XML Schema提供三個註解的元素
- 使用**annotation**元素
 - 子元素documentation
 - ◆ 建議使用xml:lang屬性標明使用訊息的語文
 - 子元素appInfo
 - ◆ 用於註記使用的工具、樣式和其他應用的說明

```
<xsd:annotation>  
  <xsd:documentation xml:lang="en">  
    this is my demo example.  
    Copyright 2009 Shien-chiang Yu. All rights reserved.  
  </xsd:documentation>  
</xsd:annotation>
```



資料型態

- 內定(**Build-in**)型態
 - 原生(Primitive)
 - 衍生(Derived)
- 自訂(**User-derived**)型態
 - 簡單(Simple)
 - 複雜(Complex)

內建資料型態

▼ 整數型別

byte
short
int
long
unsignedByte
unsignedShort
unsignedInt
unsignedLong
integer
positiveInteger
nonPositiveInteger
negativeInteger
nonNegativeInteger

▼ 浮點型別

float
double
decimal

▼ 布林型別

boolean

▼ 二進位型別

hexBinary
base64Binary

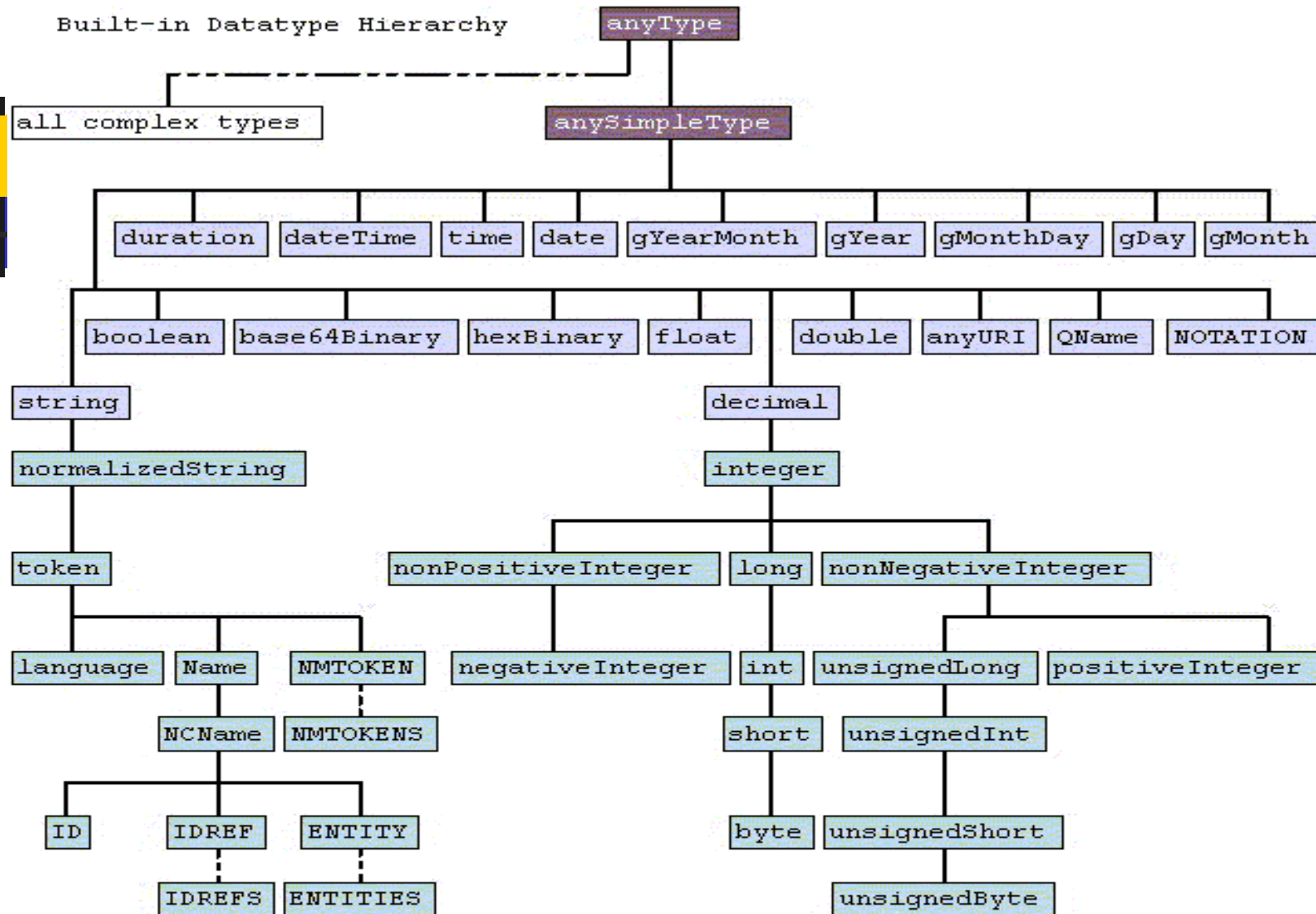
▼ 字串型別

String
normalizedString
Token

▼ 時間型別

dateTime
date
time
duration
gYear
gMonth
gDay
gYearMonth
gMonthDay

Built-in Datatype Hierarchy



- ur types
- built-in primitive types
- built-in derived types
- complex types

- derived by restriction
- derived by list
- derived by extension or restriction



元素定義

- 簡單型態
 - 以內定的資料型態為基礎來定義元素
- 複雜型態
 - 元素擁有屬性
 - 元素包含下一層元素

簡單資料型態之元素定義

Simple Type



簡單型態

■ 單一元素

- 僅包含內容，不包含其他元素或屬性

- ◆ Ex. 定義一個包含string型別的元素title，與包含decimal型態的元素price

```
<element name="title" type="string" />  
<element name="price" type="decimal"/>
```

- ◆ Ex. 指定元素預設值

```
<element name="currency" type="string" default="NT"/>
```

- ◆ Ex. 指定元素固定值

```
<element name="currency" type="string" fixed="NT"/>
```



簡單資料型態

- 透過內建資料型態產生衍生資料型態
- 透過**simpleType**和下列元素產生
 - **restriction**
 - ◆ 限制數值的範圍/型態
 - **union**
 - ◆ 一個衍生型態包含數種型態
 - **list**
 - ◆ 定義一個包含序列值的型態
 - **enumeration**
 - ◆ 定義列舉型態
 - **pattern**
 - ◆ 以regular expression為基礎定義樣板型態



簡單資料型態 - restriction

- **Ex.** 設定學生成績內容的範圍由**0**至**100**

```
<xsd:element name="score" type="scoreRange" />
<xsd:simpleType name="scoreRange">
  <xsd:restriction base="xsd:float">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="100"/>
  </xsd:restriction>
</xsd:simpleType>
```

簡單資料型態 - union

- **Ex.** 定義一個可以輸入字串或數值的尺寸元素

```
<xsd:element name='size'>
  <xsd:simpleType>
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base='xsd:integer'/>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base='xsd:string'/>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
</xsd:element>
```

- 產生的XML文件內容可以是：

```
<size>1</size>
<size>large</size>
```



簡單資料型態 - list

- **Ex.** 定義一個包含多個顏色的元素

```
<xsd:element name="colors" type="ColorList" />
<xsd:simpleType name="ColorList">
  <xsd:list itemType="xsd:string"/>
</xsd:simpleType>
```

- 產生的**XML**文件內容可以是：

```
<colors>red blue green</colors>
```




簡單資料型態 - enumeration

- **Ex.** 假設顏色元素只能使用的值為 **red, blue** 或 **green**

```
<xsd:element name="colors" type="ColorList" />
<xsd:simpleType name="ColorList">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="red"/>
    <xsd:enumeration value="blue"/>
    <xsd:enumeration value="green"/>
  </xsd:restriction>
</xsd:simpleType>
```

簡單資料型態 - pattern

- **Ex.** 定義一個長度**32**，僅允許英文字母開頭，其餘部分則可以是英文、數字或是底線的元素

```
<xsd:simpleType name="editType">  
  <xsd:restriction base="xsd:string">  
    <xsd:maxLength value="32"/>  
    <xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]*"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

XSD Sample

複雜資料型態之元素定義

Complex Type

基本架構

```
<xsd:element name="element_name">
```

```
  <xsd:complexType>
```

```
    <xsd:sequence>
```

```
      ... ..
```

```
    </xsd:sequence>
```

```
    <xsd:attribute name="attr_name" type="data_type"/>
```

```
    ... ..
```

```
  </xsd:complexType>
```

```
</xsd:element>
```

Sequence: 元素循序出現

Choice: 元素任意出現

All: 元素可任意出現，但各元素限定
minOccur=0或maxOccur=1



複雜型態

- 元素擁有屬性

```
<xsd:element name="name" >  
  <xsd:complexType>  
    <xsd:attribute name="gender" type="xsd:string" >  
    <xsd:attribute name="age" type="xsd:positiveInteger"/>  
  </xsd:complexType>  
</xsd:element>
```



複雜型態

- 元素包含下一層元素

XML文件範例(片斷)

```
<Book>  
  <Title>XML理論與實作導引</Title>  
  <Author>余顯強 譯</Author>  
</Book>
```

以XML Schema描述的型別定義:

```
<element name='Book' type='BookType'/>  
<complexType name='BookType'>  
  <element name='Title' type='string'/>  
  <element name='Author' type='string'/>  
</complexType>
```



複雜型態

- 使用「全域簡單型態」定義之複雜型態
 - 自定資料型態

```
<element name='Title' type='string'/>  
<element name='Author' type='string'/>  
<element name='Book' type='BookType'/>  
<complexType name='BookType'>  
  <element ref='Title'/>  
  <element ref='Author'/>  
</complexType>
```



複雜型態

- 使用「全域簡單型態」定義之複雜型態
 - 將BookType型態隱藏

```
<element name='Title' type='string'/>
<element name='Author' type='string'/>
<element name='Book'>
  <complexType>
    <element ref='Title'/>
    <element ref='Author'/>
  </complexType>
</element>
```




定義元素出現次數

- 使用**minOccurs**和**maxOccurs**屬性

```
<element name='Title' type='string'/>
<element name='Author' type='string'/>
<element name='Book'>
  <complexType>
    <element ref='Title' minOccurs='0'/>
    <element ref='Author' minOccurs='1' maxOccurs='2'/>
  </complexType>
</element>
```

可省略



出現次數定義

- **minOccurs**: 最小出現次數
- **maxOccurs**: 最多出現次數
 - 預設為1
 - ◆ 若未設定minOccurs表示必備
 - ◆ 若未設定maxOccurs表示不可重複
 - ◆ 若兩者均未設定，表示該欄位必備且只可出現一次
 - unbounded表示「無限」

屬性定義

Attribute



屬性定義

- 簡單資料型態不包含屬性
- 包含屬性的資料必須為複雜資料型態
- 屬性本身必須為簡單資料型態

屬性範例

- ◆ 定義一個名為name的元素，包含gender和age兩個屬性

```
<xsd:element name="name" >  
  <xsd:complexType>  
    <xsd:attribute name="gender" type="xsd:string" use="required"/>  
    <xsd:attribute name="age" type="xsd:positiveInteger"/>  
  </xsd:complexType>  
</xsd:element>
```

use 設定值：
optional
prohibited
required

- ◆ Ex. 指定元素預設值

```
<xsd:attribute name="currency" type="xsd:string" default="NT" >
```

- ◆ Ex. 指定元素固定值

```
<xsd:attribute name="currency" type="xsd:string" fixed="NT" >
```



屬性定義注意事項

- 同一個元素內定義多個屬性時，需避免
 - 具有相同名稱
 - 不同的命名空間，但參考到相同的URI

```
<Book xmlns:n1="http://www.seljuk.com.tw"
      xmlns:n2="http://www.seljuk.com.tw">
  <Author1 year="92" year="2003">... ..</Author>
  <Author2 n1:year="92" n2:year="2003">... ..</Author>
</Book>
```

✘ Author1有兩個同名的屬性

✘ Author2雖有不同名稱空間，但都參考同一個URI

XML編輯工具



XML IDE 工具

- **Integrated Development Environment**
 - 依據相關規範所開發出來的工具，可協助系統開發人員方便利用
 - 可有效的協助我們節省編輯文件時間與驗證文件合法性，另外可產生及製作相關文件
 - 選用原則應以標準規範之相容性高低考量

工具介紹-Schema Checkers

- 線上schema驗證服務
- <http://www.w3.org/2001/03/webdata/xsv>

XML Schema (REC (20010502) version, as amended) Checking Service - Windows Internet Explorer

<http://www.w3.org/2001/03/webdata/xsv>

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

XML Schema (REC (20010502) version, as amended) ...

W3C XML LG

Validator for XML Schema REC (20010502) version, as amended

XSV version: XSV 3.1-1 of 2007/12/11 16:20:05

NOTICE: This is a service for a W3C [approved recommendation](#). This version is for schema documents with the namespace URI <http://www.w3.org/2001/XMLSchema> and is being actively developed: see [XSV for XML Schema 20000922 version](#) for the no longer maintained previous version, for schema documents with the namespace URI <http://www.w3.org/2000/10/XMLSchema>, and [XSV for XML Schema 200004007 version](#) for the no longer maintained even earlier version, for schema documents with the namespace URI <http://www.w3.org/1999/XMLSchema>.

Use this form for checking a schema which is accessible via the Web, and/or schema-validating an instance with a schema of your own.

Address(es):

[Show warnings](#) [Keep Going](#) [Check as complete schema](#)

Default output is now text/xml with an XSLT stylesheet. Select fallbacks for browsers which don't support <http://www.w3.org/1999/XSL/Transform> stylesheets:

- text/xml + official XSLT (suits IE5 and greater, Mozilla)
- text/xml + early MS XSL support (suits vanilla IE5)
- text/html (styled server-side: suits Netscape, older IE)
- xml, but labelled text/plain (works for any browser, but hard to read)

Get Results

輸入你的XML Schema
檔案所在位置

選擇輸出方式



工具介紹-Authoring Tools

- **XML Schema (XSD)**可使用任何**XML**編輯器
 - 大部分XML編輯器並不感知(aware) XSD的綱要，而僅能處理基本的文法錯誤檢查
 - 主要用來檢查已建立之XSD是否符合XML語法
- 具備**XSD**編輯與驗證的工具
 - Turbo XML (TIBCO)
 - ◆ http://www.tibco.com/software/business_integration/turboxml.jsp
 - XML Spy (Altova)
 - ◆ <http://www.altova.com/xml-editor/>
 - 免費工具: Xray



XML Spy 簡介

- 至**2010**年止，最新版本為**v.2010**
 - 3.5版本尚未完全支援XSD名稱空間
 - 4.0版本之後完全支援W3C XML Schema建議書
- 提供**XML**相關延伸技術文件的編輯功能



XSD編輯練習

■ 1. 宣告 Prolog

```
<?xml verison="1.0" encoding="UTF-8"?>
```

■ 2. 宣告Schema根元素

- 其中宣告一個名稱空間作為此XSD全域之名稱空間

```
<xsd:schema xmlns:xsd:="http://www.w3.org/2001/XMLSchema">
```

■ 3. 宣告註解

```
<xsd:annotation>  
  <xsd:documentation>  
    This is a sample XML Schema.  
  </xsd:documentation>  
</xsd:annotation>
```



XSD編輯練習

■ 4. 定義文件主要結構之元素

```
<xsd:element name="Book" >
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Title" ref="Title"/>
      <xsd:element name="AuthorList" ref="AuthorList"/>
      <xsd:element name="Publisher" ref="Publisher"/>
    </xsd:sequence>
    <xsd:attribute name="code" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```



XSD編輯練習

■ 5. 定義此XSD使用之基本元素

```
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="AuthorList">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Author" ref="Author"
        minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
```

■ 6. 結尾宣告

```
</xsd:schema>
```



XML文件與XSD連結

```
<?xml version="1.0" encoding="big5"?>
<Book code="NE20118">
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="D:\Edit\Present\XML\book.xsd">
    <Title>XML理論與實作導引</Title>
    <AuthorList>
      <Author>Simon St. Laurent</Author>
      <Author>余顯強 譯</Author>
    </AuthorList>
    <Publisher>DrMaster</Publisher>
  </Book>
```




XML 應用

- **Presentation**
 - CSS
 - XSL
 - HTML island
- **Parser Interface**
 - DOM
 - SAX
- **XML-RPC**
 - SOAP
- **Metadata**
- **Description of resource**
 - RDF
- **XML Database**



XML+文件樣式

- XML+CSS
- XML+XSL

XML + CSS

XML + XSL

XML data islands

- HTML網頁內可包含XML文件資料
- 使用<xml>標籤，定義所要包含的XML資料
 - 直接將XML資料獨立在另一個XML文件中
 - 將XML資料包含在同一個HTML檔案中

```
<xml id="id" src="xml檔案位址&名稱">..  
</xml>
```

XML

HTML

```
<xml id="id">  
  xml資料  
</xml>
```

HTML



DOM

- **Document Object Model**
- 透過 **tree mode** 剖析 XML
 - 從 **root element** 開始依序往下將文件內的 **element, attribute, entity, content** 拆成一個個獨立的節點
- **1998 Oct. 01-version 1**
- **2000 Nov. 13-version 2**

DOM 1

DOM parser



SAX

- **Simple API for XML**
- **透過event driven mode剖析XML**
 - 解讀到element, attribute, character 時，就會產生事件，而應用程式便透過event handler來提取這些事件。

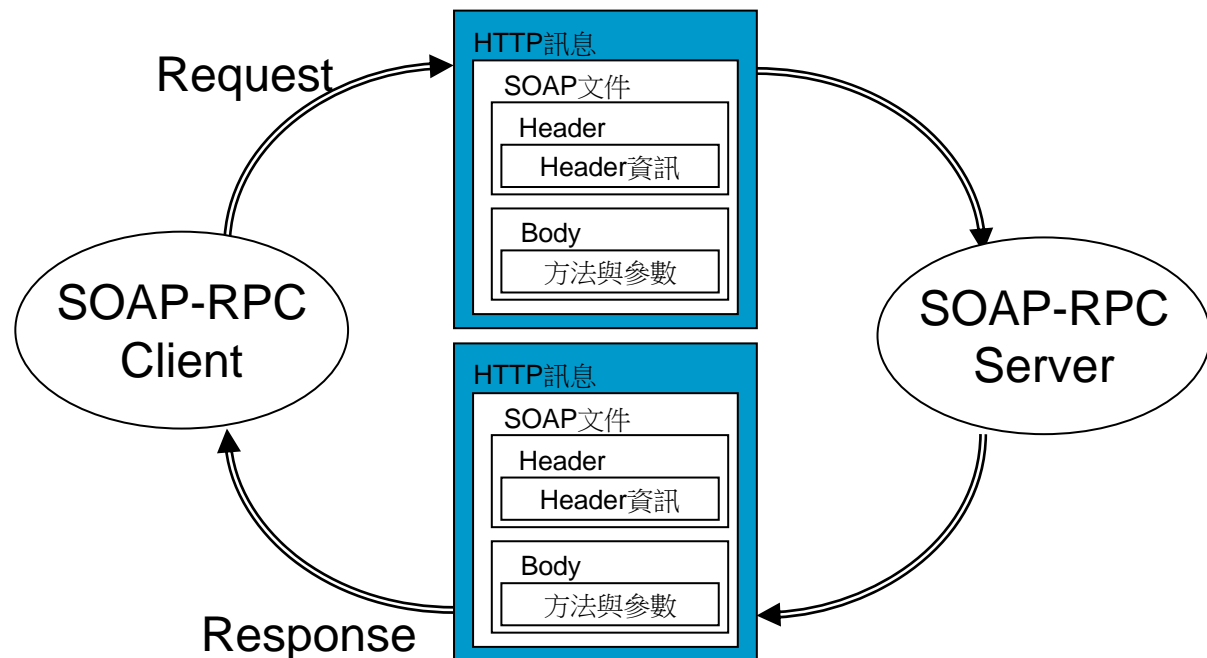


Web Service

- 一組封裝完善(**self-contained**)的應用程式
- 分散式環境的公開服務機制
- 透過**RPC**呼叫**Web Service**介面
- 支援**loose couple**的**client/service**架構
- 支援不同語言來定義服務介面

SOAP

- Simple Object Access Protocol
- 分散式環境中資訊交換的輕量協定





RDF

- **Resource Description Framework**
- **W3C Recommendation**
 - 22-February-1999
- 制定目的
 - 為Metadata在Web上的各種應用提供一個基礎建設
 - 提供應用程式能夠在Web上交換Metadata
- **W3C建議以XML作為RDF資料模式的標準語法，但非強制性**



RDF資料模式

■ 資源(Resource)

- 以RDF表示法描述的物件均稱為資源
 - ◆ 例如：網站、網頁、數位內容等
- 所有資源都能以一個資源識別號(URI)標示

■ 屬性(Property)

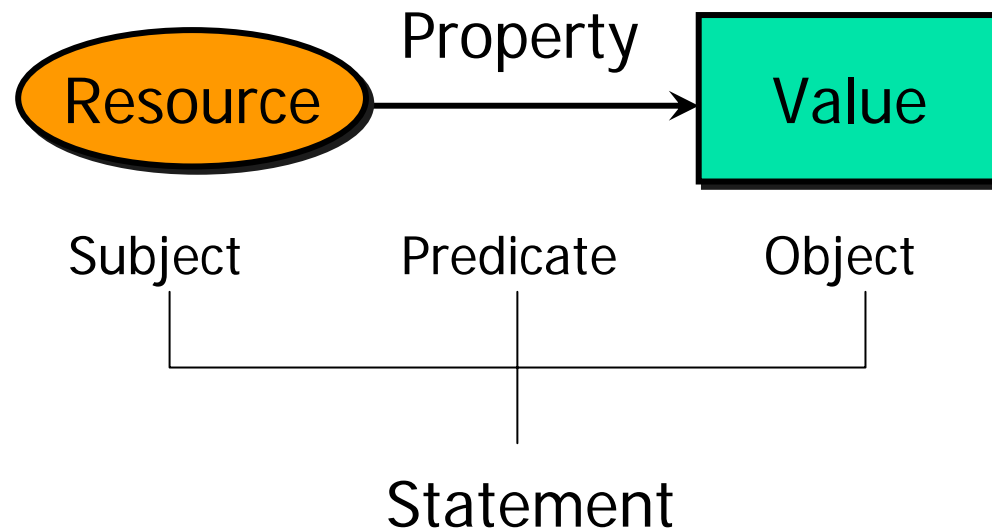
- 描述資源的特性或特徵
 - ◆ Property name
 - ◆ Property value

■ 陳述(Statement)

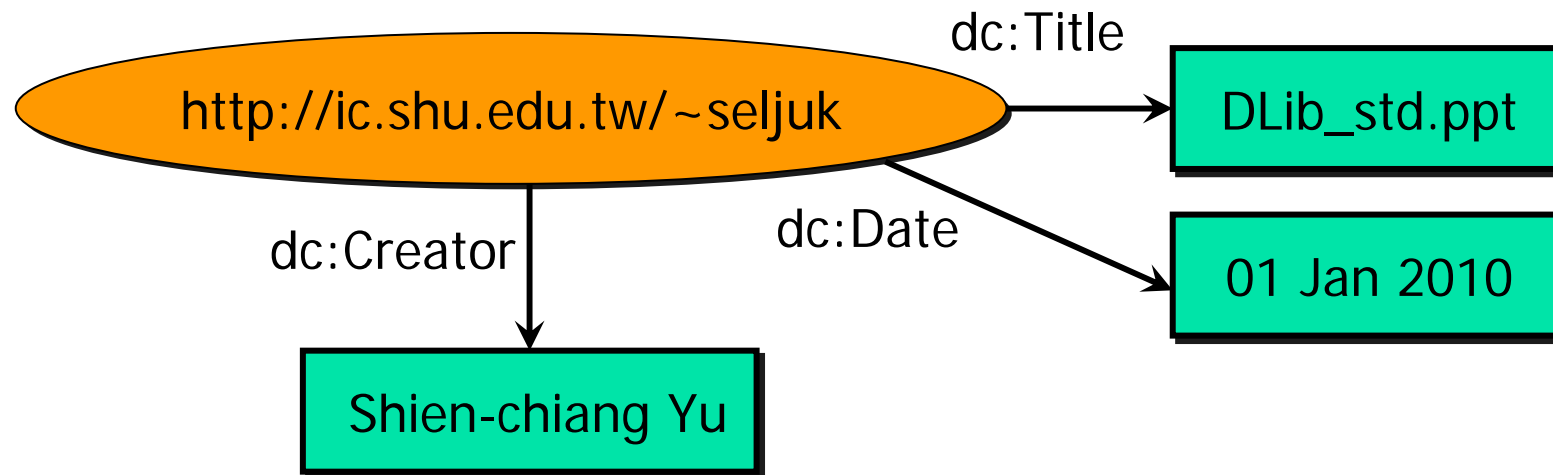
- 以一項property name與property value來描述特定的資源

RDF基本資料模型

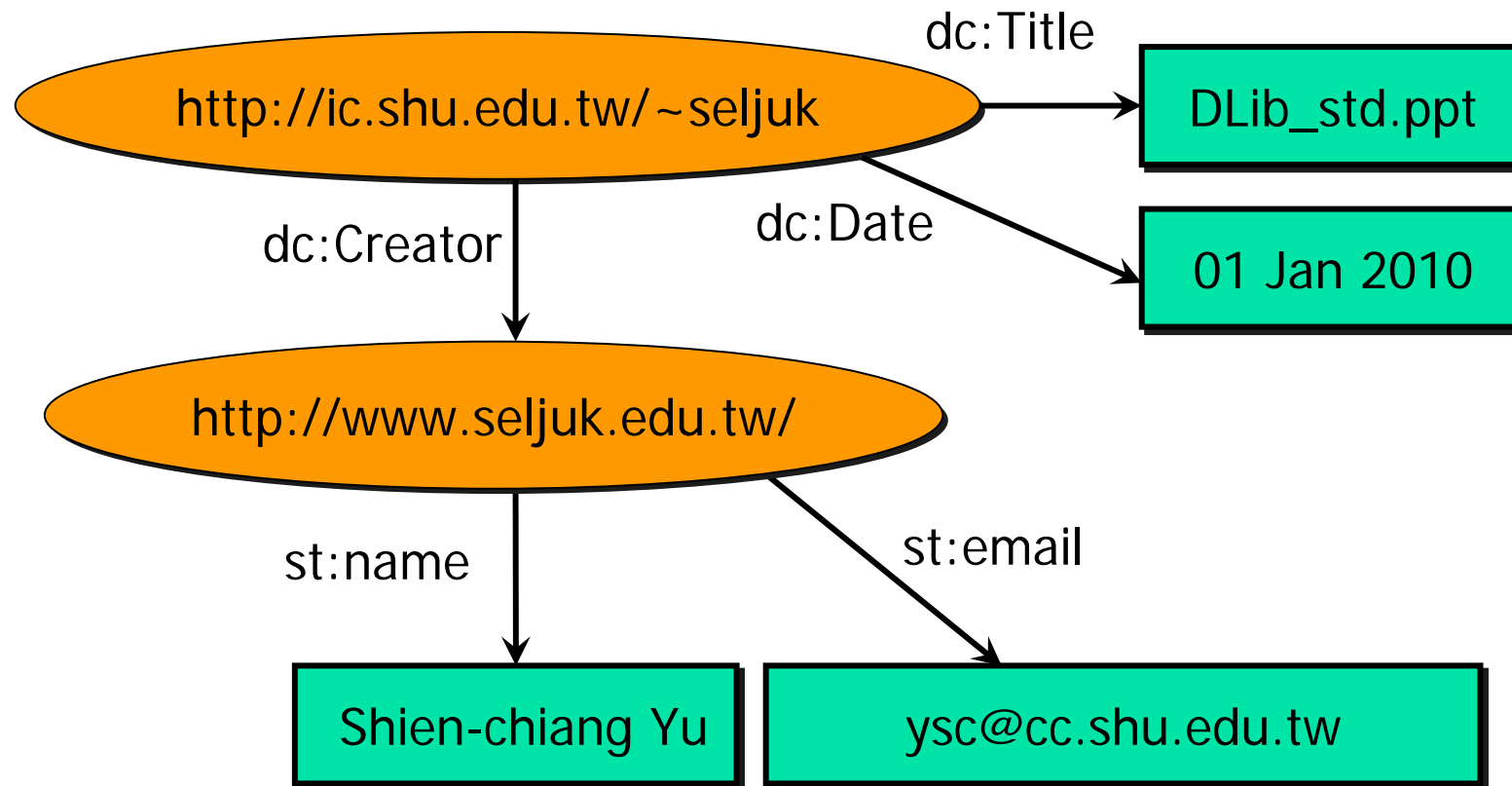
- 主詞：資源
- 述詞：屬性
- 受詞：屬性值



RDF模式範例(2)



RDF模式範例(2)





RDF範例語法

- `<?xml version="1.0">`
- `<rdf:RDF`
- `xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"`
- `xmlns:dc= "http://purl.org/metadata/dublin_core#"`
- `xmlns:st= "http://mycorp.com/schemas/my-schema#" >`
- `<rdf:Description about= "http://ic.shu.edu.tw/~seljuk" >`
- `<dc:Creator resource= "http://www.seljuk.edu.tw/" >`
- `<dc>Title>DLib_std.ppt</dc>Title>`
- `<dc>Date>01 Jan 2010</dc>Date>`
- `</rdf:Description>`
- `<rdf:Description about= "http://www.seljuk.edu.tw/" >`
- `<st:name>Shien-chiang Yu</st:name>`
- `<st:email>ysc@cc.shu.edu.tw</st:email>`
- `</rdf:Description>`
- `</rdf:RDF>`

RSS

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
<channel>
  <title>XML 教學講義</title>
  <description>RSS簡介</description>
  <link>http://newsmeta.shu.edu.tw/seljuk/</link>
  <item>
    <title>xml是資訊傳播最基本的資料核心技術</title>
    <description>世新資訊傳播學系是網路傳播最專業的系所 </description>
    <link> http://newsmeta.shu.edu.tw/seljuk/mov/protocol/XMLSchema.mov</link>
  </item>
</channel>
</rss>
```

RSS頻道

RSS頻道描述

RSS基本項目

驗證：<http://feedvalidator.org/>



結論

- 支援語言中立(**language neutral**)
- 支援平台中立(**platform neutral**)
- 能提供定義在**Web**環境上結構化文件交換的資料格式
- **W3C**制定簡化**SGML**的子集合規範
- 於**1998年2月**正式公布**1.0**版本
- **XML**允許使用者自行定義所需的標示語言
 - 可以將資料內容以清楚的標籤表現其意義
- 並可廣泛地應用在各種領域
 - 例如：化學、電子、商業等等
- 應用相關的標籤界定個別的資料項目或資料群組
 - 應用程式能夠很容易地將**XML**文件中的資料分離而加以利用
- 可提供機器之間方便的資料交換格式

End of Part 1.

Unit. 2

Metadata



Metadata功能

- 在歐洲**DESIRE (Development of a European Service for Information on Research and Education)**數位圖書博物館計畫中，明確將**Metadata**的功能定位在位置標示：
 - 指引(location)
 - 尋找或發掘(discovery)
 - 文件記錄(documentation)
 - 評鑑(evaluation)
 - 判斷選擇(selection)

什麼是後設資料(metadata)?

- 參考農委會林業試驗所「[昆蟲標本館](#)」網站的「黑脈樺斑蝶」資料：

館號	00014002	種名	Salatura genutia genutia 【黑脈樺斑蝶】		
採集地點	TAIWAN Keelung City Keelung 【中華民國台灣 基隆市 基隆】				
採集者	S. T. Tsai				
採集時間	1993-06-23				
模式標本	【】	性別		體長	
保存方法	針插標本				
鑑定者	莊鈴木	鑑定日期		進館時間	
收藏歷史					
備註英文			備註中文		
標本照片					

什麼是後設資料(metadata)?

- 再到訪數位典藏「聯合目錄」看看相同的資料



出版者：

典藏機構與計畫:公開徵選計畫:行政院農業委員會:台灣林業試驗所昆蟲標本館

貢獻者：

採集者英文名:S. T. Tsai

日期：

採集日期:1993/6/23

來源：

<http://fact.tfri.gov.tw/smec/TFRI/>

語言：

中文

範圍：

採集地:基隆

採集地英文名:Keelung

緯度:25.108981

經度:121.708145

縣英文名:Keelung City

標本館號:00014002



簡而言之

- 上述例子的內容就是一筆**metadata**
 - 針對某一對象(通常指的是典藏品)所描述的資料紀錄
 - 這個資料紀錄除可詮釋典藏品背後的資訊
 - ◆ 對使用者而言，可更容易找尋資料、辨識資料；
 - ◆ 對管理者而言，可提供管理、統計、應用、交換...等建置管控的機制
- 簡單的講：**metadata**原本是運用在圖書館記錄書目資料的一種資訊組織方式，現在也被運用在描述記錄美術館、博物館藏品

上述是最常對**metadata**所作的解釋!!

如果**meatadata**如此容易解釋，就不會困擾大家許久!!

接著我們來討論**metadata**到底代表什麼意義？

可能討論的內容會非常艱澀與枯燥!!

但是**metadata**對資料傳播與處理有其重要性，因此我們還是需要釐清一下

依據應用的本質探討 **Metadata**



起源

- 源自希臘文：μ ε τ α

- 資料來源：David Haynes (2004) *Metadata for Information Management and Retrieval*, London: Facet, p.6.

- 最早可追溯**1960**年代，比較常出現的時候，則是在**1980**年代大量使用在資料庫管理系統**(DBMS)**的文獻上，通常是用來描述在“資料庫中資訊內容特性”的文件資訊

- 資料來源：John T. Phillips (1995) “Metadata: information about electronic records,” *ARMA Records Management Quarterly*, 29(4), p.52-57.



為何使用 metadata 這個詞

- 專業化的資訊組織
 - 涵蓋描述性的資料、結構性的資料、管理性的資料
- 標準化的資料描述
 - 考量資料交換、資料永久保存、通用的格式、軟體系統的支援度



不同應用角度的Metadata

■ 資料管理

- 用來描述資料庫內容資料的資料字典或目錄

- 資料來源：Bhavani Thuraisingham (2002) *XML Databases and the Semantic Web*, Boca Raton: CRC press, p.109.

- 資料的結構，及在組織之間被記錄、控制與發行(published)的行為(behavior)

- 資料來源：Guy V. Tozer(1999) *Metadata Management for Information Control and Business Success*, Boston, MA: Artech House. p.xix

■ 資料著錄

- 描述性(Descriptive) Metadata
- 結構性(Structural) Metadata
- 管理性(Administrative) Metadata
 - ◆ Rights management metadata
 - ◆ Preservation Metadata

- 資料來源：NISO. (2004) *Understanding Metadata*. Bethesda, MD: NISO Press, p.1



各種文獻對Metadata的定義

- 資料中的資料(Data about data)
- 描述資源屬性的資料 (Data describes attributes of resources)
- 有關資料背景與關聯性、資料內涵以及資料控制等相關資訊 (Information about the context of data and the content of data and the control of or over data)
- 任何協助辨識、描述與放置網路化電子資源的資料。目前存在許多不同的後設資料格式，有些在描述上很簡單，有些則相當複雜而豐富 (國際圖書館協會- IFLA)
- 記載資料元素或屬性(如：名稱、大小、資料類型等)的資料、有關記錄或資料結構(如：長度、欄位、行列等)，以及有關資料的資料(如：位置、關聯、擁有者等)
- 後設資料就是資料 (Metadata is data)



各種文獻對Metadata的定義(續)

- 後設資料是有關一個數位典藏品的資料，通常由典藏品的創作者或提供者來建立，並將數位典藏品串聯或埋置於後設資料中。因此，後設資料可以作為資訊儲存與檢索系統很有用的基礎（美國圖書館學會- **ALA**）
- 它最簡單的定義是“有關資料的結構性資料”（都柏林核心集- **DC Element Set**）



Metadata功能的解釋

- **Metadata**是描述資源屬性的資料，以及彼此間關係的特性，使資源能夠存在電子環境中有效地被找尋、管理和利用
 - ◆ Sherry L. Vellucci, 2000, Metadata and Authority Control, Library Resources & Technical Services, 44 no.1, pp.33-43



國內常見翻譯用語

- 元資料
- 詮釋資料
- 後設資料





元

■ 古籍

- 爾雅：元，始也
- 廣韻：長也，大也
- 前漢書：基事之本也
- 後漢書：元元本本，又百姓曰元元
- 公羊：元年者何？君之始年也

■ →歸納：最大的、開始的

■ 教育部字典補充

- 基本的：元素
- 量詞：計算錢幣的單位
- 朝代名
- 代數式中，稱表示未知數的文字：一元一次方程式



詮釋

■ 詮

- 說文：具也
- 玉篇：治亂之體也
- 廣韻：平也
- 類篇：擇言也，又解喻也
- 晉書：具說事理也

→歸納：具體、解說

■ 釋

- 說文：解也，取其分別物也
- 廣韻：捨也
- 前漢書：消也，散也
- 爾雅：釋，服也

→歸納：取捨、釋去

■ 詮釋

→歸納：具體的闡述，並分別內容、取捨內容



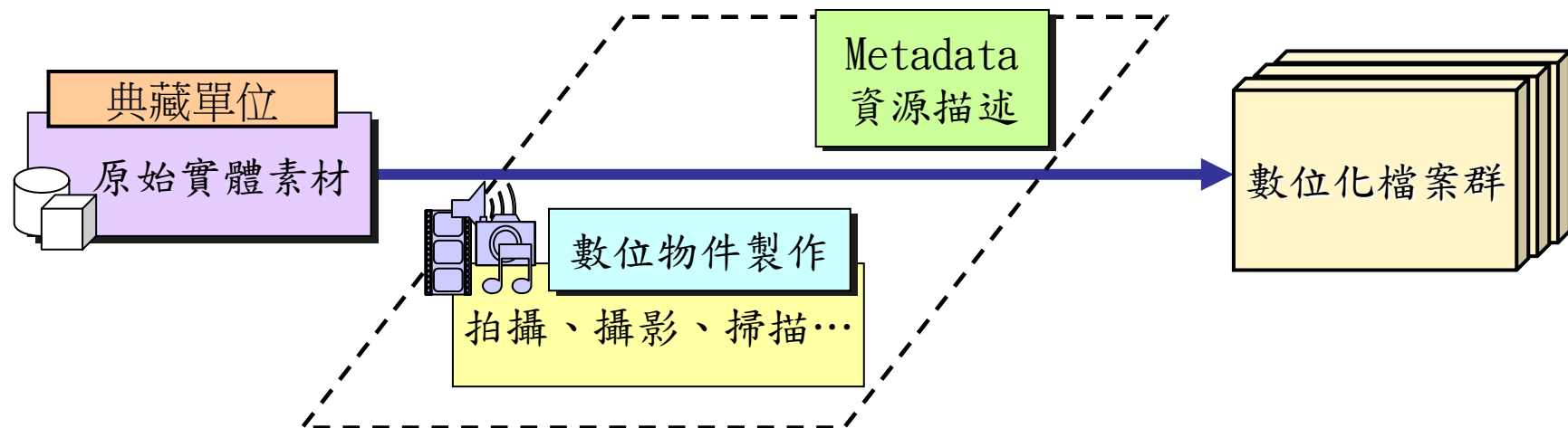
後設

- 文學中，一句話的出現受到前後章句的影響
 - 從語言及認知的過程，人類使用語言描繪一件事物，或敘述一個事件，都會受制於個人主觀的認知
 - →語言不純然是中性或客觀的
 - →後設小說
- 後設的思想源頭是尼采的詮釋論，愛因斯坦的相對論與海森堡的測不準原理
 - 尼采主張任何一項事實在認定之前，就滲入主觀的認知，也就是所謂的詮釋
 - 愛因斯坦認為真理不是絕對的
 - 海森堡的不確定原理，任何人為的衡量標準都會干擾到衡量的事物
 - ◆ 資料來源：蔡源煌(1992) "從浪漫主義到後現代主義", 台北：雅典，頁189-193。
- 後設認知：個人控制及引導心智歷程的現象
- 其他：後設戲劇、後設空間、後設分析、後設評鑑.....

數位資料組件

- 文件架構(Framework)
- 數位物件(Digital object)

- → Metadata + Multimedia





Metadata的類型-系統觀點

■ 描述性metadata

- 描述一項文件或資源的內涵與關聯性，以便發現與辨識資源
- 如：書目記錄、Dublin Core

■ 結構性metadata

- 數位典藏品實際的結構，提供資源的瀏覽、檢索與呈現
- 如：讓電子全文具翻頁功能、全文與相關影像的適切連結、書的章節結構

■ 管理性metadata

- 為長久管理、使用、呈現與保存數位化資源的相關資料
- 如：權利管理資訊、長久保存所需資訊

Metadata所指為何？

數位典藏聯合目錄 - Windows Internet Explorer

http://catalog.ndap.org.tw/dacs5/System/Introduction/Introduction_1.jsp

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

數位典藏聯合目錄

目錄系統，以匯整12主題約百個數位典藏子計畫的資料庫成果，提供民眾資訊的查詢，以及取用數位典藏資訊。

二、93年度計畫目標

本年度(93年)以上年度所開發的聯合目錄系統為基礎，重新規劃介面，並繼續對應各子計畫資料庫欄位，校正去年度匯入時的欄位錯誤，調整網站瀏覽的分類架構，並協調專家設法解決時間、空間的分類架構與缺字、異體字等技術問題。在匯入資料筆數方面，在去年共250筆測試的基礎上，今年將大幅增加，93年度上半年有15萬筆以上的後設資料 (metadata)可以匯入，未來將不斷擴增。聯合目錄系統的建置，由聯合目錄系統建置計畫、技術發展分項計畫人員共同合作。聯合目錄資訊系統，原則上以OAI方式建置。當更多的子計畫資料庫開始進行資料匯入工作，而且匯入資料的筆數直線上升的情況下，我們更有基礎來研究、分析OAI方式的優缺點。

三、現階段成果

(一) 建置聯合目錄系統平台雛形

完成

網際網路 110%

Metatata 意涵-1

定義

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="school">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name" />
        <xs:element ref="addr" />
        <xs:element ref="depart" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
      <xs:attribute name="id" type="xs:string" use="required" />
      <xs:attribute name="web" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="addr">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="city" minOccurs="0" />
        <xs:element ref="detail" />
      </xs:sequence>
      <xs:attribute name="zip" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="city">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
    </xs:complexType>
  </xs:element>
  <xs:element name="detail">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Metadata (format/scheme)

描述資料

題名	RFID sourcebook / Sandip Lahiri.
著者	Lahiri, Sandip.
出版者	Upper Saddle River, NJ : IBM Press, c2006.
稽核項	xvii, 276 p. : ill. ; 25 cm.
題名	RFID原理與應用 / 社團法人日本自動認識システム協會原著 刁建成編譯
著者	社團法人日本自動認識システム協會 著.
出版者	臺北市 : 全華科技圖書公司, 2005[民94].
稽核項	1冊 圖 23公分
語言別	chi jpn
標題	無線射頻辨識系統
其他著者	刁建成 編譯
索書號	312.9842 8546
ISBN	957-21-4843-5 (平裝) 新臺幣250元
稽核項	系統號 000164065
備註	Includes index.
標題	Radio frequency -- Identification. Radio frequency identification systems.
電子位置	數位物件
ISBN	0131851373 (hbk.) : NT\$1179

實體典藏品



資料

描述

資料

Metatata 意涵-2

定義

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="school">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name" />
        <xs:element ref="addr" />
        <xs:element ref="depart" minOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
      <xs:attribute name="id" type="xs:string" use="required" />
      <xs:attribute name="web" type="xs:string" />
    </xs:complexType>
  </xs:element>
  <xs:element name="addr">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="city" minOccurs="0" />
        <xs:element ref="detail" />
        <xs:attribute name="zip" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="city">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
    </xs:complexType>
  </xs:element>
  <xs:element name="detail">
    <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded" />
    </xs:complexType>
  </xs:element>
</xs:schema>
  
```

描述資料

題名	RFID sourcebook / Sandip Lahiri.
著者	Lahiri, Sandip.
出版者	Upper Saddle River, NJ : IBM Press, c2006.
稽核項	xvii, 276 p. : ill. ; 25 cm.
題名	RFID原理與應用 / 社團法人日本自動認識システム協會原著 刁建成編譯
著者	社團法人日本自動認識システム協會 著.
出版者	臺北市 : 全華科技圖書公司, 2005[民94].
稽核項	1冊 圖 23公分
語言別	chi jpn
標題	無線射頻辨識系統
其他著者	刁建成 編譯
索書號	312.9842 8546
ISBN	957-21-4843-5 (平裝) 新臺幣250元
稽核項	系統號 000164065
備註	Includes index.
標題	Radio frequency -- Identification. Radio frequency identification systems.
電子位置	數位物件
ISBN	0131851373 (hbk.) : NT\$1179

實體典藏品



Metadata (record)

資料

描述

資料

Metatata 意涵-3

定義

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
- <xs:element name="school">
- <xs:complexType>
- <xs:sequence>
- <xs:element ref="name" />
- <xs:element ref="addr" />
- <xs:element ref="depart" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="name">
- <xs:complexType mixed="true">
- <xs:choice minOccurs="0" maxOccurs="unbounded" />
- <xs:attribute name="id" type="xs:string" use="required" />
- <xs:attribute name="web" type="xs:string" />
</xs:complexType>
</xs:element>
- <xs:element name="addr">
- <xs:complexType>
- <xs:sequence>
- <xs:element ref="city" minOccurs="0" />
- <xs:element ref="detail" />
- <xs:attribute name="zip" type="xs:string" />
</xs:complexType>
</xs:element>
- <xs:element name="city">
- <xs:complexType mixed="true">
- <xs:choice minOccurs="0" maxOccurs="unbounded" />
</xs:complexType>
</xs:element>
- <xs:element name="detail">
- <xs:complexType mixed="true">
- <xs:choice minOccurs="0" maxOccurs="unbounded" />
</xs:complexType>
</xs:element>

```

描述資料

題名	●RFID sourcebook / Sandip Lahiri.
著者	●Lahiri, Sandip.
出版者	Upper Saddle River, NJ : IBM Press, c2006.
稽核項	xvii, 276 p. : ill. ; 25 cm.
題名	●RFID原理與應用 / 社團法人日本自動認識システム協會原著 刁連成編譯
著者	●社團法人日本自動認識システム協會 著.
出版者	臺北市 : 全華科技圖書公司, 2005[民94].
稽核項	1冊 圖 23公分
語言別	chi_jpn
標題	●無線射頻辨識系統
其他著者	●刁連成 編譯
索書號	312.9842 8546 D
ISBN	957-21-4843-5 (平裝) 新臺幣250元
稽核項	系統號 000164065
備註	Includes index.
標題	●Radio frequency -- Identification. ●Radio frequency identification systems.
電子位置	●數位物件
ISBN	0131851373 (hbk.) : NT\$1179

實體典藏品



Metadata (label)

由「功能與作用」思考

- 文內詮釋
- 結構定義
- 資料描述
- 內容界定

```
<?xml version="1.0" encoding="Big5"?>
<!DOCTYPE Booklist SYSTEM "D:\Book.dtd">
  <bib code="NE20118" type="b">
    <title>XML理論與實作導引</title>
    <authorlist>
      <author>余顯強</author>
    </authorlist>
    <price>390</price>
  </bib>
  <bib code="PG20030" type="s">
    <title>Delphi4系統開發與應用</title>
    <authorlist>
      <author>余顯強</author>
    </authorlist>
    <price>590</price>
  </bib>
</Booklist>
```

Attribute used as metadata

Data element used as metadata

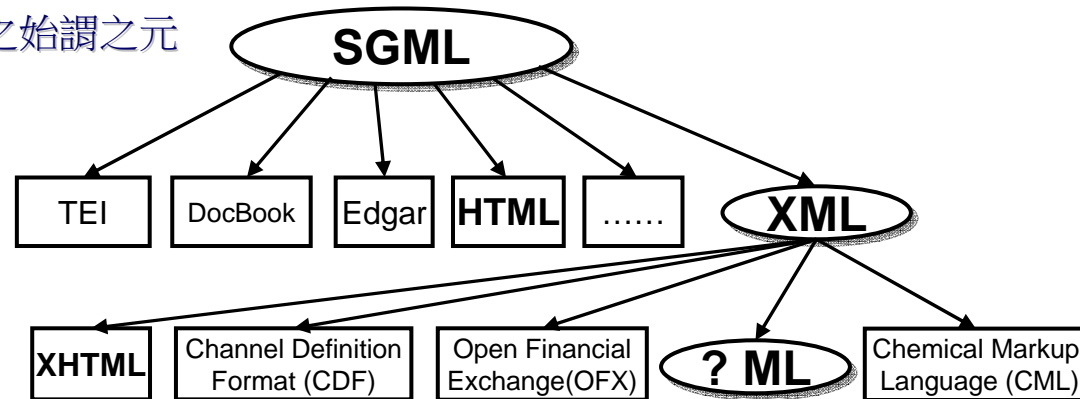


定義

- 就系統角度
 - metadata是規範著錄內容的格式，包括欄位名稱、順序、資料型態等。因此完整的名稱應稱為**metadata** 格式
- 就資料提供者的角度
 - metadata是為了達到原生資料(典藏品)的數位資訊應用，包括描述、管理、保存、利用等目的，而著錄的所有描述內容。因此完整的名稱應稱為**metadata**紀錄
- 就使用者的角度而言
 - metadata即是系統呈現任何有關原生資料(典藏品)的資訊內容

由「Meta字根延伸意義」思考

- 軟體工程的「繼承」
 - 易經—大哉乾元：萬物之始謂之元
- Meta-language



- Super-metadata
 - 用於繼承出下一代metadata的原始metadata
 - 資料來源：Chilvers Alison, Feather John (1998), "The management of digital data: a metadata approach", *The Electronic Library*, 16(6), p.365-372.
- Meta-metadata
 - 描述metadata記錄之實際個體片段的資料
 - (What makes a particular piece of data metadata is its context.)
 - 資料來源：Kal Ahmed et al.,(2001) *Professional XML meta data*, UK: Wrox Press, p.108-110.
- Meta-data ?

由「語法與結構模式」思考

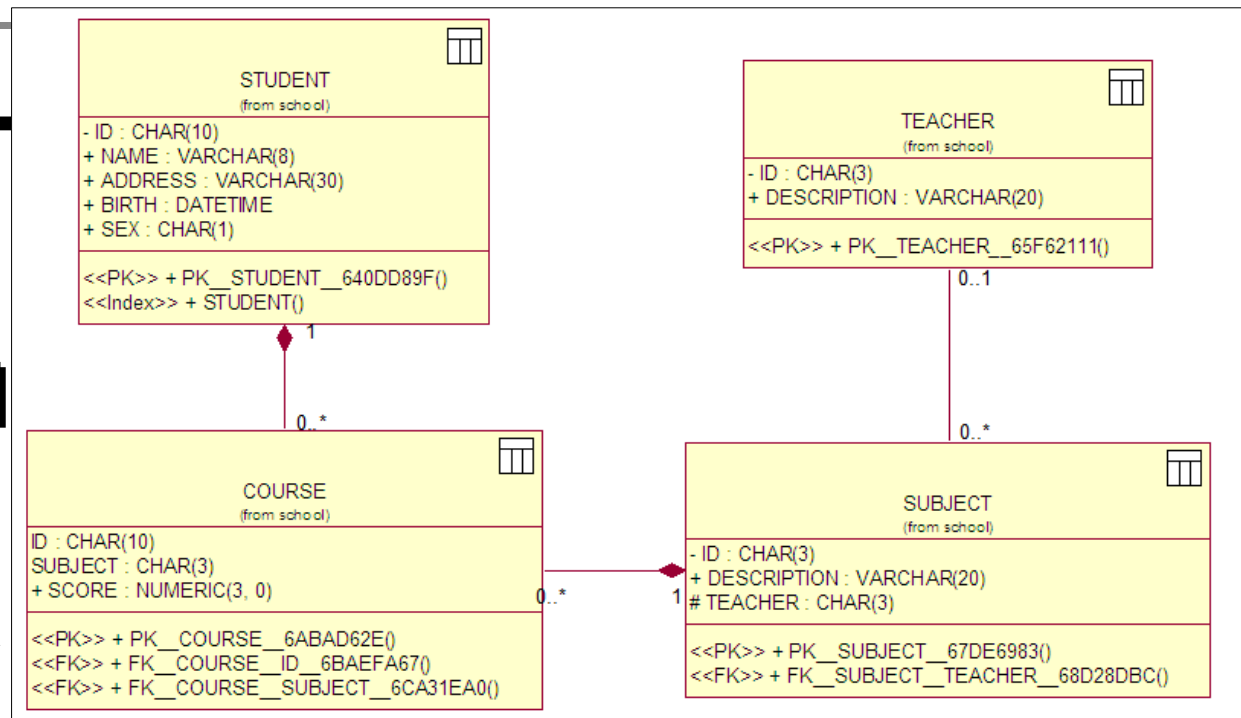
■ HTML <MET

■ Validate XM

- DTD

- XML Schema

■ Relational database tables



```
<xs:attribute name="zip" type="xs:string" />
</xs:complexType>
</xs:element>
- <xs:element name="city">
- <xs:complexType mixed="true">
- <xs:choice minOccurs="0" maxOccurs="unbounded" />
- </xs:complexType>
- <xs:element name="detail">
- <xs:complexType mixed="true">
- <xs:choice minOccurs="0" maxOccurs="unbounded" />
- </xs:complexType>
</xs:element>
```



Metadata 中譯如何？

- 審視電腦處理**Metadata**的語法標準
 - 標示語言的功能性
- **Metadata**記錄了哪些內涵
 - 資料結構在資訊系統上用途
- 一致使用的名稱 **vs.** 貼切的中文語意
 - 望文生意是否真能表達現實



Metadata相關技術

- **Metadata的語意**
 - AACR2、Dublin Core、AAT (Art and Architecture Thesaurus)、LCSH (Library of Congress Subject Heading)、LCNA (Library of Congress Name Authority File)
- **Metadata的結構**
 - RDF, DTD, XML Schema
- **Metadata的語法**
 - SGML, HTML, XML
- **Metadata的查詢(發佈與分享)**
 - Z39.50, OAI, OpenURL
- **Metadata的顯示**
 - XSL/XSLT, CSS

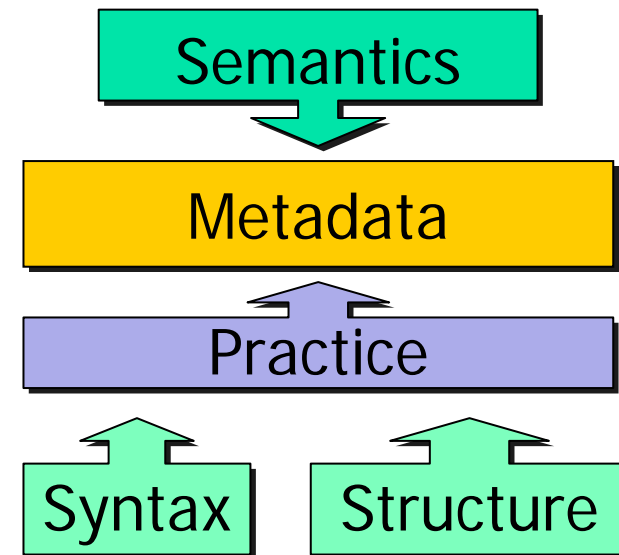


Metadata格式範圍-中研院分類

- 檔案全文
 - 如EAD (Encoded Archival Description) 、 ISAD (International Standard Archival Description) 、 TEI (Text Encoding and Interchange)等
- 博物館典藏品
 - 如CDWA (Categories for the Description of Works of Art 、 CIMI (Consortium for the computer Interchange of Museum Information)等
- 地理空間資訊
 - 如 FGDC's CSDGM(Content Standards for Digital Geospatial Metadata) 、 NGDF等
- 網路WWW資源
 - 如Dublin Core 、 IAFA等
- 共通性或跨學科領域的資訊
 - 如Dublin Core
- **Metadata**的架構
 - 如Warwick Framework 、 RDF (Resource Description Framework)等
- 其他
 - 如GILS (Government Information Locator Service) 、 MARC (MACHINE Readable Cataloging)等

Metadata的涵義範圍

- **Metadata**格式的設計必需考慮三個層面
 - 語意(semantic)
 - 語法(syntax)
 - 結構(structure)
- 若加上應用，則需再考慮
 - 整合檢索的問題





Metadata的功効

- 如何管理與搜尋龐大的數位典藏品
(Discovery)
- 如何確保數位典藏品的長久保存與取得
(Longevity)
- 如何讓數位典藏品達到互通與共享
(Interoperability)



討論

- **MARC會被淘汰?**
- **MARC format 下一步?**
 - 僅注重制定欄位的規範? 還是涵蓋格式的宣告?
 - 格式宣告的模式採用哪種方式?
- **ISO2709應該轉型?**
- **圖書館界的關鍵點會在哪裡?**



參考資源

- 電子書

- Baca, Murtha, ed. Introduction to Metadata: Pathways to Digital Information. Los Angeles: J. Paul Getty Trust, 1998.

- ◆ <http://www.getty.edu/research/institute/standards/intrometadata/>

- 研究報告

- Committee on Cataloging: Description and Access Task Force on Metadata: Final Report, June 2000

- ◆ <http://www.ala.org/alcts/organization/ccs/ccda/tf-meta6.html>



問題與討論

