

XML/DTD理論實務與應用(2) - DTD介紹

華藝數位 陳嵩榮

甚麼是DTD?

- 文件型別定義(Document Type Definition)
- DTD的概念緣於SGML，每一份SGML文件，均有相應的DTD
- XML沿用了SGML DTD的概念，但對XML文件而言，DTD可有可無
 - well-formed XML文件：
 - 沒有對應DTD或XML Schema
 - 只需符合well-formed XML 規則
 - valid XML文件：
 - 有對應DTD或XML Schema
 - 除符合well-formed XML規則外，文件內容結構須符合DTD或XML Schema之規範

DTD的功能

- 定義該類型文件所包含的元素(Element)，並定義每個元素的內容，包含子元素、文字內容與屬性。
- 規範各元素(Element)的排列組合方式，包含出現的順序與可出現的次數。
 - 順序性與重複性

email DTD範例

```
<!-- Elements Content -->
<!-- ----->
<!ELEMENT email (from,date,to+,subject,body?)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (#PCDATA)>
<!-- End of email DTD -->
```

DTD的組成

- 元素 (Elements)
- 屬性 (Attribute)
- 實體 (Entities)
- 註解 (Comments)

XML DTD /元素(Element)

- 元素宣告語法如下：
<!ELEMENT 元素名稱 元素內容>
- 語法說明：
 - <!ELEMENT 元素宣告的起始符號
 - 元素內容可以是一群子元素、#PCDATA(表示文字資料)、EMPTY(空元素)、ANY(不做限制的元素)
 - > 宣告的結束符號

元素宣告實例(1)

元素內容包含一群子元素

<!ELEMENT email (from, date, to+, subject, body?)>

● 語法說明：

- email 是宣告的元素名稱
- (from, date, to+, subject, body?) 是email的內容，包含一群子元素

元素內容為文字資料

<!ELEMENT from (#PCDATA)>

● 語法說明：

- from 是宣告的元素名稱
- (#PCDATA) 是 from 的內容，代表資料型態是可被剖析的文字資料 (Parsed Character DATA)

XML文件標示實例

```
<?xml version="1.0" standalone="no">
```

```
<!DOCTYPE Email SYSTEM "http://mydtlds.com/email.dtd">
```

```
<email>
```

```
<from>Joe</from>
```

內容包含了 From, Date, To, Subject, Body 等子元素

```
<date>1999-7-14 AM 09:20</date>
```

內容是文字資料

```
<to>Jay</to>
```

to 重複出現

```
<to>Jennifer</to>
```

```
<subject>Learning XML</subject>
```

```
<body>學XML就像學HTML一樣簡單</body>
```

```
</email>
```

元素宣告實例(2)

❁ 空元素(Empty Element)

- 元素之下沒有任何的子元素或內容

❁ <!ELEMENT br EMPTY>

● 語法說明：

- br 是宣告的元素名稱
- EMPTY 代表 br 是一個空元素，br 沒有包含任何子元素或內容

● XML文件標示片段：

```
.....  
<br/>  
.....
```

元素宣告實例(3)

❁ 內容不做限制的元素：

- 元素之下可以包含任何被宣告過的元素或文字資料(#PCDATA)
- 這些被包含的元素出現順序與次數皆不受限制

❁ <!ELEMENT comment ANY>

● 語法說明：

- comment 是宣告的元素名稱
- ANY 代表 comment 是一個內容不做限制的元素，可包含任何宣告過的元素或文字資料

XML DTD / 元素的内容模型(Content Model)

()	一個表示式
A	A 必須出現一次 (出現1次)
A?	A 可出現一次或不出現 (出現0 or 1次)
A+	A 可重複出現且至少出現一次 (出現1次以上)
A*	A 可重複出現或不出現 (出現0次以上)
(A B C)	A, B, C 任選一出現, 且僅能有一出現
(A, B, C)	A, B, C 同時出現, 且出現順序為 A, B, C
(A B C)+	...
(A?, B*, C+)	...

混合型(Mixed) 內容模型

- 當一個元素內容同時包含文字資料與子元素時, 稱做混合型內容
- `<!ELEMENT 父元素 (#PCDATA | 子元素1 | 子元素2 | | 子元素n)*>` 或
`<!ELEMENT 父元素 (#PCDATA | 子元素1 | 子元素2 | | 子元素n)+>`
 - 語法說明:
 - 宣告時 (#PCDATA) 一定要放在其他子元素前面
 - 限制: 無法規範文字資料與各元素的出現順序與重複次數
 - 與 ANY 宣告比較?

XML DTD / 屬性(Attributes)

- 提供元素額外的描述資訊

- 屬性宣告語法如下：

<!ATTLIST 元素名稱 屬性名稱 屬性值型態 屬性的內定值>

- 語法說明：

- <!ATTLIST 屬性宣告的起始符號
- 元素名稱必須是宣告過的元素
- 屬性值型態：指定該屬性的資料型態
- 屬性的內定值：屬性是必備或可有可無？是否有固定值或列舉值？
- > 宣告的結束符號

- 範例：

```
...  
<!ELEMENT tel (#PCDATA)>  
<!ATTLIST tel zone CDATA #IMPLIED>  
...
```

指定屬性值型態

- <!ATTLIST 元素名稱 屬性名稱 屬性值型態 屬性的內定值>

<!ATTLIST tel zone CDATA #IMPLIED>

- 幾種常用的屬性值型態：

- CDATA : Character DATA，代表一般文字。
- 列舉值(Enumerated)：列舉該屬性的值，應用時屬性值必須採用列舉值其中之一
- ID：屬性值在XML文件中必須是唯一的
- IDREF：屬性值必須是XML文件中某個ID
- IDREFS：屬性值由XML文件中一個以上的ID所組成，之間以空白格開
- NMTOKEN：英文字母、數字、“_”、“-”、“.”、“:”
- NMTOKENS：一個以上的NMTOKEN所組成，之間以空白隔開
- NOTATION：屬性值是 DTD 宣告的 NOTATION

指定屬性的內定值

❏ <!ATTLIST 元素名稱 屬性名稱 屬性值型態 屬性的內定值>
<!ATTLIST tel zone CDATA #IMPLIED>

❏ 幾種屬性的內定值：

- #REQUIRED：該元素必須設定該屬性
- #IMPLIED：該元素不一定要設定該屬性
- #FIXED：有固定的屬性值
- 預設的字串：若沒有指定屬性值，則採用預設值

屬性宣告實例(1)

❏ <!ELEMENT tel (#PCDATA)>
<!ATTLIST tel zone CDATA #IMPLIED>

- 語法說明：
 - tel 是元素名稱
 - zone 是 tel 的屬性名稱
 - CDATA 代表 zone 是文字資料型態
 - #IMPLIED 代表 zone 屬性可有可無
- XML文件標示片段：

```
...  
<tel zone="02">82215585</tel>  
...
```


屬性宣告實例(2)

❖ <!ELEMENT name (#PCDATA)>
<!ATTLIST name sex (M | F) #REQUIRED>

- 語法說明：
 - name 是元素名稱
 - sex 是 name 的屬性名稱
 - (M | F) 是 sex 的列舉值，屬性值只能從中選一
 - #REQUIRED 代表 sex 屬性必備
- XML文件標示片段：

```
...  
<name sex="M">Joe</name>  
...
```

XML DTD /實體(Entity)

❖ XML允許使用者將某一段任意型別的文字資料宣告為實體，以簡化XML文件或DTD內容的撰寫，XML 剖析器處理到這些實體時，會自動以預先定義的資料來取代

❖ 實體宣告語法如下：

<!ENTITY 實體名稱 實體內容>

- 語法說明：
 - <!ENTITY 實體宣告的起始符號
 - > 宣告的結束符號

實體宣告實例(1)

通用實體(General Entity)

- 主要用來取代XML文件中常用到的長字串

<!ENTITY ROSS “Resource Organization and Searching Specification”>

- 語法說明：
 - ROSS 是宣告的實體名稱
 - “Resource Organization and Searching Specification” 是 ROSS 所代表的字串，在XML文件中，可以 &ROSS; 表示
- XML文件標示片段：

```
...  
Metalogy is produced by the research group under &ROSS;  
...
```

實體宣告實例(2)

參數實體(Parameter Entity)

- 主要用來取代DTD中常用到的長字串

<!ENTITY %m.phrase.links “ilink | nameloc | nmlist | ptr | extptr”>

- 語法說明：
 - % 參數實體名稱的前引字元
 - m.phrase.links 是宣告的實體名稱
 - “ilink | nameloc | nmlist | ptr | extptr” 是 %m.phrase.links 所代表的內容，在XML文件中，可以 %m.phrase.links; 表示
- DTD內容片段：

```
...  
<!ELEMENT a (%m.phrase.links)*>  
...
```

XML DTD / 註解(Comments)

● 增加DTD的可讀性

● 註解語法如下：

`<!-- 註解內容 -->`

● 語法說明：

- `<!--` 註解的起始符號
- `-->` 註解的結束符號

註解實例

● `<!--3 ELEMENT 版本名稱 (#PCDATA) -->`
`<!ELEMENT editionName (#PCDATA) >`

`<!--ATTLIST 版本名稱 使用代碼 FIXED "npm-doc-
editionName"-->`

`<!ATTLIST editionName scheme FIXED "npm-doc-
editionName">`

● 說明：

- 一般元素名稱與屬性名稱會採用英文，每一行宣告可加一行註解，記錄元素與屬性的中文名稱，增加可讀性
- 元素宣告註解可以數字表示該元素在DTD內的層次(根元素編為0)，有助於辨識父元素與子元素

如何開發 DTD

分析文獻結構

- 畫出結構圖或以表格表現層次

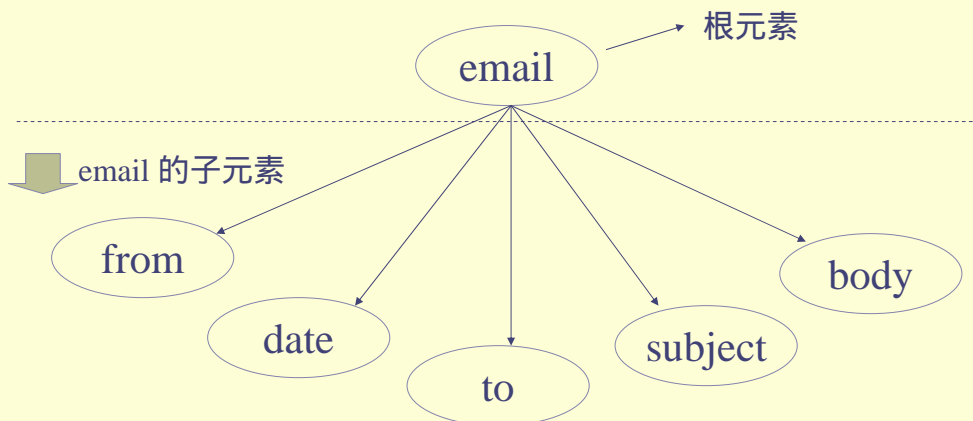
確認每個元素的內容規格

- 包含子元素、文字資料或是空元素？
- 所包含的每個子元素是否必備？是否可重複？
- 所包含的子元素間順序性如何？
- 是否有屬性？屬性值是否有規範？

開始撰寫DTD

- 加入適當註解，增加可讀性

email的文件結構圖



使用DTD的好處

- 統一文件格式：藉DTD來進行文件內容與結構的驗證。
- 提高文件的再利用性

使用DTD的代價

- 如果文件結構過於複雜，訂定出完整且具擴充彈性的DTD相對不易。
- 必須搭配相關的應用程式或剖析器(parser)才能發揮DTD的作用。

如何取得 DTD?

- 使用某單位或某人已開發的DTD.

```
<!DOCTYPE Email PUBLIC "-//OPEN DTD//DTD  
Email//EN" "http://opendtd.org/dtd/email.dtd">
```

- 自行開發DTD.

```
<!DOCTYPE Email SYSTEM  
"http://mydtlds.com/email.dtd">
```

Valid XML 文件

- 一份附帶有外部或內部DTD的Well-formed XML 文件，並且文件內容符合DTD所定義的文件格式，稱之為Valid XML文件。

撰寫Valid XML文件

1. 使用純文字編輯器(如Windows的記事本)
2. 撰寫XML宣告
3. 設計DTD，存檔時副檔名設為 .dtd
4. 撰寫XML標籤與內容
5. XML文件存檔時副檔名設為 .xml
6. 以軟體(如XMLSPY) 驗證XML文件是否Valid

撰寫 XML宣告

🌐 幾種XML宣告：

- <?xml version="1.0"?>
- <?xml version="1.0" encoding="UTF-8" ?>
- <?xml version="1.0" encoding="big5" ?>
- <?xml version="1.0" standalone="yes" ?>
- <?xml version="1.0" standalone="no" ?>

🌐 語法說明：

- <? 是XML文件宣告的起始符號
- xml 表示這是一份XML文件
- version 用來指定XML標準的版本，目前是"1.0"
- encoding 用來指定XML文件所使用的字集編碼，預設是"UTF-8"
- standalone 用來指定XML文件是否獨立存在，預設是 "yes"
- ?> 是XML文件宣告的結束符號

設計DTD

(<http://mydtlds.com/email.dtd>)

```
<?xml version="1.0">
<!ELEMENT email (from,date,to+,subject,body?)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

撰寫XML標籤與內容

- DTD宣告
 - 內部(Internal) DTD 宣告
 - DTD在XML文件內部
 - 外部(External) DTD 宣告
 - DTD在XML文件外部

內部DTD宣告實例

```
<?xml version="1.0">
<!DOCTYPE email [
<ELEMENT email (from,date,to+,subject,body?)>
<ELEMENT from (#PCDATA)>
<ELEMENT date (#PCDATA)>
<ELEMENT to (#PCDATA)>
<ELEMENT subject (#PCDATA)>
<ELEMENT body (#PCDATA)>
]>
<email>
  <from>Joe</from>
  <date>1999-7-14 AM 09:20</date>
  <to>Jay</to>
  <to>Jennifer</to>
  <subject>Learning XML</subject>
  <body>學XML就像學HTML一樣簡單</body>
</email>
```

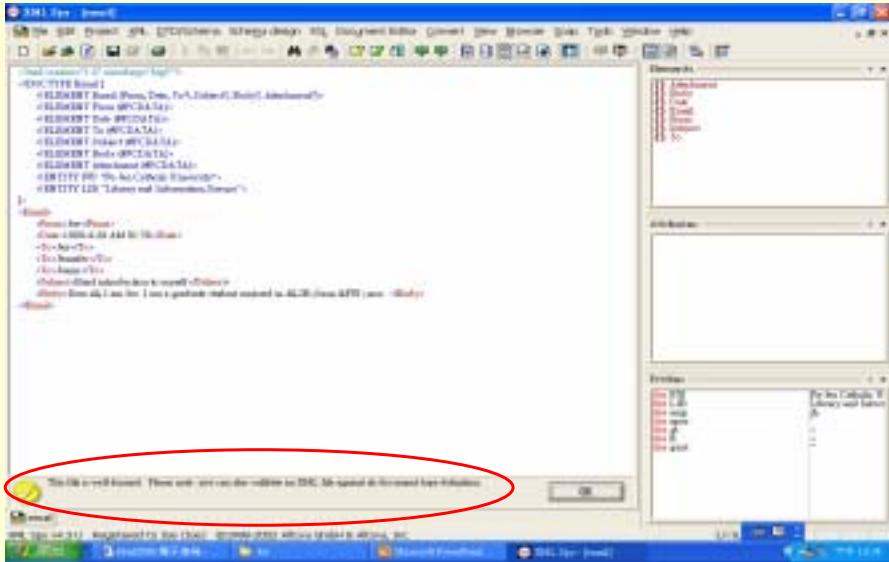
語法說明：

- <!DOCTYPE 是DTD宣告的起始符號
- email 是根元素名稱
- [是DTD內容的起始符號
-] 是DTD內容的結束符號
- > 是DTD宣告的結束符號

外部DTD宣告實例(1/2)

```
<?xml version="1.0" standalone="no">
<!DOCTYPE email SYSTEM "http://mydtlds.com/email.dtd">
<email>
  <from>Joe</from>
  <date>1999-7-14 AM 09:20</date>
  <to>Jay</to>
  <to>Jennifer</to>
  <subject>Learning XML</subject>
  <body>學XML就像學HTML一樣簡單</body>
</email>
```

利用XML SPY 2004 對XML文件進行驗證(1) - 檢查文件是否well-formed

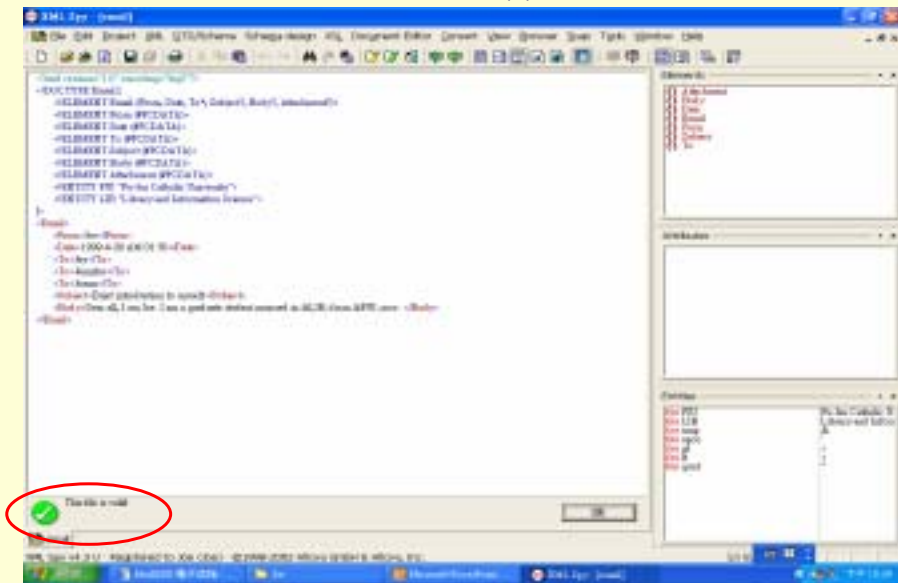


2004/7/26

華藝數位藝術股份有限公司 版權所有
2004 All Rights Reserved

35

利用XML SPY 2004對XML文件進行驗證(2) - 檢查文件是否Valid



2004/7/26

華藝數位藝術股份有限公司 版權所有
2004 All Rights Reserved

36