linguaggio xml

A cura dell'Ing. Buttolo Marco

SOMMARIO:

Contenuti

INTRODUZIONE:

XML è l'acronimo di **eXtensible Markup Language** ed è un metalinguaggio di markup. In poche parole, XML è un linguaggio che permette di definire la struttura di base di un generico documento, ossia permette di generare documenti ben strutturati. Attualmente i documenti XML hanno sostituito i vecchi file INI ossia i vecchi file di inizializzazione ed i vecchi file di configurazione per molti programmi.

STRUTTURA DOCUMENTO XML:

XML, le cui specifiche ufficiali sono state definite dal W3C (World Wide Web Consortium), ha lo scopo di creare nuovi linguaggi atti a descrivere documenti ben strutturati. Un documento XML alla fine dei conti altro non è che un banalissimo file di testo contenente tutta una serie di attributi, tag, e testo secondo regole precise.

Vediamo un semplicissimo esempio:

```
<?xml version="1.0"?>
<Libro>
  <titolo="IT">
  </titolo>
  <autore="Stephen King">
  </autore>
  <editore="Sperling paperback">
  </editore>
</Libro>
```

In poche parole un documento XML è composto da una intestazione che indica la versione del documento (1.0 in questo caso). Successivamente inizia il vero e proprio contenuto del documento XML. Ogni tag è delimitato dai due caratteri '<' e '>'. A differenza dell'HTML in cui i tag sono predefiniti, XML lascia piena libertà al progettista sulla definizione dei vari tag. Come si vede chiaramente dall'esempio, è possibile specificare un attributo per ciascun tag inserendo il nome dell'attributo con il relativo valore all'interno del tag stesso di apertura.
Osservando attentamente la struttura del file XML (file con estensione .xml) è possibile intuire la struttura intrinsecamente gerarchica dello stesso:

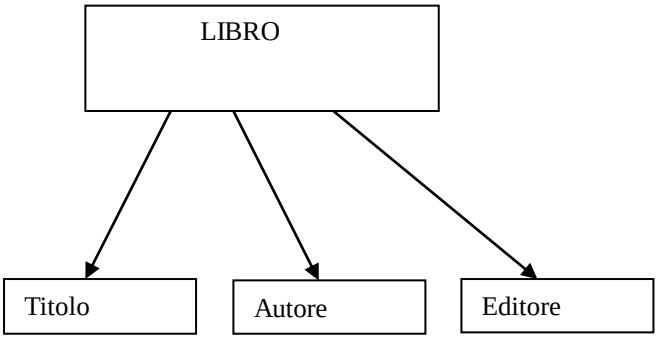

Figura 1

Quindi un documento XML è composto da:

- **ELEMENTI** (es: LIBRO)
- **ATTRIBUTI** (es: <titolo="IT">)
- **DICHIARAZIONI** (<!....>)
- **COMMENTI** (<!--....-->)
- **SEZIONI DATA**
- **PROCESSING INSTRUCTION** (<?...?>)
- **XML DECLARATION**

L'istruzione seguente:

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

È una processing instruction, e viene usata per definire la versione del documento XML. Non è obbligatoria ma è sempre consigliata. Se presente, deve stare in cima al documento, e l'attributo VERSION è sempre obbligatorio. La codifica di default, se non specificata, è UTF-8.
Un elemento è delimitato obbligatoriamente da un tag di apertura e da un tag di chiusura. Ogni cosa tra i due tag viene detta **contenuto**. Pertanto se abbiamo una situazione del seguente tipo:

<tag> xxx </tag>

Xxx è il relativo contenuto. I caratteri speciali '<' e '>' vengono detti **delimitatori di markup**. Osservando attentamente la figura 1 si intuisce la struttura ad albero di un documento XML. In particolare, ogni documento XML ha un elemento padre di tutti denominato **root element**. Ogni documento XML ha un solo elemento che non ha il genitore ed è proprio il root element. Ogni elemento, eccetto root element, ha un solo padre. Per quanto riguarda gli attributi, un elemento può non avere attributi oppure averne uno o più di uno. In particolare, ogni attributo è così strutturato:

(nome, valore)

Il carattere separatore è il carattere '=' ed il valore viene racchiuso o tra doppi apici o tra singoli apici. Vediamo il seguente esempio:

```
<?xml version="1.0"?>
<auto>
  <colore="rosso">
  </colore>
  <targa="123">
  </targa>
</auto>
```

I nomi degli elementi e degli attributi possono iniziare con qualsiasi carattere o con '_'. I documenti XML vengono salvati in file con estensione .XML.

DOCUMENTI XML BEN FORMATI:

Un documento XML è tale se è ben formato. Ma cosa vuol dire che un documento XML è ben formato? In poche parole, un documento XML ben formato è un documento che segue determinate caratteristiche dettate da W3C:

- Ogni documento XML deve contenere uno ed un solo element root.
- Ogni elemento aperto deve possedere un tag di chiusura
- I tag di chiusura devono seguire quelli di apertura nel giusto ordine.
- Dato che **XML è case-sensitive** (fa distinzione tra maiuscolo e minuscolo) i nomi dei tag in fase di apertura e chiusura devono essere identici.
- I valori degli attributi devono essere racchiusi tra doppi o singoli apici.

Come evidenziato, una fondamentale caratteristica dell'XML è la sua estendibilità. Ciò chiaramente porta la necessità di definire regole grammaticali ai quali gli elementi devono attenersi. La **DTD** (**Document Type Definition**) permette di realizzare ciò. In poche parole, un DTD è un documento che descrive i tag utilizzabili nel documento XML e la loro relazione. La sintassi di un DTD si basa su due dichiarazioni:

- <!ELEMENT>
- <!ATTLIST>

La prima definisce gli elementi utilizzabili nel documento XML, mentre la seconda definisce la lista degli attributi per ciascun elemento. Vediamo, a titolo di esempio, un DTD:

```
<!ELEMENT Autore (#PCDATA)>
<!ELEMENT Titolo (#PCDATA)>
<!ELEMENT test ANY)
<!ATTLIST test attr1 CDATA #REQUIRED>
```

Analizziamo brevemente l'esempio appena mostrato. Autore, Titolo, e test sono elementi. (#PCDATA) specifica che l'elemento contiene del testo. In caso contrario, ossia se è presente ANY allora vuol dire che l'elemento in questione ha un contenuto non definibile a priori. L'elemento "test" ha un attributo denominato attr1 obbligatorio (specificato da #REQUIRED).
Un file DTD può essere esterno, e quindi richiamato dal file XML oppure all'interno del documento XML stesso. Un documento ben formato e che presenta una DTD si dice **valido**.

VISUALIZZAZIONE DOCUMENTI XML:

Un documento XML non può essere visualizzato così come è. Per poter visualizzare un documento XML senza visualizzarne la struttura è possibile seguire due strade:

- I fogli di stile CSS
- XSL (XML StyleSheet Language)

I fogli di stile sono sostanzialmente dei files con estensione **CSS** (**Cascade Style Sheet**). Il CSS è fondamentalmente un linguaggio per definire gli stili di visualizzazione degli elementi nei linguaggio HTML, XML, e XHTML. Quindi con il CSS è possibile associare ad ogni elemento il suo stile di visualizzazione. Vediamo subito un esempio:

```
titolo
{
    margin-bottom: 1pt;
    text-align: center;
    font-size: 24pt;
    display: block;
    color: gray;
}
```

Come si può facilmente notare è possibile associare al tag "titolo" tutta una serie di parametri di visualizzazione come per esempio il margine in alto, l'allineamento del testo, la dimensione del font, eccetera. Il seguente esempio mostra l'uso del foglio di stile all'interno del file XML.

```
<?xml:stylesheet type="text/css" href="test.css" ?>
<Libro>
  <Titolo titolo="IT">
  </Titolo>
  <Autore autore="Stephen King">
  </Autore>
  <Editore editore="Sperling paperback">
  </Editore>
</Libro>
```

XSL (XML StyleSheet Language) è un linguaggio del W3C (World Wide Web Consortium). E' composto da:

- **XSLT** il quale è un linguaggio in grado di trasformare documenti XML in altri formati.
- **XPATH**, il quale permette di definire dei metodi per accedere ai nodi del documento XML.
- **XSL FO (Formatting Object)** che consente di trasformare in modo preciso un oggetto trasformato.

In questa guida ci concentreremo sui primi due. Come già accennato tramite XSLT è possibile effettuare la migrazione da un documento XML ad un documento valido e ben formato sempre XML, per esempio da un documento XML ad un documento XHTML.

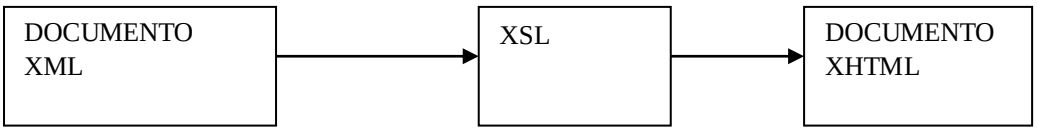

Figura 2

XSL è più potente di CSS in quanto permette di:

- Aggiungere nuovi elementi alla struttura
- Creare nuovi contenuti
- Filtrare ed ordinare dati
- Generare documenti con differenti gradi di compatibilità

I files con estensione XSL sono i files con dentro le regole specificate tramite il linguaggio XSL. Tramite le seguenti istruzioni è possibile associare un file XSL ad un documento XML:

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="news.xsl"?>

Un document XSL è un document XML ben formato. Esso è composto da:

- Un prologo XML
- Un elemento radice
- I vari TAG intermedi che devono essere obbligatoriamente chiusi.

Il seguente esempio mostra un file XSL.

```
<?xml version="1.0" encoding="UTF-8"?>   <!-- Prologo XML -->
<xsl:stylesheet version="1.0"
     xmlns:xsl="http://www.w3.org/1999/XSL/Transform">   <!-- Dichiarazione del documento XSLT -->
</xsl:stylesheet>
```

In un generico documento XSL l'elemento radice è: <xsl:stylesheet>. Tale istruzione è composta da:

- Un attributo che definisce la versione
- Una dichiarazione del namespace (xmlns:xsl="http://www.w3.org/1999/XSL/Transform">)

E' possibile utilizzare la seguente istruzione per poter definire il formato del documento finale. Per esempio:

<xsl:output method="html"/>

Pertanto un foglio di stile XSLT vede un document XML come un albero fatto di nodi, e pertento un processore XSLT può trattare i seguenti tipi di nodi:

- **l'elemento radice**
- **gli attributi e i loro valori**
- **i commenti**
- **gli elementi**
- **i namespace**
- **le istruzioni di elaborazione**
- **il testo contenuto in un nodo**

Nel foglio XSLT bisogna specificare delle regole per la trasformazione dei singoli nodi presenti in un documento XML. Per esempio:

<xsl:template match="test">

<xsl:value-of select="test"/>

</xsl:template>

In questo modo il processore quando, scandendo il documento XML, troverà il tag TEST, ne restituirà il valore dell'attributo o del singolo elemento. Qui di seguito vengono elencati i tag XSLT più comunemente usati:

TAG	DESCRIZIONE
Xsl:apply-templates	Applica le regole definite in un template.
Xsl:attribute	Crea un attributo per elemento
Xsl:element	Crea un nuovo elemento
Xsl:for-each	Applica un template ogni volta che incontra un determinato nodo

Xsl:if	Per strutture con condizioni
Xsl:include	Per includere un foglio XSL esterno
Xsl:template	Genera un template
Xsl:value-of	Restituisce il valore o di un elemento o di un attributo.
Xsl:text	Genera nuovo testo del documento di output

Rintracciare i nodi

Per individuare i singoli nodi si ricorre alla sintassi Xpath. Per capire come funziona Xpath si può fare riferimento al modo in cui si individuano file e cartelle in un file-system. Se scrivo: C:documentiritratto.jpg significa che individuo:

- un elemento radice (C:)
- una cartella al suo interno
- un file all'interno della cartella

Se volessi specificare il path per l'elemento <testo> del file news.xml scriverei così in Xpath:

<xsl:template match="news/post/testo">

La sintassi Xpath è notevolmente più complessa e se ben usata consente di individuare e trasformare ogni aspetto di documenti XML, anche molto complessi.