

XML DTD

Se vogliamo specificare esattamente il tipo di documento rubrica e la sua struttura dobbiamo definire una DTD, cioè una Document Type Definition. Ad esempio, per la rubrica una DTD può essere la seguente:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE rubrica [
  <!ELEMENT rubrica (persona)+>
  <!ELEMENT persona ( nome, cognome, (telefono)+, (email)*)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT cognome (#PCDATA)>
  <!ELEMENT telefono (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
]>
```

La sintassi di un Dtd si basa principalmente sulla presenza di due dichiarazioni: <!ELEMENT> e <!ATTLIST>. La prima definisce gli elementi utilizzabili nel documento e la struttura del documento stesso, la seconda definisce la lista di attributi per ciascun elemento e verrà analizzata nel seguente paragrafo.

Quindi !DOCTYPE rubrica serve per dichiarare l'elemento radice del documento XML; dentro le parentesi quadre vi sono definiti ed elencati tutti gli elementi costitutivi, il relativo tipo di dato e modalità di inserimento.

Per quanto riguarda quest'ultima, vediamo come il primo elemento, (persona)+, sia seguito dal simbolo "+"; ciò afferma che ci deve essere uno o più elementi "persona" all'interno del documento. Esistono quindi differenti modi per definire la struttura logica di un elemento; questi sono schematizzati qui sotto:

(xxx,yyy)	L'elemento deve contenere xxx e yyy nell'ordine definito
(xxx yyy)	L'elemento deve contenere xxx o yyy
xxx?	L'elemento può avere nessuno o un xxx
xxx*	L'elemento può avere nessuno o più xxx
xxx+	L'elemento può avere uno o più xxx
xxx	L'elemento deve contenere xxx

Quindi una persona è composta da un nome, un cognome, uno o più telefoni, nessuna o più email.

nome (#PCDATA) specifica invece che tipo di dato sarà l'elemento nome, #PCDATA indica un insieme generico di caratteri. Altri tipi di dati sono mostrati di seguito:

#PCDATA	(Parsable Character Data) Sono caratteri di markup, che possono quindi contenere tags e quindi vanno parsati (questo è anche il tipo di default).
EMPTY	Questo elemento non può contenere dati.
ANY	Questo elemento può contenere qualsiasi tipo.
#CDATA	Dati carattere (Character Data). Sono caratteri che in ogni caso il parser deve ignorare. I dati di caratteri sono costituiti da testo ordinario che può includere caratteri in genere riservati al markup. In base all'impostazione predefinita, gli elaboratori di XML presuppongono che il contenuto di un file XML sia costituito da dati di caratteri.

Mentre i dati di caratteri analizzabili sono in genere utilizzati nel contenuto di un documento XML, i dati di carattere possono essere utilizzati nel caso in cui un autore desideri includere dati che non possono essere analizzati. Per dichiarare una sezione come dati di carattere, è necessario indicare l'inizio della sezione con la sequenza `<![CDATA[` e la fine con due parentesi di chiusura `]]`. Tutti i dati che risiedono all'interno di questo insieme di marcatori verranno interpretati come semplici dati non analizzabili.

In generale gli errori che un processore XML (è un modulo software utilizzato per leggere documenti XML e verificare che soddisfino la specifica XML. La sua funzione principale consiste nell'analisi e nella convalida dei dati XML.) riporta sono di due tipi:

- *Error*: violazione di una regola della specifica; il processore notifica all'applicazione l'errore rilevato e continua sia ad analizzare il documento, sia a fornirne il contenuto all'applicazione; è quest'ultima che deve decidere il comportamento rispetto all'errore (ignorarlo, cercare di correggerlo, etc.).
- *Fatal error*: grave violazione di una regola della specifica; il processore notifica all'applicazione l'errore rilevato e può o meno continuare l'analisi del documento per cercare altri errori; non deve però continuare a fornire il contenuto del documento all'applicazione.

Ad esempio una violazione di un vincolo espresso in una DTD definita dall'utente è considerata un "error"; una violazione di una regola di buona formattazione è invece considerata un "Fatal error" (violare una regola di buona formattazione vuol dire violare un vincolo espresso nel DTD dell'XML).

DTD interne

Si definiscono DTD interne quelle DTD che vengono inserite nello stesso documento XML; ecco un esempio:

```
<?xml version="1.0"?>
<!DOCTYPE libreria [
  <!ELEMENT libreria (categorie)?>
  <!ELEMENT categorie ((gialli)?,(romanzi)?,(cultura)?)>
  <!ELEMENT gialli (libro)+>
  <!ELEMENT romanzi (libro)+>
  <!ELEMENT cultura (libro)+>
  <!ELEMENT libro (autore,casaeditrice,titolo,anno,genere,prezzo)>
    <!ELEMENT autore (#PCDATA)>
    <!ELEMENT casaeditrice (#PCDATA)>
    <!ELEMENT titolo (#PCDATA)>
    <!ELEMENT anno (#PCDATA)>
    <!ELEMENT genere (#PCDATA)>
    <!ELEMENT prezzo (#PCDATA)>
  ]>
<libreria>
  <categorie>
    <gialli>
      <libro>
        <autore>Agata Christy</autore>
        <casaeditrice>nn lo so</casaeditrice>
        <titolo>aboooooooo</titolo>
        <anno>1980</anno>
        <genere>Thriller</genere>
        <prezzo>30</prezzo>
      </libro>
    </gialli>
    <romanzi>
      <libro>
        <autore>Frewre</autore>
        <casaeditrice>Mi nonna</casaeditrice>
        <titolo>Amore folle</titolo>
        <anno>1990</anno>
        <genere>amore</genere>
        <prezzo>20</prezzo>
      </libro>
    </romanzi>
    <cultura>
      <libro>
        <autore>TRewvvc</autore>
        <casaeditrice>De Agostini</casaeditrice>
        <titolo>L'Egitto</titolo>
        <anno>2003</anno>
        <genere>geografia</genere>
        <prezzo>50</prezzo>
      </libro>
    </cultura>
  </categorie>
</libreria>
```

```

</libro>
</cultura>
</categorie>
</libreria>

```

La dichiarazione <!DOCTYPE> indica che il documento individuato dall'elemento root <libreria> segue le regole definite tra le parentesi quadre.

Questo sarà un unico documento xml (**libreria.xml**), rappresentante appunto una libreria, costituita da più categorie ed ognuna delle quali da diversi libri. Il file xml verrà visualizzato dal browser nel seguente modo:

```

<?xml version="1.0" standalone="yes"?>
<!DOCTYPE libreria (View Source for full doctype...)>
-<libreria>
  -<categoria>
    -<gialli>
      +<libro>
    </gialli>
  -<romanzi>
    +<libro>
  </romanzi>
  -<cultura>
    +<libro>
  </cultura>
</categoria>
</libreria>

```

DTD esterne

In XML è possibile definire una DTD esterna ed importarla nel proprio documento riuscendo così ad utilizzare la stessa struttura in diversi documenti.

Ad esempio il seguente file sarebbe salvato come **lib.dtd**:

```

<!ELEMENT libreria (categorie)?>
<!ELEMENT catgorie ((gialli)?,(romanzi)?,(cultura)?)>
<!ELEMENT gialli (libro)+>
<!ELEMENT romanzi (libro)+>
<!ELEMENT cultura (libro)+>
<!ELEMENT libro (autore,casaeditrice,titolo,anno,genere,prezzo)>
  <!ELEMENT autore (#PCDATA)>
  <!ELEMENT casaeditrice (#PCDATA)>
  <!ELEMENT titolo (#PCDATA)>
  <!ELEMENT anno (#PCDATA)>
  <!ELEMENT genere (#PCDATA)>
  <!ELEMENT prezzo (#PCDATA)>

```

Potremmo poi dire al sistema che utilizziamo la precedente DTD, richiamandola all'inizio del nostro documento (**libreria2.xml**) e l'output sarebbe sempre lo stesso del precedente:

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE libreria SYSTEM "lib.dtd">
...

```