

# XML

## Séance 2

### Table des matières

|                                  |   |
|----------------------------------|---|
| I. Schéma de validation.....     | 1 |
| II. Ecrire un DTD.....           | 1 |
| 1. Structure globale du DTD..... | 2 |
| 2. Les éléments.....             | 2 |

### I. Schéma de validation

On donne la « forme » du fichier XML (des différents éléments) et on fait valider automatiquement. Il existe différents types de schémas de validation (par exemple DTD, Relax NG, XSD).

Dans ce cours, nous verrons les schémas DTD (Document Type Definition).

Le but de cette séance et de la suivante est de : savoir comprendre un DTD pour un fichier donné, savoir corriger un fichier qui ne passe pas la validation DTD, pouvoir compléter un fichier XML quand le DTD est fourni.

On peut ajouter un DTD directement dans un fichier XML, ou de manière externe (dans un autre fichier). Dans ce cours, nous verrons seulement les DTD internes.

### II. Ecrire un DTD

Un DTD ressemble à cela (ici, on modélise un bloc-notes, qui contient des notes) :

```
<!DOCTYPE bloc_notes [  
  <!ELEMENT bloc_notes (note)*>  
  <!ELEMENT note (dest+, titre?, corps)>  
  <!ELEMENT dest (#PCDATA)>  
  <!ELEMENT titre (#PCDATA)>  
  <!ELEMENT corps (#PCDATA)>  
  <!ATTLIST note priorite (urgent | non-urgent) "non-urgent">  
  <!ENTITY source "Moi.">  
>
```

Un fichier XML qui correspond pourrait être celui-ci :

```
<?xml version="1.0" encoding="UTF-8"?>  
<bloc_notes>  
  <note priorite="urgent">
```

```
<dest> Le chef </dest>
<titre> Absence </titre>
<corps> Je suis malade aujourd'hui. Signé : &source; </corps>
</note>
<note priorite="non-urgent">
  <dest> Papa </dest>
  <dest> Maman </dest>
  <corps> Je suis bien arrivée à Berlin ! </corps>
</note>
</bloc_notes>
```

## 1. Structure globale du DTD

On peut voir 4 blocs dans ce DTD : DOCTYPE, ELEMENT, ATTLIST, et ENTITY. Pensez à les écrire dans cet ordre ! (en particulier ATTLIST doit être après ELEMENT)

La première ligne est le DOCTYPE qui donne le nom de la racine du fichier XML. La ligne termine par un crochet ouvrant [ (et pas par le chevron !).

La dernière ligne est la suivante : ]>.

## 2. Les éléments

Les blocs ELEMENT représentent chacun un type d'élément du fichier XML.

Un ELEMENT est soit de type #PCDATA (pour « Parsed Character Data », données en caractères traités) ou comprend un ou des sous-éléments.

Si l'élément est « simple » (c'est-à-dire n'a pas d'enfant), comme le destinataire de la note dans notre exemple, on peut le noter ainsi :

```
< !ELEMENT dest (#PCDATA) >
```

Attention, si l'élément est de type PCDATA, il faut toujours mettre un # devant, et des parenthèses.

Si l'élément possède des enfants (ou sous-éléments), on peut les noter. Par exemple, le bloc-notes contient des notes :

```
< !ELEMENT bloc_notes (notes) >
```

Par contre, si on n'ajoute rien, cette ligne signifie que l'élément bloc\_notes contient toujours uniquement un élément notes (pas plus, pas moins). Mais on veut que le bloc\_notes contienne plusieurs notes si possible. On modifie donc la ligne :

```
< !ELEMENT bloc_notes (notes*) >
```

Ajouter le signe \* à un élément signifie qu'il peut être présent 0, 1, ou plusieurs fois.

De la même manière, le signe + signifie qu'il doit être présent au moins 1 fois, et le signe ? indique qu'il peut être présent 0 ou 1 fois.

Un élément peut avoir plusieurs enfants, on les note alors à la suite, dans une même parenthèse :

```
<!ELEMENT note (date, dest+, titre?, corps)>
```

Enfin, on peut donner le choix entre deux sous-éléments (l'un ou l'autre sera présent) :

```
<!ELEMENT note (titre | corps)>
```

### Exemple :

La ligne `<!ELEMENT menu (entree?, plat, (fromage | dessert))>` signifie que l'élément menu contient 0 ou 1 élément entree, un élément plat et soit un élément fromage, soit un élément dessert.

### Exercices :

Ecrire les lignes de DTD correspondant aux éléments suivants :

- Un journal contient un titre, un ou plusieurs articles, et un éditeur.
- Un article contient un titre, peut-être un sous-titre, et au moins un auteur.
- Un auteur contient soit un nom et un prénom, soit un pseudonyme.
- Un article contient peut-être des traducteurs, et toujours une langue originale.

Correction :

```
<!ELEMENT journal (titre, article+, editeur)>
```

```
<!ELEMENT article (titre, sous-titre?, auteur+)>
```

```
<!ELEMENT auteur ((nom, prenom) | pseudonyme)>
```

```
<!ELEMENT article (traducteur*, langue)>
```