Le langage XML

eXtensible Markup Language

et

DTD

Document Type Definition

XML : **D**ocument **T**ype **D**efinition

XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée

La norme XML définit ainsi une définition de document type appelée DTD (*Document Type Definition*),

une grammaire permettant de vérifier la conformité du document XML

La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.

XML : **D**ocument **T**ype **D**efinition

- Deux fichiers d'étudiants avec leur adresse et leurs notes dans les matières diffèrent dans leurs valeurs (nom, prénom, adresse et notes) mais pas dans leur sémantique

- On a deux façons de décrire cette sémantique
  - À travers une DTD : Document Type Definition.
  - A travers un schéma : description sous forme de classes. Un fichier XML associé à ce schéma est une instance.

- Nous nous intéresserons à une représentation sous forme de *DTD*.

XML : Un exemple de DTD

Une DTD peut être définie de 2 façons :

- sous **forme interne**, c'est-à-dire en incluant la grammaire au sein même du document

- sous **forme externe**, soit en appelant un fichier contenant la grammaire à partir d'un fichier local ou bien en y accédant par son URL

Déclaration d'un élément

Pour pouvoir créer un document XML il est utile dans un premier temps de définir les éléments pouvant être utilisés, ou plus exactement les informations que l'on désire utiliser.

Pour définir un élément on utilisera la syntaxe suivante :

<! ELEMENT Nom Modèle >

XML : Modèle

Le paramètre modèle représente :

- soit un type de donnée prédéfini,
- soit une règle d'utilisation de l'élément.

Type prédéfini Description	
ANY	L'élément peut contenir tout type de données
EMPTY	L'élément ne contient pas de données spécifiques
#PCDATA	L'élément doit contenir une chaîne de caractères

Les éléments d'un arbre XML

- Un arbre XML est un ensemble de nœuds délimités par une balise ouvrante et la même balise fermante.
- Une balise ouvrante à un nom et contient de 0 à plusieurs attributs et est définie comme suit :
  <nomBalise att1= " … " … attn= " … " >.
- Une balise fermante se définit comme suit :
  </nomBalise>
- Un attribut a un nom et une valeur
- Un nœud contient de 0 à plusieurs nœuds et de 0 à plusieurs zones de textes (PCDATA) appelés nœuds textes.
- On peut définir des nœuds vides comme suit :
  <nomBalise listeAttributs/>. La liste des attributs peut être vide.

Le mot clé ELEMENT et les symboles associés

- Dans une DTD, on utilise le mot clé ELEMENT pour définir la structure d'un nœud.
- Pour décrire un nœud, on décrit ses fils de premier niveau en précisant leurs ordres d'apparition et leurs multiplicités.
- Pour la multiplicité, on utilise les symboles +, * et ?
  - ? : un seul ou pas du tout (cardinalité (0 ou 1)
  - + : un ou plusieurs
  - * : zéro ou plusieurs
- On utilise le symbole "," pour définir une séquence de type de nœuds.
- On utilise le symbole "|" pour définir un choix de nœud.
- On utilise le symbole #PCDATA pour définir un nœud texte.

XML : Résumé

Opérateur	Signification	Exemple
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A?
|	L'élément A ou l'élément B peuvent être présents	A|B
,	L'élément A doit être présent et suivi de l'élément B	A,B
()	Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs	(A,B)+

Le mot clé ATTLIST et les symboles associés

- Dans une DTD, on utilise le mot clé ATTLIST pour définir les attributs d'un nœud.
- Avec ce mot clé, on doit préciser de quel nœud il s'agit et la liste des attributs.

Déclaration des attributs

- Il est possible d'ajouter des propriétés à un élément particulier en lui affectant un attribut, c'est-à-dire une paire clé/valeur.

Ainsi avec XML la syntaxe pour définir un attribut est la suivante :

<! ATTLIST Elément Attribut Type >

Déclaration des attributs

Type représente le type de donnée de l'attribut, il en existe trois :

- **littéral**: il permet d'affecter une chaîne de caractères à un attribut. Pour déclarer un tel type il faut utiliser le mot clé *CDATA*

- **l'énumération**: cela permet de définir une liste de valeurs possibles pour un attribut donné, afin de limiter le choix de l'utilisateur. La syntaxe de ce type d'attribut est :

<! ATTLIST Elément Attribut (Valeur1 | Valeur2 | ...) >

Pour définir une valeur par défaut il suffit de faire suivre l'énumération par la valeur désirée entre guillemets :

<! ATTLIST Elément Attribut (Valeur1 | Valeur2) "valeur par défaut" >

- **atomique**: il permet de définir un identifiant unique pour chaque élément grâce au mot clé *ID*.

Déclaration des attributs

Chacun de ces types d'attributs peut être suivi d'un mot clé particulier permettant de spécifier le niveau de nécessité de l'attribut :

- **#IMPLIED** signifie que l'attribut est optionnel, c'est-à-dire non obligatoire
- **#REQUIRED** signifie que l'attribut est obligatoire
- **#FIXED** signifie que l'attribut sera affecté d'une valeur par défaut s'il n'est pas défini. Il doit être immédiatement suivi de la valeur entre guillemets

XML : Un exemple de DTD (1)

```
<!ELEMENT etudiant(ou, matiere*)>
<!ATTLIST etudiant
    nom CDATA #REQUIRED
    prenom CDATA #REQUIRED
    annee (ing1|ing2|ing3|autre) "autre" #REQUIRED
    age CDATA #IMPLIED
    >
<!ATTLIST ou
   adresse CDATA #REQUIRED
   ville CDATA #REQUIRED
   cp CDATA #REQUIRED
   >
```


XML : Un exemple de DTD (2)

```
<!ELEMENT matiere(appreciation,note*) >
<!ATTLIST matiere
    libelle CDATA #REQUIRED
    >
<!ELEMENT appreciation (#PCDATA| attention)+ >
<!ELEMENT attention (#PCDATA)>
<!ATTLIST note
    epreuve CDATA #REQUIRED
    valeur CDATA #IMPLIED
    >
```