

UE – IDC

Mémo

Présentation des DTD

XML permet d'utiliser une ressource (interne ou externe) afin de vérifier qu'une instance XML est conforme à la syntaxe qui est décrite dans cette ressource. La norme XML donne ainsi la possibilité d'utiliser un langage de définition afin de décrire cette syntaxe. Historiquement le premier langage est appelé DTD (Document Type Definition). Avec ce type de langage on définit une grammaire permettant de vérifier la conformité de l'instance XML.

NB : La norme XML n'impose pas l'utilisation d'une DTD pour une instance XML, mais elle impose par contre le respect exact d'un certain nombre de règles générales.

Ainsi on parlera de :

- document ou instance valide pour un document ou instance XML comportant à minima sa syntaxe décrite dans une ressource en DTD
- document ou instance bien formé pour un document ou instance XML ne comportant aucune ressource mais répondant aux règles de base du XML

Une DTD peut être définie de 2 façons

- sous forme interne, c'est-à-dire en incluant la grammaire au sein même de l'instance
- sous forme externe, soit via l'adresse locale de la ressource (fichier local) ou son URL

Pour pouvoir créer une instance 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.

Déclarer un élément :

Pour définir un élément on utilisera la syntaxe suivante :

`<! ELEMENT Nom Modèle >`

Le paramètre modèle représente soit un type de donnée prédéfini, soit une règle d'utilisation de l'élément.

Les types prédéfinis utilisables sont les suivants :

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

Ainsi un élément nommé Nom contenant un type #PCDATA sera déclaré de la façon suivante dans la DTD :

`<! ELEMENT Nom (#PCDATA) >`

NB : Le mot clé #PCDATA doit nécessairement être écrit entre parenthèses, au risque sinon d'obtenir une erreur du parseur !

Cet élément pourra être écrit de la façon suivante dans l'instance XML :

`<Nom>Caius Julius Caesar</Nom>`

D'autre part, il est possible de définir des règles d'utilisation, c'est-à-dire les éléments XML qu'un élément peut ou doit contenir. Par défaut, l'élément doit être présent 1 et une seule fois. Cette syntaxe se fait à l'aide de notations

spécifiques dont voici un récapitulatif (opérateur, signification, exemple) :

+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent 0 ou plusieurs fois	A*
?	L'élément peut être présent 0 ou 1 fois au plus	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)+

Déclarer 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 Élément Attribut Type >
```

Type représente le type de donnée de l'attribut, il en existe 4 :

littéral : il permet d'affecter une chaîne de caractères à un attribut. Pour déclarer un tel type il faut utiliser le mot clé CDATA. Un attribut de type CDATA peut contenir n'importe quel caractère, s'il respecte les contraintes de forme.

littéral contraint : Les attributs de type NMTOKEN ne peuvent contenir que des lettres, des chiffres, un point [.], un tiret [-], un trait de soulignement [_] et un deux-points [:]. Les attributs de type NMTOKENS peuvent contenir les mêmes caractères que les attributs de type NMTOKEN, plus des espaces blancs. Par espace blanc, on entend un ou plusieurs espaces, retours chariot, sauts de ligne ou tabulations.

énuméré : 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 Élément Attribut (Valeur1 | Valeur2 | ... ) >
```

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

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

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

Enfin chacun de ces types d'attribut 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