

XML

DTD : Document Type Definition

Qu'est qu'une DTD ?

- ▶ Permet de vérifier qu'un document XML est conforme à une syntaxe donnée (à une grammaire)
- ▶ La vérification de la conformité d'un document XML se fait dans une DTD (Document Type Definition)
- ▶ La norme XML n'impose pas l'utilisation de DTD mais elle impose le respect exact des règles de base de la norme XML
- ▶ On distingue 2 types de conformité
 - ▶ Les documents valides : les documents XML avec une DTD
 - ▶ Les documents bien formés : les documents XML ne comportant pas de DTD mais répondant aux règles de base du XML
- ▶ Une DTD peut être définie de 2 façons
 - ▶ Sous forme interne, incluant la grammaire dans le 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

Exemple de DTD

- ▶ Définissons la conformité d'un fichier XML qui contient des informations concernant une personne
 - ▶ Nécessaire d'avoir au moins les 3 informations suivantes dans un élément personne:
 - ▶ Nom
 - ▶ Prénom
 - ▶ Téléphone
 - ▶ L'élément email est optionnel

```
<!ELEMENT personne (nom,prenom,telephone),email? >  
<!ELEMENT nom (#PCDATA) >  
<!ELEMENT prenom (#PCDATA) >  
<!ELEMENT telephone (#PCDATA) >  
<!ELEMENT email (#PCDATA) >
```

Explication de la syntaxe

- ▶ Déclaration d'un élément
 - ▶ `<! ELEMENT nom modèle>`
- ▶ Le paramètre modèle représente soit un type de données prédéfinies, soit une règle d'utilisation de l'élément
- ▶ Les types prédéfinis utilisables sont les suivants :
 - ▶ ANY : l'élément peut contenir tout type de données. A utiliser avec précaution car il supprime quasiment tout contrôle de validité
 - ▶ EMPTY : l'élément ne contient pas de données spécifiques
 - ▶ #PCDATA : l'élément doit contenir une chaîne de caractère
- ▶ Un élément dont son nom est « produit » et qui ne doit contenir que du #PCDATA, sera défini de la façon suivante :
 - ▶ `<!ELEMENT produit (#PCDATA)>`

Règles d'utilisation d'un élément

- ▶ Un élément doit ou peut contenir du #PCDATA. Cette capacité s'exprime avec les notations spécifiques suivantes :
 - ▶ A^+ : au moins un élément A
 - ▶ A^* : l'élément A peut être présent plusieurs fois ou aucune
 - ▶ $A?$: au plus un élément A
 - ▶ $A|B$: l'élément A ou l'élément B peuvent être présent
 - ▶ A,B : l'élément A doit être présent et suivi de l'élément B
 - ▶ $()$: permet de regrouper des éléments pour leur appliquer le même opérateur
 - ▶ Exemple : $(A,B)^+$ l'élément A puis B doivent être présent au moins une fois
 - ▶ $,$: les éléments enfants doivent figurer dans l'ordre spécifié

Contenu mixte

- ▶ Si un élément peut contenir à la fois du #PCDATA et d'autres éléments
- ▶ Le mot-clé #PCDATA doit être en début de la liste des éléments
`<!ELEMENT personne (#PCDATA,nom,prenom,telephone)>`
- ▶ Les modèles de contenu mixte n'accepte ni de suites d'éléments enfants, ni de choix d'opérateurs de cardinalité. L'ordre utilisé n'a donc pas d'importance sur les éléments suivants #PCDATA

Exemple

- ▶ Un élément NomPersonne est composé
 - ▶ Soit d'un sigle M, Mme, Mlle
 - ▶ D'un prénom
 - ▶ D'un 2ième prénom
 - ▶ Et d'un nom de famille
- ▶ Ce qui donne

```
<!ELEMENT NomPersonne ( ( M | Mme | Mlle), Prenom, Prenom2, Nom ) >
<!ELEMENT M EMPTY>
<!ELEMENT Mme EMPTY>
<!ELEMENT Mlle EMPTY>
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT prenom2 (#PCDATA) >
<!ELEMENT nom (#PCDATA) >
```
- ▶ Le document suivant est donc conforme

```
<NomPersonne>
  <M/>
  <Prenom>John</Prenom>
  <Prenom2>Edouard</Prenom2>
  <Nom>Martin</Nom>
</NomPersonne>
```

Exercice

- ▶ On impose maintenant que l'élément Prenom2 soit optionnel, que l'on puisse en avoir plusieurs et qu'au plus un élément M, Mme ou bien Mlle soient présents (solution Fichier Exo1.dtd)
- ▶ Rédiger une DTD pour une bibliographie (Solution Exo2.dtd). Cette bibliographie :
 - ▶ contient des livres et des articles
 - ▶ les informations nécessaires pour un livre sont :
 - ▶ son titre général
 - ▶ les noms des auteurs
 - ▶ ses tomes et pour chaque tome, leur nombre de pages
 - ▶ des informations générales sur son édition comme par exemple le nom de l'éditeur, le lieu d'édition, le lieu d'impression, son numéro ISBN
 - ▶ les informations nécessaires pour un article sont :
 - ▶ son titre
 - ▶ les noms des auteurs ;
 - ▶ ses références de publication : nom du journal, numéro des pages, année de publication et numéro du journal
 - ▶ on réservera aussi un champ optionnel pour un avis personnel.

Déclarations d'attributs (1)

- ▶ La DTD permette de déclarer des attributs à des éléments
- ▶ On utilise pour cela une déclaration de liste d'attributs
`<!ATTLIST nomElement nomAttr typeAttr attrParDéfaut valeurParDéfaut>`
- ▶ `nomElement` désigne le nom de l'élément auquel l'attribut de nom `nomAttr` doit être associé
- ▶ `TypeAttr` définit les types d'attribut qui sont au nombre de 10 dans la recommandation XML 1.0. Les principaux types d'attribut sont
 - ▶ CDATA pour des données textuelles
 - ▶ Valeur énumérée, liste de choix. *Exemple* (M | Mme | Mlle)
 - ▶ ID sert d'identifiant unique pour cette instance d'élément. 2 attributs de type ID ne peuvent donc pas avoir la même valeur dans un même document XML

Déclarations d'attributs (2)

- ▶ Le paramètre `attrParDéfaut` définit le comportement par défaut du parseur. Il indique si la présence de l'attribut est obligatoire, optionnelle et comment il faut réagir en fonction de l'absence de l'attribut
 - ▶ `#REQUIRED` : l'attribut doit figurer dans chaque instance de l'élément

```
<!ATTLIST Livre titre #REQUIRED >
```
 - ▶ `#IMPLIED` : l'attribut est facultatif

```
<!ATTLIST Article avis CDATA #IMPLIED >
```
 - ▶ `#FIXED "valeur par défaut"` : l'attribut peut figurer ou non dans le document. S'il est présent, il doit alors avoir la valeur "toto" sinon le parseur peut fournir la valeur par défaut

```
<!ATTLIST Livre editeur #FIXED "Edit-Corp">
```
 - ▶ Une valeur par défaut : l'attribut peut apparaître ou non dans le document. S'il est absent, le parseur peut fournir la valeur par défaut. Si une valeur est définie, il peut être n'importe quelle valeur figurant dans la liste `ATTLIST`

Exercice

- ▶ Modifier la DTD précédente... (solution : Exo3.dtd)
 - ▶ en ajoutant un attribut optionnel soustitre à l'élément titre ;
 - ▶ en faisant de l'élément tome un élément vide et en lui ajoutant un attribut requis nb_pages et un attribut optionnel soustitre ;
 - ▶ en faisant de l'élément nom_journal un attribut de l'élément journal et en lui donnant comme valeur par défaut Feuille de Chou ;
 - ▶ en faisant de l'élément annee un attribut de type énuméré, prenant comme valeurs possibles 2002, 2003, 2004, "avant_2002" et "inconnue" et proposant comme valeur par défaut inconnue.

Déclarations d'entités (1)

- ▶ Permet de disposer de l'équivalent de raccourcis clavier et de caractères à priori non accessibles par le jeu de caractères sélectionné
- ▶ Entités paramétriques
 - ▶ Définit des symboles qui seront utilisés ailleurs dans la DTD

```
<!ENTITY %heading "H1 | H2 | H3 | H4 | H5 | H6" >
```
- ▶ Entités analysables externes
 - ▶ Permet d'enregistrer dans un fichier texte externe les références

```
<!ENTITY texte SYSTEM "file:///chemin/fichier.txt" >
```
 - ▶ Inclusion de fichier dans le document XML en utilisant la référence &texte dans notre exemple

Déclarations d'entités (2)

- ▶ Entités analysables internes

- ▶ Donne un nom facilement lisible à des caractères ne figurant pas dans notre alphabet

```
<!ENTITY nbsp " ">  
<!ENTITY eacute "é">
```

- ▶ Ils peuvent être utilisées dans un document XML référençant cette DTD à l'aide de la notation `&NomEntité`

```
<!ENTITY copyright "©">  
<!ENTITY copy "copyright &copyright; 2004 Edit-Corp,  
Tous droits réservés">
```

Déclaration d'une DTD

- ▶ Dans un document XML, dans le cas de l'utilisation d'une DTD externe, on doit alors avoir :
 - ▶ standalone="no"
 - ▶ Puis l'élément `<!DOCTYPE elt_racine SYSTEM "filename.dtd">`

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE carnet SYSTEM "../Cours5-Solution/Ex01.dtd">
<carnet>
  <NomPersonne>
    <M/>
    <Prenom>George</Prenom>
    <Nom>FILOCHE</Nom>
  </NomPersonne>
  <NomPersonne>
    <Mlle/>
    <Prenom>Martine</Prenom>
    <Prenom2>Yvonne</Prenom2>
    <Nom>GETUDAVE</Nom>
  </NomPersonne>
</carnet>
```

Déclaration d'une DTD (2)

```
<!DOCTYPE elt_racine SYSTEM|PUBLIC emplacement1 emplacement2>
```

- ▶ **SYSTEM** s'utilise avec une DTD externe dont l'emplacement est
 - ▶ Soit une URL `http://www.....`
 - ▶ Ne pas utiliser le caractère `#` dans une URL
 - ▶ Soit une URI (Uniform resource Identifier)
 - ▶ `file:///home/user/DTD/.....`
- ▶ **PUBLIC** permet l'utilisation d'une référence générique à la DTD par l'intermédiaire d'un URI, voire un second URI
 - ▶ Plutôt utiliser avec des DTD normalisées disponibles au public
 - ▶ Exemple : validité d'un document HTML4

```
<!doctype html public '-//W3C//DTD HTML 4.0//EN'  
'http://www.w3.org/TR/REC-html40/strict.dtd'>
```