

DTD (Document Type Définition)

Cours DAX pour Master-2 option TI

DTD (Document Type Définition)

Comment définir une DTD : éléments et imbrication, attributs, entités.

Définition

DTD (*Document Type Definition*), est une Grammaire permettant de vérifier la conformité du document XML;

- Elle est constituée d'une liste de déclarations qui définissent :
 - Les éléments qui peuvent apparaître dans le contenu d'un élément;
 - L'ordre de ces éléments;
 - La présence de texte brut;
 - Les attributs autorisés et les attributs obligatoires, pour chaque élément;
 - Les entités et les déclarations de notations;
 - Les commentaires;
- Elle doit être déclarée au début, avant l'*élément racine*;
- Simple a utilisée mais limitée;
- La DTD peut être **interne**, **externe** ou **mixte**.

NB : On parlera de **document valide** pour un document XML comportant une DTD.

Déclaration de la DTD

La déclaration de la DTD du document doit être placée dans le prologue. La DTD peut être interne, externe ou mixte.

DTD interne

La DTD est incluse dans le document, son contenu est encadré par des crochets ' [' et '] ' . elle est déclarée comme suit :

<!DOCTYPE Nom-Racine [Liste-Déclarations]>

Exemple :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- DTD -->
<!DOCTYPE Personne[
    <!--début de la DTD interne -->
    <!ELEMENT Personne (Nom,Prenom)>
    <!ELEMENT Nom (#PCDATA)>
    <!ELEMENT Prenom (#PCDATA)>
    <!--fin de la DTD interne -->
]>
<Personne>
    <Nom>Toto</Nom>
    <Prenom>Tutu</Prenom>
</Personne>
```

La DTD déclare un élément racine **Personne**, qui contient deux éléments **Nom** et **Prenom** qui contiennent du texte (*Parsed Characters DATA*).

DTD (Document Type Définition)

DTD externe

La DTD est contenue dans un autre fichier dont l'extension est **.dtd**. Le document XML se contente alors de donner l'adresse de la DTD.

L'adresse de de la DTD peut être donnée explicitement par une URL ou par un FPI (*Formal Public Identifier*);

- La référence à une URL est introduite par le mot clé **SYSTEM** suivi de l'URL;

<!DOCTYPE Nom-Racine SYSTEM "url">

- L'URL peut être une URL complète commençant par http:// ou ftp://

Exemple :

<!DOCTYPE html SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- Ou tout simplement le nom d'un fichier local.

Exemple :

<!DOCTYPE personne SYSTEM "personnes.dtd">

- La référence à un FPI est introduite par le mot clé **PUBLIC** suivi du FPI et d'une URL;

<!DOCTYPE Nom-Racine PUBLIC "fpi" "url">

Exemple :

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

DTD mixte

Il est possible d'avoir simultanément une DTD externe adressée par URL ou FPI et des déclarations internes. La syntaxe est comme suit :

<!DOCTYPE Nom-Racine SYSTEM "url" [Liste-Déclarations]>

- Le document devra être valide par rapport à une DTD égale à la réunion de la DTD externe et de la DTD interne;
- La partie interne de la DTD surcharge la partie externe;
- tout élément doit être déclaré dans la partie interne ou externe mais pas dans les deux;
- Il n'est pas possible de redéfinir un élément;
- il est possible de déclarer plusieurs fois le même attribut ou entité paramètre, la première déclaration a priorité.

DTD (Document Type Définition)

Exemple

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE Personnes SYSTEM "Personnes.dtd" [
  <!ATTLIST Personne age CDATA #REQUIRED>
]>
<Personnes>
  <Personne age="30">
    <Nom>Pitchou</Nom>
    <Prenom>Adel</Prenom>
  </Personne>
  <Personne age="26">
    <Nom>Tchoumpouh</Nom>
    <Prenom>Aymen</Prenom>
  </Personne>
</Personnes>
```

```
<!--DTD externe -->
<!ELEMENT Personnes (Personne*)>
<!ELEMENT Personne (Nom,Prenom)>
<!ATTLIST Personne age CDATA #IMPLIED>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
```

Déclaration d'élément

Un élément se déclare selon la syntaxe suivante :

<!ELEMENT Nom-Element Modèle-Contenu>

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

Les types prédéfinis

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)>

Exemple

```
<Nom>TOTO</Nom>
```

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

DTD (Document Type Définition)

Les règles d'utilisation

Les règles d'utilisation d'éléments permettent de combiner des sous éléments comme suit :

Opérateur		Signification	Exemple
Séquence	,	L'élément A doit être présent et suivi de l'élément B	(A,B)
	()	Permettent de regrouper des éléments	
Alternative		L'élément A ou l'élément B peuvent être présents	(A B)
Indicateurs d'occurrence	*	Élément présent plusieurs fois ou aucune (0-n)	A*
	+	Élément présent au moins une fois (1-n)	A+
	?	Élément optionnel (0-1)	A ?

Exemple :

```
<!-- DTD -->
<!ELEMENT Personne (nom,prenom,telephone,email?)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

<!-- Document XML-->

```
<personne>
  <nom>Trinita</nom>
  <prenom>Rassime</prenom>
  <telephone>0566778899</telephone>
  <email>RTrinita@umbb.dz</email>
</personne>
```

Ou bien

```
<personne>
  <nom>Trinita</nom>
  <prenom>Rassime</prenom>
  <telephone>0566778899</telephone>
</personne>
```

Déclarer des attributs

La syntaxe pour définir des attributs est la suivante :

```
<!ATTLIST Nom-Elem Nom-Attr Type-Attr Decl-Default
              Nom-Attr Type-Attr Decl-Default
              ...
>
```

- **Nom-Elem** : Le nom de l'élément auquel est rattaché l'attribut;
- **Nom-Attr** : Le nom de l'attribut;
- **Type-Attr** : Type de l'attribut. Les types prédéfinis utilisables sont les suivants :
 - **Littéral** : Affectation d'une chaîne de caractères à un attribut. Mot clé **CDATA**;
<!ATTLIST Nom-Element Nom-Attribut CDATA >
 - **l'énumération** : Permet de définir une liste de valeurs possibles pour un attribut;
<!ATTLIST Nom-Element Nom-Attribut (Val1,Val2,...,Valn)>

DTD (Document Type Définition)

- **atomique** : Définition d'un identifiant unique. Mot clé **ID**.
<!ATTLIST Nom-Element Nom-Attribut ID>
 - ✓ **ID** : Unique dans un document, il y a au plus un attribut de type ID par élément ;
 - ✓ **IDREF** ou **IDREFS** : identificateur ou suite d'identificateurs, il doit être égal, dans le document XML, à l'identifiant d'un autre élément;
 - ✓ **NMTOKEN** ou **NMTOKENS** : token de nom ou suite de tokens de nom, (nom1 ... nomn)

Exemple :

```
<!--DTD-->
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE DOCUMENT [
<!ELEMENT DOCUMENT(PERSONNE*)>
<!ELEMENT PERSONNE (#PCDATA)>
<!ATTLIST PERSONNE PNUM ID #REQUIRED>
<!ATTLIST PERSONNE MERE IDREF #IMPLIED>
<!ATTLIST PERSONNE PERE IDREF #IMPLIED>
]>

<!--Documeent XML-->
<DOCUMENT>
  <PERSONNE PNUM = "P1">Maman</PERSONNE>
  <PERSONNE PNUM = "P2">Papa</PERSONNE>
  <PERSONNE PNUM = "P3" MERE="P1" PERE="P2">Pitchou</PERSONNE>
  <PERSONNE PNUM = "P4" MERE="P1" PERE="P2">Trinita</PERSONNE>
</DOCUMENT>
```

- **Decl-Defaut** : Les types d'attributs peuvent être suivis d'un mot clé particulier permettant de spécifier le statut de l'attribut et sa valeur par défaut :
 - **#IMPLIED** : Attribut facultatif ;
 - **#REQUIRED** : Attribut obligatoire;
 - **#FIXED (suivi d'une valeur)** : L'attribut lui sera affecté une valeur par défaut s'il n'est pas défini. Il doit être immédiatement suivi de la valeur entre guillemets

Exemple :

```
<! ATTLIST disque IDDisk ID #REQUIRED
                Type(K7|MiniDisc|Vinyl|CD)"CD"
>
```

On affecte à l'élément disque deux attributs :

- **IDDisk**, de type atomique (identifiant unique) obligatoire;
- **Type**, de type énumération il peut être soit *K7*, *MiniDisc*, *Vinyl* ou *CD*, et le dernier sera affecté par défaut.

NB : l'ordre des attributs n'est pas important.

DTD (Document Type Définition)

Déclarer des entités

XML permet de déclarer un groupe d'éléments sous un nom, afin de ne pas avoir à réécrire ces derniers plusieurs fois dans la DTD s'ils se répètent, l'avantage:

- une meilleure lisibilité
- un contrôle accru sur le contenu
- une plus grande facilité de mise à jour

On distingue plusieurs types d'entités dans XML :

- les entités générales
- les entités paramètres
- les entités caractères

Entités générales

Les entités générales permettent de définir des éléments pouvant être substitués dans le corps du document XML. La syntaxe d'une entité générale est la suivante :

<!ENTITY Nom-Entité "Contenu de l'entité">

Les entités définies dans la DTD peuvent être utilisées dans le code XML en les appelant avec la syntaxe suivante :

&Nom-Entité;

Exemple : **<!ENTITY site "UMBB.dz">**

<!--Document XML-->

```
<personne>
  <nom>Pitchou</nom>
  <prenom>Aymen</prenom>
  <site>&site;</site>
  <email>webmaster@&site;</email>
</personne>
```

Résultat

```
<personne>
  <nom>Pitchou</nom>
  <prenom>Aymen</prenom>
  <site>UMBB.dz</site>
  <email>webmaster@UMBB.dz</email>
</personne>
```

Les entités générales peuvent être contenues dans un fichier extérieur, on parle alors d'*entité externe*.

<!ENTITY Nom-Entité SYSTEM "Nom-Fichier">

Exemple : **<!ENTITY test SYSTEM "test.txt">**

Le contenu du fichier **test.txt** sera inséré dans le fichier XML à chaque fois que l'entité **&test;** sera rencontrée

DTD (Document Type Définition)

Entités paramètres

Elles permettent d'utiliser des entités dans les DTD elles-mêmes, Le nom doit :

- Commencer par une lettre ou un underscore (_)
- Etre composé uniquement de lettres, chiffres, tirets (-), underscore (_), points (.) ou du caractère 'deux points' (:)

<!ENTITY % Nom-Entité "définition">

Exemple :

<!ENTITY % support #REQUIRED "type(K7|MiniDisc|Vinyl|CD) 'CD'">

Il est possible de définir une entité paramètre externe.

<!ENTITY % Nom-Entité SYSTEM "fichier.dtd">

Entités caractères

Les entités caractères sont des caractères réservés du XML représentés sous forme d'entités générales afin de pouvoir insérer ces caractères réservés dans le document XML.

Voici la liste des entités caractères principales :

Entité caractère	Représentation
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

Il est également possible de définir des entités caractères pour n'importe quel caractère en utilisant le code hexadécimal du caractère :

<!ENTITY Nom-Entité "ODEHEXA;">

Exemple : **<!ENTITY ccedille "ç">**

Déclarer des notations

XML permet de définir une application par défaut à lancer pour ouvrir des documents non XML encapsulés dans le code XML.

Exemple : **<!NOTATION jpg SYSTEM "psp.exe">**

Associer les images JPG au programme Paint Shop Pro (*psp.exe*)