

# Module: Données Semi Structurées

Cours3 : La DTD (Document Type Definition)

# DTD : une grammaire qui définit et limite la structure du document

Les DTD assurent l'homogénéité des documents

Présentation

- pas de balise arbitraire
- pas d'information dans le mauvais ordre

Echange de données entre applications

- être sûr que le document sera reconnu par le logiciel de traitement

# La DTD : Définition d'un type de document

➤ Un document XML peut posséder une Déclaration formelle de sa structure, une sorte de grammaire (avec des règles d'utilisation) que doit respecter le document.

➤ La DTD de l'annuaire :

```
<! ELEMENT Annuaire ( personne)* >
```

```
<! ELEMENT personne ( nom, prénom, bureau,  
    poste, email?) >
```

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

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

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

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

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

(PCDATA: Parsed Character DATA , chaîne de caractères analysée)

# Documents générés

```
<Annuaire>
  <personne>
    <nom> Drici </nom>
    <prénom> Morad </prenom>
    <bureau> B520 </bureau>
    <poste> 4823 </poste>
    <email> MDrici@yahoo.fr
      </email>
  </personne>
</Annuaire>
```

```
<Annuaire>
  <personne>
    <nom> Yahyaoui </nom>
    <prénom> Rida </prenom>
    <bureau> B520 </bureau>
    <poste> 4823 </poste>
  </personne>
</Annuaire>
```

# Les règles d'utilisation DTD

## notations

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 <b>ou</b> B peuvent être présents (pas les deux)	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)+

# Exemple commenté

- `< ! ELEMENT carnet ( fiche *) >`

L'élément carnet peut contenir un nombre arbitraire d'éléments fiche.

- `< ! ELEMENT fiche ( nom, prénom, téléphone, email*)`

L'élément fiche contient exactement un nom , un prénom, un téléphone, et autant de email qu'on le souhaite

- `< ! ELEMENT nom (# PCDATA)>`

L'élément nom ne contient que du texte

- `< ! ELEMENT téléphone (fixe| fax| portable) *>`

L'élément téléphone contient dans n'importe quel ordre et en n'importe quelle quantité des éléments: fixe, fax et portable.

# DTD associé à un sommaire

```
< ! ELEMENT Sommaire ( Numéro, Date, RubriqueSommaire
+)>
  < ! ELEMENT numéro (# PCDATA)>
  < ! ELEMENT date(# PCDATA)>
  < ! ELEMENT RubriqueSommaire( RubriqueTitre,
ArticleSommaire+)>
    < ! ELEMENT RubriqueTitre(# PCDATA)>
    <! ELEMENT ArticleSommaire(ArticleTitre, ArticlePage)>
      < ! ELEMENT ArticleTitre (# PCDATA)>
      < ! ELEMENT ArticlePage (# PCDATA)>
```

# Validité d'un document

Un document XML qui possède une DTD/schémaXML associée est dit **valide** si:

1. Il respecte les règles de son grammaire (DTD/Schéma XML).
2. Il est bien formé.

Exemple:

- Toutes les balises ouvrantes ont une balise fermante associée.
- Les balises sont correctement imbriquées.

# Deux types de DTD

**Sous forme interne:** la grammaire du document est plac   au sein m  me du document.

**Sous forme externe:** un fichier ext  rieur contient la grammaire du document .

# Doc avec DTD interne

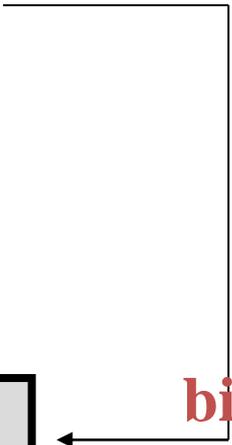
```
<?xml version = « 1.0 » ? Stalone = yes >
<! DOCTYPE bibliothèque
  [ <!ELEMENT bibliothèque ( livre+)>
    <!ELEMENT livre (Titre, auteur, ref)>
    <!ELEMENT Titre(# PCDATA)>
    <!ELEMENT auteur (# PCDATA)>
    <!ELEMENT ref (# PCDATA)>
  ]>
<bibliothèque>
<livre>
  <titre > Dune </titre>
  <auteur> Frank Heckbert </auteur>
  <ref> Fiction-H-1</ref>
</livre>
</bibliothèque>
```

# Doc avec DTD externe

```
<?xml version = «1.0» ?>  
<! DOCTYPE bibliothèque SYSTEM «biblio.dtd»>  
< bibliothèque>  
  <livre>  
    <titre> Dune </titre>  
    <auteur> Frank Heckbert </auteur>  
    <ref> Fiction-H-1 </ref>  
  </livre>  
< / bibliothèque>
```

```
< ! ELEMENT bibliothèque ( livre+)>  
< ! ELEMENT livre (Titre, auteur, ref)>  
< ! ELEMENT Titre( #PCDATA)>  
< ! ELEMENT auteur ( #PCDATA)>  
< ! ELEMENT ref ( #PCDATA)>
```

**biblio.dtd**



# Exemple (XML) disques.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE disques SYSTEM "DisquesML.dtd">
<disques>
  <groupe >
    <nom>MUSE</nom>
    <membre>Matthew Bellamy</membre>
    <membre>Dominic Howard</membre>
    <membre>Chris Wolstenholme</membre>
  </groupe>
  <disque>
    <interprete > TOTO </interprete>
    <titre>Showbiz</titre>
  </disque>
  <disque>
    <interprete> RIRI </interprete>
    <titre>Origin of symmetry</titre>
  </disque>
</disques>
```

# Exemple (DTD)

DisquesML.dtd:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

# DTD INTERNE

DTD interne :

1 <?xml version="1.0" encoding="ISO-8859-1"?>

2 <!DOCTYPE carnet [

3 <!-- contenu de la DTD -->

4 ]>

le nom qui suit DOCTYPE est celui de l'élément racine

# Décrire une conversation

Une DTD pour définir la structure d'un document XML, valide pour une conversation.

```
[<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE log [
  <!ELEMENT log ( message*)>
  <!ELEMENT message (From, To, Text)>
  <!ELEMENT From (User)>
  <!ELEMENT To (User)>
  <!ELEMENT User( #PCDATA) >
  <!ELEMENT Text ( #PCDATA) >
]>
```

# Exercice: donner le dtd externe correspondant?

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Contacts>
```

```
  <Person>
```

```
    <Firstname>John</Firstname>
```

```
    <Lastname>Smith</Lastname>
```

```
    <Position>CEO</Position>
```

```
  </Person>
```

```
  <Person>
```

```
    <Firstname>Tom</Firstname>
```

```
    <Lastname>Dunne</Lastname>
```

```
    <Company>Today FM</Company>
```

```
    <Position>DOI</Position>
```

```
    <Email>tom.dunne@todayfm.com</Email>
```

```
  </Person>
```

```
</Contacts>
```

# Déclaration d'attributs

<! ATTLIST *balise* *Attribut* *Type* *Mode* >

*balise* spécifie l'élément auquel est attaché l'attribut

*Attribut* est le nom de l'attribut déclaré

*Type* définit le type de donnée de l'attribut choisi parmi:

CDATA

Chaînes de caractères entre guillemets ("aa") non analysées

Enumération

Liste de valeurs séparées par |

<! ATTLIST *balise* *Attribut* (Valeur1 | Valeur2 | ... ) >

ID et IDREF

Clé et référence à clé

*Mode* précise le caractère obligatoire ou non de l'attribut

#REQUIRED, #IMPLIED ou #FIXED

# Exemple d'attributs

```
<! ATTLIST personne  
    num ID,  
    age CDATA,  
    genre (Masculin | Feminin ) >
```

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

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

```
<!ATTLIST auteur  
    genre (Masculin | Feminin )  
#REQUIRED  
    ville CDATA #IMPLIED>
```

```
<!ATTLIST editeur  
    ville CDATA #FIXED "Paris">
```

# Exemple de DTD

```
<!ELEMENT doc (livre* | article+) >  
<!ELEMENT livre (titre, auteur+) >  
<!ELEMENT article (titre, auteur*) >  
<!ELEMENT titre(#PCDATA) >  
<!ELEMENT auteur(nom, adresse) >  
<!ATTLIST auteur id ID #REQUIRED >  
<!ELEMENT nom(prenom?, nomfamille) >  
<!ELEMENT prenom (#PCDATA) >  
<!ELEMENT nomfamille (#PCDATA) >  
<!ELEMENT adresse (ANY) >
```

# Exemple de DTD interne

```
<?XML version="1.0" standalone="yes"?>
<!DOCTYPE CATALOGUE [
  <!ELEMENT CATALOGUE (Produit+)>
  <!ELEMENT Produit (SPECIFICATION+, ANNEE, PRIX)>
  <!ATTLIST Produit NOM CDATA #REQUIRED>
  <!ELEMENT SPECIFICATION EMPTY>
  <!ATTLIST SPECIFICATION PROVENANCE CDATA #REQUIRED COULEUR CDATA
  #REQUIRED>
  <!ELEMENT ANNEE (#PCDATA)>
  <!ELEMENT PRIX (#PCDATA)> ]>
<CATALOGUE>
  <Produit NOM=« Table">
    <SPECIFICATION PROVENANCE=« Oran" COULEUR=« Blue"/>
    <ANNEE>2009</ANNEE>
    <PRIX>2500 DA</PRIX>
  </Produit>.....
</CATALOGUE>
```

# Exemple de ID et IDREF

```
<?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>
]>
<DOCUMENT>
  <PERSONNE PNUM = "P1">Ali</PERSONNE>
  <PERSONNE PNUM = "P2">Samira</PERSONNE>
  <PERSONNE PNUM = "P4" MERE="P1" PERE="P2">Ramzi</PERSONNE>
</DOCUMENT>
```

# Déclaration d'attributs

Les entités sont utilisées pour définir des raccourcis à des symboles spéciaux ou de simple standard texte.

- Déclaration:

Syntaxe:

```
<!ENTITY entity-name "entity-value">
```

Exemple:

DTD Example:

```
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

```
<author>&copyright;</author>
```

# DTD: quelques règles d'écriture

## Modularité

définir dans des entités séparées les parties réutilisables

## Précédence

Regrouper les déclarations d'entités en tête

## Abstraction

Utiliser des entités pour les modèles de contenus

## Spécificité

Éviter les DTD trop générales

## Simplicité

Découper les DTD trop complexes

# Insuffisance des DTD

Questions ?

Pas de types de données

difficile à interpréter

difficile à traduire en schéma

objets

Pas en XML

langage spécifique

Propositions de compléments

XML-data de Microsoft (BizTalk)

XML-schema du W3C

