



## II- Validation d'un document XML DTD (*Document Type Definition*)

# Validation d'un document XML par une DTD

- Document XML valide
- Types de DTD (interne, externe)
- Déclarations d'éléments
- Déclaration d'attributs
- Déclarations d'entités

# Document XML valide, DTD

- Une DTD (*définition du type de document*) structure un document XML. On dit qu'elle **valide** un document XML.
- Dans un document XML **valide** apparaît une déclaration du type de document (**DTD: définition du type de document**).
- Cette déclaration définit:
  - le nom des éléments, leur contenu, le nombre de fois et l'ordre d'apparition,
  - les attributs éventuels et leurs valeurs par défaut,
  - les noms des entités qui peuvent être utilisées.
- Les documents XML valides doivent respecter les règles données d'une DTD.

# Types de DTD (interne, externe)

- La déclaration d'une **DTD** doit apparaître après la déclaration XML, mais avant l'élément racine.

```
<!xml version="1.0" ....>
```

```
<!DOCTYPE élément_racine ....>
```

- La déclaration de la DTD peut contenir :
  - la DTD elle-même à l'intérieur du fichier XML (**DTD interne**)
  - ou une adresse URL qui indique le fichier contenant la DTD (**DTD externe**).
  - En cas de conflit, les déclarations de la DTD interne prime.

# Types de DTD (interne, externe)

- Une déclaration d'une DTD commence par :  
**<!DOCTYPE ElementRacine**  
et se termine par **>**
- Si la **DTD** est **interne** elle est entre crochets après  
**<!DOCTYPE [ ..... ]>**
- Si la **DTD** est **externe**, la déclaration XML doit fixer la valeur de l'attribut **standalone** à la valeur **"no"**

**<?xml version = "1.0 " standalone= "no " ?>**

# DTD externe

**DTD externe**: 2 types

➤ 1- DTD privée:

```
<!DOCTYPE ElementRacine SYSTEM "URLDeLaDTD">
```

exemples:

```
<!DOCTYPE CATALOGUE SYSTEM "catcd.dtd">
```

```
<!DOCTYPE CATALOGUE SYSTEM  
"http://www.fsr.ac.ma/dtd/catcd.dtd">
```

➤ 2- DTD publique:

```
<!DOCTYPE ElementRacine PUBLIC "NomDeLaDTD" "URLDeLaDTD">
```

Le nom de la **DTD** appelée **URN** (*Universal Resource Name*) doit avoir la forme:

```
-//W3C//DTD catalogue //FR
```

Le nom du propriétaire suivie du type de document, suivi de la langue

# Exemples: association DTD/document XML

- Hello XML sans DTD

```
<?xml version="1.0" standalone="yes"?>
```

```
<hello> Hello XML ! </hello>
```

- Hello XML avec DTD interne

```
<?xml version="1.0" standalone="yes"?>
```

```
<!DOCTYPE hello [
```

```
<!ELEMENT hello (#PCDATA)> ]>
```

```
<hello> Hello XML ! </hello>
```

- Hello XML avec DTD externe

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

```
<!DOCTYPE hello SYSTEM "hello.dtd">
```

```
<hello> Hello XML ! </hello>
```

# DTD, Déclaration des éléments

- Une déclaration d'un élément fournit le nom de l'élément et décrit son contenu.
- Le nom d'un élément utilisé dans le document XML doit être identique à celui déclaré dans la DTD.
- Un élément peut:
  - contenir du texte,
  - contenir d'autres éléments,
  - contenir un mélange de texte et d'éléments (contenu mixte)
  - être vide.

# DTD, Déclaration des éléments

Chaque type d'élément doit être déclaré, cette déclaration respecte un des formats suivants:

**<!ELEMENT NOM (CONTENU)>**

**<!ELEMENT NOM (CONTENU\_MIXTE)\*>**

**<!ELEMENT NOM ANY>** *n'importe quelles données*

**<!ELEMENT NOM EMPTY >** *élément vide (<NOM/>)*

## Exemples:

**<!ELEMENT Nom (#PCDATA)>** *PCDATA: chaîne de caractères.*

**<Nom>**Victor Hugo**</Nom>**

**<!ELEMENT DATE (#PCDATA)>**

**<DATE>**26 Octobre 2010**</DATE>**

**<!ELEMENT BIBLIOTHEQUE (LIVRE)\*>**

**<!ELEMENT LIVRE (PREFACE, CHAPITRE+, INDEX?)>**

# Exemple

- DTD

```
<!ELEMENT personne (nom, prenom+, tel?,email,adresse >
```

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

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

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

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

```
<!ELEMENT adresse (ANY) >
```

- Document XML associé

```
<personne>
```

```
  <nom>Bennani</nom>
```

```
  <prenom>Mohammed</prenom>
```

```
  <prenom>Ali</prenom>
```

```
  <tel>0683000000</tel>
```

```
  <email>bennani@fsr.ac.ma</email>
```

```
  <adresse><rue/><ville>Rabat</ville></adresse>
```

```
</personne>
```

# DTD, Spécification du contenu d'un élément

La spécification du contenu d'un élément précise:

- le genre d'informations que l'élément peut contenir (texte, sous éléments, mixte) ,
- les contraintes sur son contenu.

# DTD, Spécification du contenu d'un élément

Les mots clés de description du contenu sont:

- **(#PCDATA)**: *Parsed Character Data*, du contenu littéral .
- **(ELEMENT)**: le sous-élément **ELEMENT**.
- **(ELEMENT1,ELEMENT2,...)** : une liste d'éléments appelée **séquence**. L'ordre d'apparition des éléments doit être respecté dans le document XML.
- **(ELEMENT1|ELEMENT2|...)** choix d'un sous-élément.
- **ELEMENT?** : zéro ou une fois.
- **ELEMENT+** : une ou plusieurs fois.
- **ELEMENT\*** : zéro ou plusieurs fois.

# DTD, Spécification du contenu d'un élément

Par exemple si on dit:

- qu'une liste de films *lfilms* contient des films, au moins un
- qu'un *film* contient un *titre* et zéro ou plusieurs *acteurs* (dans cet ordre),
- qu'un titre et un acteur sont des *chaines de caractères PCDATA* (*Parsed Character Data*)

On écrira la DTD suivante :

```
<!ELEMENT lfilms (film+)>
```

```
<!ELEMENT film (titre, acteur*)>
```

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

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

# DTD, Spécification du contenu d'un élément

## D'autres exemples:

➤ (ELEMENT1|ELEMENT2|....)

exemple: <!ELEMENT PAIEMENT (CASH| CB | CHEQUE)>

remarque: <!ELEMENT PAIEMENT (CASH| CB | CHEQUE)\*>

pas de contrainte sur l'ordre d'apparition et le nombre des éléments.

➤ ELEMENT?

exemple: <!ELEMENT FILM(AFFICHE?)>

➤ ELEMENT+

exemple: <!ELEMENT LIVRE(CHAPITRE+)>

# DTD interne (exemple 1)

## parents.xml

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<!-- Comme on définit une DTD interne, le fichier est indépendant
      (standalone).-->
<!DOCTYPE parents [
  <!ELEMENT parents (fille,garcon)>
  <!ELEMENT fille(#PCDATA)>
  <!ELEMENT garcon (#PCDATA)>
]>
<parents>
  <fille>Jalila</fille>
  <garcon>Sami</garcon>
</parents>
```

# DTD interne (exemple 2)

## bibliotheque.xml

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

```
<!DOCTYPE BIBLIOTHEQUE [  
  <!ELEMENT BIBLIOTHEQUE (LIVRE)* >  
  <!ELEMENT LIVRE (AUTEUR, TITRE,  
    EDITEUR)>  
  <!ELEMENT AUTEUR (PRENOM, NOM) >  
  <!ELEMENT TITRE (#PCDATA) >  
  <!ELEMENT EDITEUR (NOM, ANNEE) >  
  <!ELEMENT PRENOM (#PCDATA) >  
  <!ELEMENT NOM (#PCDATA) >  
  <!ELEMENT ANNEE (#PCDATA) >  
>
```

```
<BIBLIOTHEQUE>  
  <LIVRE>  
    <AUTEUR>  
      <PRENOM>Rolf</PRENOM>  
      <NOM> MAURERS</NOM>  
    </AUTEUR>  
    <TITRE>JAVA</TITRE>  
    <EDITEUR>  
      <NOM>Micro Application</NOM>  
      <ANNEE> 1996</ANNEE>  
    </EDITEUR>  
  </LIVRE>  
  ...  
</BIBLIOTHEQUE>
```

# DTD, Spécification du contenu d'un élément

## Contenu mixte:

Un élément peut contenir du texte et d'autres éléments.

Pour un contenu mixte, ni le nombre, ni l'ordre des éléments ne peuvent être spécifiés explicitement et le mot clé *#PCDATA* doit être le premier.

## Exemple 1: Document DTD

```
<!ELEMENT citation (#PCDATA|auteur)*>
```

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

Document XML valide:

```
<citation>
```

```
  <auteur>William Shakespeare</auteur> la folie chez les grands  
  ne doit pas aller sans surveillance.
```

```
</citation>
```

# DTD, Spécification du contenu d'un élément

## Exemple 2:

```
<!ELEMENT DATE (#PCDATA|DATEFR)*>
```

```
<!ELEMENT DATEFR (JJ,MM,AA)>
```

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

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

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

Voici des exemples de dates valides:

1- `<DATE> 12 Novembre 1999 </DATE>`

2- `<DATE>`

12 Novembre 1999

```
<DATEFR> <JJ>12</JJ> <MM>11</MM><AA>1999</AA> </DATEFR>
```

```
</DATE>
```

# Exercice 1: énoncé

Ecrire une DTD **biblio.dtd** pour une bibliographie d'articles.

Les informations associées à 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 réserve aussi un champ optionnel pour un avis personnel.

Tester cette DTD avec un fichier XML.

# Exercice 1: correction

## biblio.dtd

```
<!ELEMENT article (titre, auteur+, journal, avis?)>
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT journal (nom_journal, page, num_journal, annee)>
    <!ELEMENT nom_journal (#PCDATA)>
    <!ELEMENT page (#PCDATA)>
    <!ELEMENT num_journal (#PCDATA)>
    <!ELEMENT annee (#PCDATA)>
  <!ELEMENT avis(#PCDATA)>
```

# Exercice 1: correction

## biblio.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE article SYSTEM "biblio.dtd">
<article>
  <titre> A data-oriented survey of context models</titre>
  <auteur> Bolchini, C.</auteur> <auteur> Tanca, L.</auteur>
  <journal>
    <nom_journal> ACM Sigmod Record</nom_journal>
    <page>pp. 19-26</page>
    <num_journal>36</num_journal>
    <annee>2007</annee>
  </journal>
  <avis>un bon article</avis>
</article>
```

# DTD, Déclaration des attributs

La description des attributs se fait par une déclaration d'une liste d'attributs (**ATTLIST**)

La syntaxe est la suivante:

**<!ATTLIST Elément Attribut Type Valeur-par-défaut>**

avec **<!ATTLIST Elément Attribut Type #FIXED Valeur>**

FIXED signifie que l'attribut a une valeur fixe

ou **<!ATTLIST Elément Attribut Type #REQUIRED>**

REQUIRED signifie que l'attribut est obligatoire et n'a pas de valeur par défaut

ou **<!ATTLIST Elément Attribut Type #IMPLIED>**

IMPLIED signifie que l'attribut n'est pas obligatoire et n'a pas de valeur par défaut

# DTD, description des attributs

Exemple 1:

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

```
<!ATTLIST MESSAGE LANGUE CDATA "Français">
```

L'élément **MESSAGE** contient des données textuelles et peut contenir un **attribut** nommé **LANGUE**, sa valeur par défaut est "**Français**".

Remarque:

- l'ordre des déclarations des attributs d'un élément n'a pas d'importance

# DTD, description des attributs

## Exemple 2:

On peut, dans une même déclaration **ATTLIST**, définir plusieurs attributs associés au même élément:

```
<!ATTLIST IMG WITH CDATA "100">
```

```
<!ATTLIST IMG HEIGHT CDATA "100">
```

peuvent se résumer en une seule déclaration:

```
<!ATTLIST IMG WIDTH CDATA "100"  
             HEIGHT CDATA "100">
```

## DTD, Attributs: type énuméré

Nous pouvons limiter la liste de valeurs possibles pour un attribut. On le définit comme un type énuméré.

Exemple de déclaration d'une liste de choix d'attributs:

```
<!ELEMENT img EMPTY>
```

```
<!ATTLIST img format (GIF|JPEG|PNG) "GIF">
```

Nous déclarons un attribut *format* d'un élément *img*.

Cet attribut peut prendre une valeur parmi *GIF*, *JPEG* et *PNG*.

La valeur par défaut est *GIF*.

**Remarque:** *ne pas mettre des guillemets dans la liste des valeurs possibles.*

# Exemples

<!ELEMENT personne (nom, prenom+, tel?, email, adresse) >

<!ELEMENT nom (#PCDATA) >

<!ELEMENT prenom (#PCDATA) >

<!ELEMENT tel (#PCDATA) >

<!ELEMENT email (#PCDATA) >

<!ELEMENT adresse ANY >

<! ATTLIST personne

    age CDATA #IMPLIED

    genre (Masculin | Feminin ) #REQUIRED >

.....

<!ELEMENT auteur (#PCDATA) >

<!ATTLIST auteur

    genre (Masculin | Feminin ) #REQUIRED

    ville CDATA #IMPLIED >

<!ELEMENT editeur (#PCDATA) >

<!ATTLIST editeur

    ville CDATA #FIXED "Rabat" >

## Exercice 2: énoncé

Modifier la DTD **biblio.dtd** précédente:

- en faisant de l'élément *nom\_journal* un attribut de l'élément *journal* et en lui donnant comme valeur par défaut *ACM* ;
- en faisant de l'élément *annee* un attribut de type énuméré, prenant comme valeurs possibles *2012*, *2013*, *2014*, "*avant\_2012*" et proposant comme valeur par défaut "*avant\_2012*".
- Utiliser cette DTD pour créer un fichier XML valide.

# Exercice 2: correction

## biblio2.dtd

```
<!ELEMENT article (titre, auteur+, journal, avis?)>
  <!ELEMENT titre (#PCDATA)>
  <!ELEMENT auteur (#PCDATA)>
  <!ELEMENT journal (page, num_journal)>
    <!ATTLIST journal
      nom_journal CDATA "ACM"
      annee (2012 | 2013 | 2014 | avant_2012) "avant_2012"
    >
  <!ELEMENT page (#PCDATA)>
  <!ELEMENT num_journal (#PCDATA)>
<!ELEMENT avis (#PCDATA)>
```

# Attributs ID et IDREF

Ce type sert à indiquer que l'attribut concerné peut servir d'identifiant dans un fichier XML.

```
<?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">Latifa</Personne>
  < Personne PNum = "P2">Rachid</Personne>
  < Personne PNum = "P3" Mere = "P1" Pere = "P2">Ali</Personne>
  < Personne PNum = "P4" Mere = "P1" Pere = "P2">Samia</Personne>
</Document >
```

# Exercice

Donner une DTD qui valide des documents sous forme de carnets d'adresses.

Un carnet d'adresse a la forme suivante:

- la personne possède un identifiant unique (obligatoire), un nom, un prénom
- on veut connaître le sexe de la personne (attribut optionnel)
- on veut connaître son email (optionnel)

# Solution exercice

- Un carnet d'adresses:
  - la personne possède un identifiant unique (obligatoire), un nom, un prénom
  - on veut connaître le sexe de la personne (attribut optionnel)
  - on veut connaître son email (optionnel)

```
<!ELEMENT carnetAdresses (personne)+>
```

```
<!ELEMENT personne (nom,email*)>
```

```
  <!ATTLIST personne id ID #REQUIRED>
```

```
  <!ATTLIST personne sexe (masculin|feminin) #IMPLIED>
```

```
<!ELEMENT nom (#PCDATA|famille|prenom)*>
```

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

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

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

# Les entités XML

- Un document XML est composé d'un prologue et d'un élément racine contenant tous les autres éléments.
- Les différents éléments contenus dans la racine peuvent provenir de différentes sources (fichier, SGBD,...)
- Les unités de stockage des composants d'un document sont appelées des **entités**.
- Les entités peuvent être de deux types:
  - **internes**: définies dans l'entité document elle-même,
  - **externes**: dépendent d'une source externe au document XML.

# Les entités XML

## Utilité des entités:

- la définition de caractères difficiles (nationaux, graphiques,..)
- la définition de notations raccourcies pour des suites de caractères saisies fréquemment . Ce mécanisme s'apparente aux macros.
- L'inclusion de fichiers externes

# Les entités XML

## Définition d'une entité interne:

- Une entité interne correspond à une version abrégée d'un texte long ou difficile à saisir.
- Une entité interne est définie à l'intérieur d'une DTD:

**<!ENTITY nom "texte associé">**

**nom** est le nom de l'entité,

## Exemples de définition d'une entité:

**<!DOCTYPE CATALOGUE [**

**<!ENTITY copyright "Copyright 2013, Eyrolles">**

**<!ENTITY email "info@fsr.ac.ma">**

**]>**

# Les entités XML

## Définition d'une entité interne:

- La référence à une entité se fait en préfixant son nom avec **&** et à la fin un **;**  
exemple:       **&copyright;**
- toute référence à l'entité **copyright** dans le document XML utilisant une DTD, sera remplacé par le texte correspondant.

### ➤ Exemple: entite1.xml

```
<!DOCTYPE exemple[  
    <!ELEMENT exemple (#PCDATA)>  
    <!ENTITY DTD "Définition d'un Type de Documents (DTD)">  
>  
<exemple> Une &DTD; est une grammaire pour valider un type de documents  
XML  
</exemple>
```

# Les entités XML

## Définition d'une entité interne:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE LIVRE [
    <!ENTITY copyright "Copyright 2013, Eyrolles">
    <!ENTITY email "info@fsr.ac.ma">
]>
<LIVRE>
    ....
    ....
    <DROITS>&copyright;</DROITS>
    <ADRESSE>&email;</ADRESSE>
</LIVRE>
```

# Les entités XML

## Définition d'entité externe:

- L'entité externe est définie à l'intérieur de la DTD:

```
<!ENTITY nom SYSTEM "URI">
```

Soit l'exemple du document XML (*livre.xml*):

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

```
<!DOCTYPE LIVRE [
```

```
    <!ENTITY legal SYSTEM "legal.xml">
```

```
]>
```

```
<LIVRE>
```

```
.....
```

```
.....
```

```
    &legal;
```

```
</LIVRE>
```

# Les entités XML

## Définition d'entité externe:

➤ Si le fichier *legal.xml* est le suivant:

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

```
<LEGAL>
```

```
  <AUTEUR>Olivier Daudel</AUTEUR>
```

```
  <DATE>1991</DATE>
```

```
  <DROITS>Copyright 1991</DROITS>
```

```
</LEGAL>
```

# Les entités XML

## Définition d'entité externe: à tester

- Tout se passe comme si le document XML *livre.xml* est:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE LIVRE [
    <!ENTITY legal SYSTEM "legal.xml">
]>
<LIVRE>
    <LEGAL>
        <AUTEUR>Olivier Daudel</AUTEUR>
        <DATE>1991</DATE>
        <DROITS>Copyright 1991</DROITS>
    </LEGAL>
</LIVRE>
```