

Chapitre 3:
DTD pour la validation des documents XML

- Introduction**
 - ✓ Langages de définition de schéma
 - ✓ DTD: Document Type Definition
 - ✓ Types des DTD
- Syntaxes de déclarations des DTD**
 - ✓ Définition d'un DTD
- Définition des éléments**
 - ✓ Indicateurs d'occurrences
- Définition des attributs**
- Définition des entités**
 - ✓ Entités générales prédéfinies
 - ✓ Entités générales non prédéfinies
 - ✓ Entités paramètres
- Conclusion**

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

INTRODUCTION

- ♦ La validation consiste à vérifier que la structure d'un document est conforme à une grammaire (modèle), on dit que le document est valide;
- ♦ La grammaire permet de définir le vocabulaire (l'ensemble des éléments et des attributs) et les règles à appliquer pour structurer les données dans les documents XML (l'ordre des éléments, cardinalités, contenu des éléments....).

2

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

LANGAGES DE DÉFINITION DE SCHÉMA

- ♦ Il existe plusieurs langages dites langages de définition de schéma, nous citons les langages suivants :
 - ✓ **DTD (Document Type Definition)** : est un langage dérivé du langage SGML ;
 - ✓ **Schema XML** : est un langage dérivé du langage XML (voir chapitre 4);
 - ✓ **RelaxNG (Regular Language for XML Next Generation)**: est un langage qui combine les avantages des DTD et des XML Schema;
 - ✓ **Schematron**: est une norme ISO ayant une syntaxe XML.

3

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

DTD: Document Type Definition

- ♦ Un DTD définit la grammaire des documents c'est-à-dire:
 - ✓ Le **vocabulaire** qui désigne l'ensemble des éléments et des attributs;
 - ✓ Les **règles** permettant d'imposer une structure arborescente et des contraintes sur les nœuds de cette structure;
- ♦ Un DTD sert comme un modèle de document, un document qui est valide pour un DTD est appelé « une **instance** » de ce DTD et un groupe de documents qui sont valides pour un DTD est appelé « une **classe** » de documents.

4

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

TYPES DES DTD

- Un DTD doit être déclaré au début du document après le prologue et avant l'élément racine;
- Il existe trois types de DTD qui sont :
 - ✓ **DTD interne**: le DTD est inclus dans le même document XML;
 - ✓ **DTD externe**: le DTD et le document XML sont définis dans des fichiers séparés;
 - ✓ **DTD mixte**: nous pouvons trouver les deux types de DTD interne et externe dans le même document XML.

5

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

SYNTAXES DE DÉCLARATIONS DES DTD

Syntaxes:

- `<!DOCTYPE racine [liste des déclarations] >` **DTD interne**
- `<!DOCTYPE racine SYSTEM "Chemin DTD" | PUBLIC "URI DTD">` **DTD externe**
- `<!DOCTYPE racine SYSTEM "Chemin DTD" | PUBLIC "URI DTD" [liste des déclarations] >` **DTD mixte**

- ✓ **DOCTYPE**: mot clé pour introduire le DTD;
- ✓ **racine**: le nom de l'élément racine du document;
- ✓ **liste de déclarations**: dans cette partie où on définit le DTD;
- ✓ **SYSTEM**: lorsqu'il s'agit de votre propre DTD, le chemin soit local ou un lien vers un serveur;
- ✓ **PUBLIC**: lorsqu'il s'agit d'un DTD défini par un propriétaire (généralement de normalisation).

6

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

DÉFINITION D'UN DTD

- La définition d'un DTD comprend généralement:
 - ✓ la définition des **éléments** ;
 - ✓ la définition des **attributs** ;
 - ✓ la définition des **entités** ;
 - ✓ la définition des **notations** ;
 - ✓ Des **commentaires** .

7

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

DÉFINITION DES ÉLÉMENTS

- Un élément est défini comme suit: `<!ELEMENT nomElement typeContenu>`
- ✓ **ELEMENT**: mot clé pour introduire l'élément;
- ✓ **nomElement**: le nom de l'élément;
- ✓ **typeContenu**: dans cette partie où on définit le type du contenu, un élément peut contenir:
 - ✓ **EMPTY**: lorsqu'il s'agit d'un élément vide (mais peut contenir des attributs);
 - ✓ **ANY**: lorsqu'il s'agit d'un élément qui contient un contenu quelconque (dans ce cas rien ne sera testé);
 - ✓ **(#PCDATA)**: lorsqu'il s'agit d'un élément qui contient un contenu textuel (signifie **Parsable Character DATA**);
 - ✓ **(el₁,el₂,...,el_n)**: lorsqu'il s'agit d'un élément qui contient une **séquence** d'élément en respectant l'ordre;
 - ✓ **(el₁ | el₂ | ... | el_n)**: lorsqu'il s'agit d'un élément qui contient un seul élément parmi l'**énumération** (liste de choix);

8

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 1

Exemple1.xml

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE Etudiant [
  <!ELEMENT Etudiant (CNE,Nom,Prenom)>
  <!ELEMENT CNE (#PCDATA)>
  <!ELEMENT Nom (#PCDATA)>
  <!ELEMENT Prenom (#PCDATA)>
]>
<Etudiant>
  <CNE>001</CNE>
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
</Etudiant>
```

Il s'agit d'un DTD interne;

L'élément racine **Etudiant** doit **obligatoirement** contenir les sous éléments: CNE, Nom et Prenom en respectant l'ordre.

CNE, Nom et Prenom sont des éléments qui doivent contenir des données textuelles.

Le document XML est **valide** pour le DTD.

9

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 2

Exemple2.xml

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE Etudiant [
  <!ELEMENT Etudiant (CNE,Nom,Prenom)>
  <!ELEMENT CNE (#PCDATA)>
  <!ELEMENT Nom (#PCDATA)>
  <!ELEMENT Prenom (#PCDATA)>
]>
<Etudiant>
  <CNE>001</CNE>
  <Nom>Marrakchi</Nom>
  <email>a.marrakchi@gmail.com</email>
</Etudiant>
```

Le document XML n'est pas valide pour le DTD: l'élément « email » n'est pas défini comme sous éléments de « Etudiant » dans le DTD.

10

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 3

Exemple3.dtd

```
<!ELEMENT Etudiant (Nom | Prenom)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
```

Il s'agit d'un DTD externe;

L'élément racine **Etudiant** doit contenir un seul élément parmi la liste de choix.

Exemple3.xml

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Etudiant SYSTEM "Exemple3.dtd">
<Etudiant>
  <Nom>Marrakchi</Nom>
</Etudiant>
```

Déclaration du DTD (le DTD et fichier XML sont dans le même dossier sinon il faut précisé le chemin complet).

Le document XML est valide pour le DTD.

11

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 4

Exemple4.dtd

```
<!ELEMENT Etudiant (Nom | Prenom)>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
```

Exemple4.xml

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Etudiant SYSTEM "Exemple4.dtd">
<Etudiant>
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
</Etudiant>
```

Le document XML n'est pas valide pour le DTD: l'élément « Etudiant » doit contenir un seul élément parmi la liste de choix.

12

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 5

Exemple5.dtd

```

<!ELEMENT Etudiant ANY>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
<!ELEMENT Adresse (#PCDATA)>
<!ELEMENT Photo EMPTY>

```

L'élément racine **Etudiant** peut contenir n'importe quel contenu et n'importe quel élément sans aucune contrainte sur l'ordre et sur les occurrences des éléments.

Exemple5.xml

```

<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Etudiant SYSTEM "Exemple5.dtd">
<Etudiant> 001
  <Adresse> Rue Hassan II N°12 Safi</Adresse>
  <Nom>Marrakchi</Nom>
  <Photo />
  <Prenom>Alae</Prenom>
</Etudiant>

```

L'élément **Photo** est vide.

Le document XML est valide pour le DTD.

13

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 6

Exemple6.xml

```

<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE Etudiant [
  <ELEMENT Etudiant (CNE,Nom,Prenom)>
  <ELEMENT CNE (#PCDATA)>
  <ELEMENT Nom (#PCDATA)>
  <ELEMENT Prenom (#PCDATA)>
]>
<Etudiant>
  <CNE>001</CNE>
  <Nom>Marrakchi</Nom>
</Etudiant>

```

Le document XML n'est pas valide pour le DTD: il manque l'élément « Prenom ».

14

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 7

Exemple6.xml

```

<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE Etudiant [
  <ELEMENT Etudiant (CNE,Nom,Prenom)>
  <ELEMENT CNE (#PCDATA)>
  <ELEMENT Nom (#PCDATA)>
  <ELEMENT Prenom (#PCDATA)>
]>
<Etudiant>
  <CNE>001</CNE>
  <Prenom>Alae</Prenom>
  <Nom>Marrakchi</Nom>
</Etudiant>

```

Le document XML n'est pas valide pour le DTD: l'ordre des éléments n'est pas respecté.

15

Introduction | Déclarations de DTD | **Définition des éléments** | Définition des attributs | Définition des entités | Conclusion

INDICATEURS D'OCCURRENCES

♦ Un élément est caractérisé par le fait de contenir plusieurs éléments y compris des éléments qui portent le même nom, il existe trois opérateurs d'occurrences (concept de cardinalité) permettant d'exprimer le nombre de répétition des sous éléments, le tableau ci-après résume ces opérateurs :

Opérateur	Signification
?	0 ou 1 occurrence (0,1)
*	0 ou plusieurs occurrences (0,n)
+	1 ou plusieurs occurrences (1,n)

16

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

EXEMPLES

```
<!ELEMENT Etudiant (CNE,Nom+,Prenom+,Tele*)>
```

- ✓ L'élément « Etudiant » doit avoir: 1 seul sous élément « CNE », au minimum 1 sous élément « Nom », au minimum 1 sous élément « Prenom » et 0 ou plusieurs sous éléments « Tele » dans cet ordre;

```
<!ELEMENT Etudiant (CNE,(Nom,Prenom)+,Tele*)>
```

- ✓ L'élément « Etudiant » doit avoir: 1 seul sous élément « CNE », 1 ou plusieurs sous éléments « Nom » et « Prenom » (les 2 à la fois) et 0 ou plusieurs sous éléments « Tele » dans cet ordre;

```
<!ELEMENT Etudiant (CNE,Nom,Prenom,Tele)*>
```

- ✓ L'élément « Etudiant » peut avoir 0 ou plusieurs fois la suite de tous les éléments entre parenthèses dans cet ordre;

17

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

EXEMPLES

```
<!ELEMENT Etudiant (#PCDATA|Nom)*>
```

- ✓ Il s'agit d'un exemple d'un élément ayant un contenu mixte (mélange des textes et des éléments).

```
<Etudiant>
  Mon nom est: <Nom>Alae Marrakchi</Nom>.
</Etudiant>
```

- ✓ **Remarques:**
 - ✓ Dans le cas d'un contenu mixte, nous ne pouvons pas exprimer l'ordre des éléments et leurs nombre de répétition;
 - ✓ Le mot clé #PCDATA doit être exprimé le 1^{er} dans la séquence.
- ✓ Nous pouvons proposer la syntaxe suivante pour un contenu mixte:

```
<!ELEMENT mixte(#PCDATA|elt1|elt2|...|eltn)*>
```

18

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

DÉFINITION DES ATTRIBUTS

► Un attribut est défini comme suit:

```
<!ATTLIST nomElement nomAttribut type obligation valeurParDefaut>
```

- ✓ **ATTLIST**: mot clé pour introduire un attribut;
- ✓ **nomElement**: le nom de l'élément auquel appartient l'attribut;
- ✓ **nomAttribut**: le nom de l'attribut;
- ✓ **obligation**: désigne la présence de l'attribut, elle prend les valeurs suivantes:
 - ✓ **#REQUIRED**: indique que l'attribut est obligatoire et n'a pas une valeur par défaut;
 - ✓ **#IMPLIED**: indique que l'attribut est optionnel et n'a pas une valeur par défaut;
 - ✓ **#FIXED**: indique que l'attribut a une valeur fixe non modifiable.

19

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

DÉFINITION DES ATTRIBUTS

- ✓ **type**: le type de l'attribut, les principaux types sont:
 - ✓ **CDATA**: une chaîne de caractères (Character Data) ;
 - ✓ **ID**: un identificateur unique qui permet de distinguer l'élément dans le document;
 - ✓ **IDREF**: une référence vers un ID, il a toujours l'obligation REQUIRED;
 - ✓ **IDREFS**: une liste de références (séparées par des espaces) vers des ID, il a toujours l'obligation REQUIRED;
 - ✓ **NMTOKEN**: un mot (mélange des : chiffres, lettres, '.', '_', '-' et ':' sans blanc);
 - ✓ **NMTOKENS**: une liste de mots (liste de **NMTOKEN** séparées par des blancs);
 - ✓ **(val₁ | val₂ | ... | val_n)**: énumération (liste de choix) de valeurs.
- valeurParDefaut**: une valeur qui sera prise en considération en cas d'absence d'une valeur de l'attribut.

20

Introduction | Déclarations de DTD | Définition des éléments | **Définition des attributs** | Définition des entités | Conclusion

EXERCICE D'APPLICATION

Exemple1.DTD

```
<!ATTLIST Photo Description CDATA #IMPLIED >
<!ATTLIST Photo H CDATA #FIXED "80">
<!ATTLIST Photo L CDATA #FIXED "60">
<!ATTLIST Etudiants Responsable IDREF #REQUIRED>
<!ELEMENT Etudiant (Nom,Prenom,Photo)>
<!ELEMENT Photo EMPTY>
<!ATTLIST Etudiant CNE ID #REQUIRED>
<!ATTLIST Etudiant Sexe (Masculin|Feminin) #REQUIRED>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
<!ATTLIST Photo Source CDATA #REQUIRED>
<!ELEMENT Etudiants (Etudiant+)>
<!ATTLIST Photo Type (JPG | PNG | GIF) "PNG">
```

Question:
Proposer un document XML valide pour ce DTD.

21

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

DÉFINITION DES ENTITÉS

- Une entité est un alias (raccourci) qui permet d'associer une valeur à un symbole, ce dernier sera utilisé dans le document XML comme un raccourci vers sa valeur (appelée souvent « **texte de remplacement** »).
- Dans le cadre de ce cours nous étudierons les entités suivantes :
 - ✓ Entités générales prédéfinies ;
 - ✓ Entités générales non prédéfinies ;
 - ✓ Entités paramètres.

22

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

ENTITÉS GÉNÉRALES PRÉDÉFINIES

- Une entité prédéfinie est une entité que nous pouvons utiliser sans l'avoir définie dans le DTD, dans la plupart des cas elle est utilisée pour inclure des caractères spéciaux, elle est introduite par le caractère "&" suivi de son nom et terminée par le caractère ";".
- Le tableau ci-contre illustre les principales entités prédéfinies:

Entité	Caractère
<	<
>	>
'	'
"	"
&	&

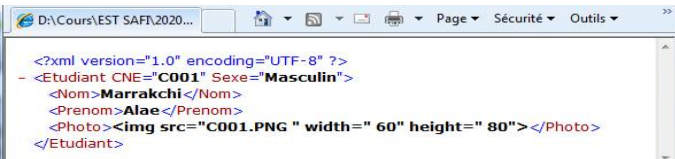
23

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

EXEMPLE

predefies.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Etudiant CNE="C001" Sexe="Masculin">
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
  <Photo> &lt;img src="C001.PNG" width="60" height="80" />
</Etudiant>
```



24

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

LA SECTION CDATA

➔ Dans le cas d'un élément ayant un contenu textuel (texte brut) qui ne doit pas être analysé par le parseur comme étant une syntaxe xml, nous pouvons utiliser une section appelée CDATA, le texte de cette dernière sera ignoré par le parseur et il sera considéré comme un texte dont les caractères seront affichés tel quels sont.

Syntaxe: `<![CDATA[texte qui sera ignoré par le parseur]]>`

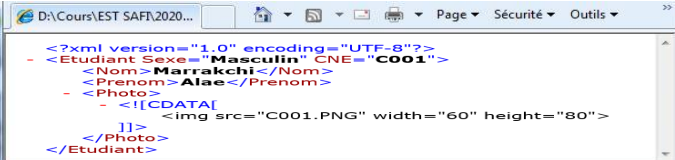
25

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

EXEMPLE

cdata.xml

```
<?xml version='1.0' encoding='UTF-8' ?>
<Etudiant CNE="C001" Sexe="Masculin">
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
  <Photo>
    <![CDATA[  ]]>
  </Photo>
</Etudiant>
```



26

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

ENTITÉS GÉNÉRALES NON PRÉDÉFINIES

➔ Il existe deux types des entités générales non prédéfinies :

- ✓ **Interne**: si la valeur correspondante au symbole est exprimée au sein du DTD;
- ✓ **Externe**: si la valeur correspondante au symbole est exprimée dans un fichier externe en dehors du DTD.

Syntaxes:

<code><!ENTITY Symbole "Texte de remplacement"></code>	Entité interne
<code><!ENTITY Symbole SYSTEM "Chemin Entité"></code>	Entité externe

✓ **ENTITY**: mot clé pour introduire l'entité; ✓ **Texte de remplacement**: la valeur du symbole;
 ✓ **Symbole**: le nom de l'entité; ✓ **Chemin Entité**: le chemin vers le fichier de l'entité.

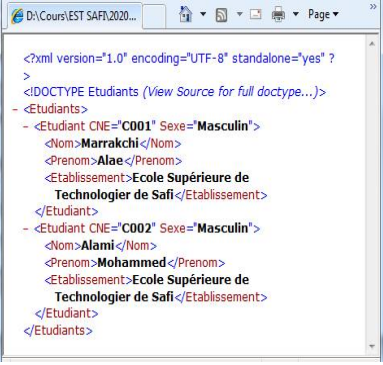
27

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | **Définition des entités** | Conclusion

EXEMPLE 1: ENTITÉ INTERNE

Exemple1.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE Etudiants [
  <ELEMENT Etudiants (Etudiant+)>
  <ELEMENT Etudiant (Nom,Prenom,Etablissement)>
  <ATTLIST Etudiant CNE ID #REQUIRED>
  <ATTLIST Etudiant Sexe (Masculin|Feminin) #REQUIRED>
  <ELEMENT Nom (#PCDATA)>
  <ELEMENT Prenom (#PCDATA)>
  <ELEMENT Etablissement (#PCDATA)>
  <ENTITY ESTS "Ecole Supérieure de Technologie de Safi">
]>
<Etudiants>
  <Etudiant CNE="C001" Sexe="Masculin">
    <Nom>Marrakchi</Nom>
    <Prenom>Alae</Prenom>
    <Etablissement>&ESTS;</Etablissement>
  </Etudiant>
  <Etudiant CNE="C002" Sexe="Masculin">
    <Nom>Alami</Nom>
    <Prenom>Mohammed</Prenom>
    <Etablissement>&ESTS;</Etablissement>
  </Etudiant>
</Etudiants>
```



28

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 2: ENTITÉ EXTERNE

Etudiant2.xml

```
<Etudiant CNE="C002" Sexe="Masculin">
  <Nom>Alami</Nom>
  <Prenom>Mohammed</Prenom>
  <Photo Source="/img/C002"/>
</Etudiant>
```

Etudiant3.xml

```
<Etudiant CNE="C003" Sexe="Feminin">
  <Nom>Nouri</Nom>
  <Prenom>Mariam</Prenom>
  <Photo Source="/img/C003" Type="JPG" Description="Photo
Nouri"/>
</Etudiant>
```

Externe.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE Etudiants SYSTEM "Externe.dtd">
<Etudiants Responsable="C001">
  &E1;
  &E12;
  &E13;
</Etudiants>
```

Externe.dtd

```
<!ELEMENT Etudiants (Etudiant+)>
<!ATTLIST Etudiants Responsable IDREF #REQUIRED>
<!ELEMENT Etudiant (Nom,Prenom,Photo)>
<!ATTLIST Etudiant CNE ID #REQUIRED>
<!ATTLIST Etudiant Sexe (Masculin|Feminin) #REQUIRED>
<!ELEMENT Nom (#PCDATA)>
<!ELEMENT Prenom (#PCDATA)>
<!ELEMENT Photo EMPTY>
<!ATTLIST Photo Source CDATA #REQUIRED>
<!ATTLIST Photo Type (JPG | PNG | GIF) "PNG">
<!ATTLIST Photo H CDATA #FIXED "80">
<!ATTLIST Photo L CDATA #FIXED "60">
<!ATTLIST Photo Description CDATA #IMPLIED>
<!ENTITY E1 SYSTEM "Etudiant1.xml">
<!ENTITY E12 SYSTEM "Etudiant2.xml">
<!ENTITY E13 SYSTEM "Etudiant3.xml">
```

Etudiant1.xml

```
<Etudiant CNE="C001" Sexe="Masculin">
  <Nom>Marrakchi</Nom>
  <Prenom>Alae</Prenom>
  <Photo Source="/img/C001"/>
</Etudiant>
```

29

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

ENTITÉS PARAMÈTRES

➔ Une entité paramètre est une entité utilisée uniquement au sein du DTD où elle a été définie, il existe deux types de ces entités: **interne** (si la valeur du symbole est exprimée au sein du DTD) et **externe** (si la valeur du symbole est exprimée dans un fichier externe au DTD).

Syntaxes:

- `<!ENTITY % Symbole "Texte de remplacement">` Entité paramètre interne
- `<!ENTITY % Symbole SYSTEM "Chemin Entité">` Entité paramètre externe

NB:

- ✓ Une entité paramètre doit être définie avant d'être utilisée;
- ✓ Une entité qui entre dans la définition d'un élément, d'un attribut ou d'une entité doit être défini **dans un DTD externe**.

30

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 1: DTD EXTERNE CONTENANT ENTITÉ INTERNE

parametre.xml

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE Staff SYSTEM "parametre.dtd">
<Staff>
  <Enseignants>
    <Enseignant Sexe="Feminin">
      <Nom>Nouri</Nom>
      <Prenom>Mariam</Prenom>
      <Departement>Informatique</Departement>
    </Enseignant>
  </Enseignants>
  <Administratifs>
    <Administratif Sexe="Masculin">
      <Nom>Marrakchi</Nom>
      <Prenom>Alae</Prenom>
      <Service>Ressources Humaines</Service>
    </Administratif>
  </Administratifs>
</Staff>
```

parametre.dtd

```
<!ENTITY % txt "#PCDATA">
<!ENTITY % infos "Nom,Prenom">
<!ENTITY % Genre "Sexe (Masculin|Feminin)">
<!ELEMENT Staff (Enseignants|Administratifs)+>
<!ELEMENT Enseignants (Enseignant+)>
<!ELEMENT Enseignant (%infos,Departement)>
<!ATTLIST Enseignant %Genre; #REQUIRED>
<!ELEMENT Administratifs (Administratif+)>
<!ELEMENT Administratif (%infos,Service)>
<!ATTLIST Administratif %Genre; #REQUIRED>
<!ELEMENT Nom (%txt:)>
<!ELEMENT Prenom (%txt:)>
<!ELEMENT Departement (%txt:)>
<!ELEMENT Service (%txt:)>
```

31

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

EXEMPLE 1: DTD INTERNE CONTENANT ENTITÉ EXTERNE

parametre.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE Staff [
  <!ELEMENT Staff (Enseignants|Administratifs)+>
  <!ENTITY % Sfr SYSTEM "parametre .dtd">
  %Sfr;
]>
<Staff>
  <Enseignants>
    <Enseignant Sexe="Feminin">
      <Nom>Nouri</Nom>
      <Prenom>Mariam</Prenom>
      <Departement>Informatique</Departement>
    </Enseignant>
  </Enseignants>
  <Administratifs>
    <Administratif Sexe="Masculin">
      <Nom>Marrakchi</Nom>
      <Prenom>Alae</Prenom>
      <Service>Ressources Humaines</Service>
    </Administratif>
  </Administratifs>
</Staff>
```

parametre.dtd

```
<!ENTITY % txt "#PCDATA">
<!ENTITY % infos "Nom,Prenom">
<!ENTITY % Genre "Sexe (Masculin|Feminin)">
<!ELEMENT Enseignants (Enseignant+)>
<!ELEMENT Enseignant (%infos,Departement)>
<!ATTLIST Enseignant %Genre; #REQUIRED>
<!ELEMENT Administratifs (Administratif+)>
<!ELEMENT Administratif (%infos,Service)>
<!ATTLIST Administratif %Genre; #REQUIRED>
<!ELEMENT Nom (%txt:)>
<!ELEMENT Prenom (%txt:)>
<!ELEMENT Departement (%txt:)>
<!ELEMENT Service (%txt:)>
```

32

Introduction | Déclarations de DTD | Définition des éléments | Définition des attributs | Définition des entités | Conclusion

CONCLUSION

- La validation consiste à vérifier que la structure d'un document est conforme à une grammaire (modèle);
- Un DTD est un modèle de documents;
- Un document valide pour un DTD est une instance du modèle défini par le DTD;
- Un groupe de documents valide pour un DTD est une classe de documents;
- Un document valide doit respecter les contraintes suivantes:
 - ✓ Être **bien formé** (respecte la syntaxe XML);
 - ✓ Respecte **la structure et les contraintes définies** dans le **modèle (DTD)**;
 - ✓ Toutes les **entités sont résolues**.

33

Merci pour votre attention !