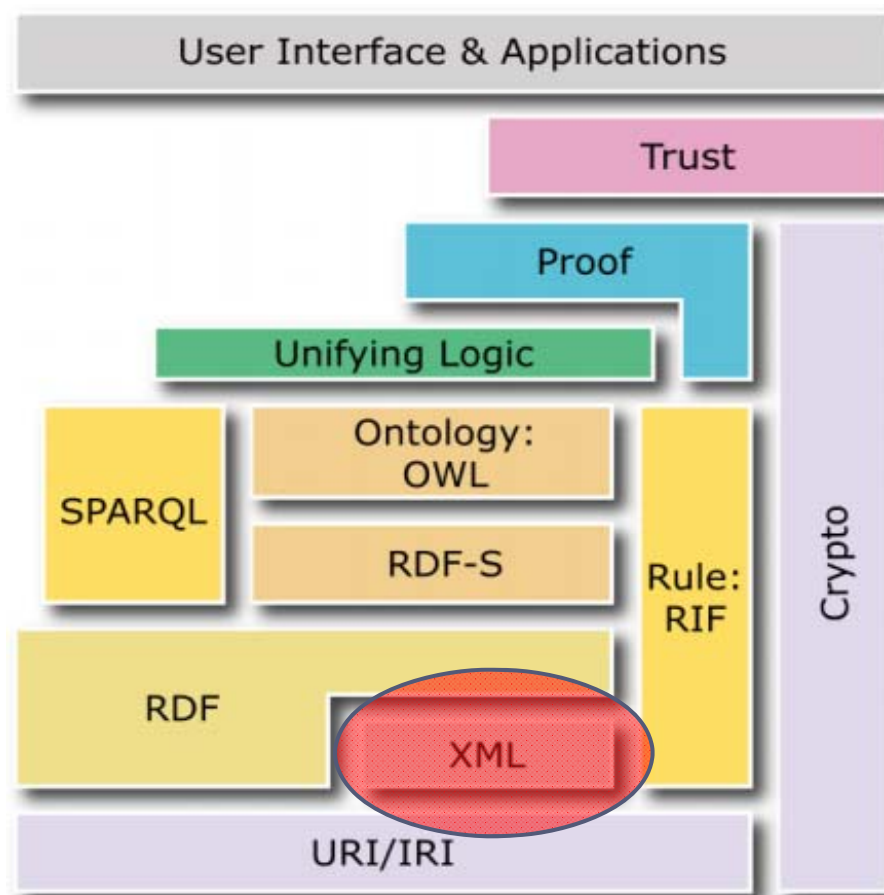

Web Sémantique & XML

Master IPS 2021/2022



Web sémantique

Approche en couches





Plan

Chapitre 2 **XML**

I – Introduction

II – Structure et Syntaxe

III – Espaces de nom

IV – Modèles de données : DTD

V – Modèles de données : Schéma XML



Document Type Definition (DTD)

DTD



IV – Document Type Definition (DTD)

Introduction

Document Type Definition (DTD)

- ▶ Une DTD permet de définir précisément la structure d'un document XML
- ▶ Il s'agit d'un certain nombre de contraintes que doit respecter ce document pour être **valide**
- ▶ Ces contraintes spécifient par exemple :
 - ▶ les éléments qui peuvent apparaître dans le contenu d'un autre élément,
 - ▶ l'ordre éventuel de ces éléments,
 - ▶ la présence de texte brut,
 - ▶ les attributs autorisés pour un élément,
 - ▶ les attributs obligatoires.



IV – Document Type Definition (DTD)

Déclaration de la DTD

- ▶ La déclaration de la DTD doit être placée dans le prologue:

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE root-element ... >
```

- ▶ *root-element* est le nom de l'élément racine du document auquel est associé la DTD
- ▶ La DTD peut être : **Interne**, **Externe** ou **Mixte**



IV – Document Type Definition (DTD)

Déclaration de la DTD

DTD Interne

- ▶ DTD est dite **interne** si elle est directement incluse dans le document.
- ▶ Son contenu est encadré par des caractères crochets '[' et ']' :

`<!DOCTYPE root-element [déclarations]>`

- ▶ **déclarations** : constituent la définition du type du document.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE simple [
3      <!ELEMENT simple (#PCDATA)>
4  ]>
5
6  <simple>
7      ...
8  </simple>
```



IV – Document Type Definition (DTD)

Déclaration de la DTD

DTD Externe

- ▶ la DTD est dite **externe** s'elle est contenue dans un autre fichier dont l'extension est généralement **.dtd**
- ▶ Le document XML se contente alors de donner l'adresse de sa DTD pour que les logiciels puissent y accéder.
- ▶ L'adresse de de la DTD peut être donnée explicitement par une **URL**
- ▶ La référence à cette URL est introduite par le mot clé **SYSTEM** suivi de l'URL délimitée par des apostrophes « ' » ou des guillemets « " ».

```
<!DOCTYPE root-element SYSTEM "url">
```




IV – Document Type Definition (DTD)

Déclaration de la DTD

DTD Externe

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

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE cours SYSTEM "http://www.omegane.org/bibliography.dtd">
3 <cours>
4     ...
5 </cours>
```

- ▶ soit simplement le nom d'un fichier local.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE cours SYSTEM "bibliography.dtd">
3 <cours>
4     ...
5 </cours>
```



IV – Document Type Definition (DTD)

Déclaration de la DTD – **DTD Mixte**

DTD Mixte

- ▶ Elle est constituée d'une partie **interne** et d'une partie **externe**.
- ▶ La DTD globale est alors constituée des déclarations **externes** suivies des déclarations **internes**.

```
<!DOCTYPE root-element SYSTEM "url" [déclarations]>
```

- ▶ Il **n'est pas possible** de déclarer plusieurs fois le **même élément** dans une DTD. Tout élément doit être déclaré en **interne** ou en **externe**
- ▶ Il est possible de déclarer plusieurs fois le **même attribut**.
 - ▶ La première déclaration est prioritaire.
 - ▶ Il est ainsi possible de donner une nouvelle déclaration à un attribut dans la partie Interne.



IV – Document Type Definition (DTD)

Contenu de la DTD - **Commentaire**

Commentaire

- ▶ Une DTD utilisent la même syntaxe que dans des commentaires XML
 - ▶ délimités par les chaînes de caractères « **<!--** » et « **-->** ».
- ▶ Ceux-ci sont placés au même niveau que les déclarations d'éléments, d'attributs et d'entités.
- ▶ **Ils ne peuvent pas** apparaître à l'intérieur d'une déclaration.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- DTD pour les bibliographies -->
3 <!ELEMENT bibliography (book)+>
4 <!-- Déclaration de l'élément book avec des enfants title, ..., isbn et url -->
5 <!ELEMENT book (title, author, year, publisher, isbn, url?)>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Entités

- ▶ Ce sont des symboles pouvant être définis dans une DTD et utilisés dans un document XML comme raccourcis d'écriture.
- ▶ Les déclarations d'entités permettent de disposer de l'équivalent de raccourcis clavier et de caractères non accessibles.
- ▶ Elles permettent également des inclusions de documents
- ▶ Les entités définies avec le caractère % ne sont utilisables que dans la DTD.
- ▶ En l'absence de ce caractère, les entités ne sont utilisables que dans le document XML.



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

- ▶ La déclaration d'une entité commence par le mot clé **ENTITY**
- ▶ Elle peut être **interne** et prend la forme suivante:

`<!ENTITY name "fragment">`

- ▶ Elle peut être **externe** et prend la formes suivantes:

`<!ENTITY name SYSTEM "url">`

- ▶ Une entité de nom *name* est utilisée (référéncée) par `&name;`



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

► Entités prédéfinies:

- Ils permettant d'inclure les caractères spéciaux '**<**', '**>**', '**&**', '**'**' et '**"**' dans les contenus d'éléments et dans les valeurs d'attributs.

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

- Les entités **<**, **>** et **&** doivent être utilisées aussi bien dans le contenu des éléments que dans les valeurs des attributs puisque les caractères '**<**', '**>**' et '**&**' ne peuvent pas y apparaître directement.
- Les entités **'** et **"** sont uniquement nécessaires pour inclure le caractère '**'**' ou le caractère '**"**' dans la valeur d'un attribut délimité par le même caractère.



IV – Document Type Definition (DTD)

Contenu de la DTD - **Entités**

Entités de caractères

- ▶ Elle servent à donner un nom facilement lisible à des caractères :
 - ▶ qui ne sont pas représentables dans l'alphabet utilisé
 - ▶ qui ne sont pas disponibles au clavier

```
<!ENTITY pound "&#163;" >
```

```
<!ENTITY eacute "&#233;" >
```

¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Les entités internes

- La **valeur d'une entité** interne est un fragment de document associé à celle-ci.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE eltRacine [
3      <!ENTITY ADN "Acide désoxyribonucléique">
4  ]>
5  <eltRacine>
6      L'&ADN; est une molécule complexe.
7  </eltRacine>
```




IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Les entités internes

- La **valeur** peut contenir des **caractères** ou des **éléments avec leurs balises**.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE price[
3     <!ENTITY euro "&#x20AC;">
4     <!ENTITY rceil '<phrase condition="html">&#x2309;</phrase>
5                     <phrase condition="fo" role="symbolfont">&#xF8F9;</phrase>'>
6 ]>
7
8 <price currency="Euro (&euro;)">30 &euro;</price>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Les entités internes

- ▶ Il est possible d'utiliser des entités dans la définition d'une autre entité pourvu que ces entités soient également définies.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE book [
3      <!-- Entités internes -->
4      <!ENTITY mci "Michel Colucci &aka; 'Coluche'">
5      <!ENTITY aka "also known as">
6  ]>
7
8  <book> &mci;</book>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Entités externes

- ▶ Elle peut désigner un fragment de document contenu dans un autre fichier.
- ▶ Ils peuvent être utilisés pour scinder un document en plusieurs fichiers.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE book [ <!-- Entités externes -->
3      <!ENTITY chapter1 SYSTEM "chapter1.xml">
4      <!ENTITY chapter2 SYSTEM "chapter2.xml">
5  ]>
6
7  <book>
8      <!-- Inclusion du fichier chapter1.xml -->
9      &chapter1;
10     <!-- Inclusion du fichier chapter2.xml -->
11     &chapter2;
12 </book>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Entités

Entités paramétriques

- ▶ Des entités **qui peuvent uniquement** être utilisées qu'à l'intérieur d'une DTD
- ▶ Elles sont utilisées pour apporter de la modularité aux DTD.
- ▶ La déclaration d'une entité paramètre prend une des formes suivantes:

```
<!ENTITY % name "fragment">
```

```
<!ENTITY % name SYSTEM "url">
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Déclaration de deux entités paramètres -->
3 <!ENTITY % idatt "id ID #REQUIRED">
4 <!ENTITY % langatt "xml:lang NMTOKEN 'fr'">
5
6 <!-- Utilisation des deux entités paramètres -->
7 <!ATTLIST chapter %idatt; %langatt;>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Déclaration d'élément

- ▶ Les déclarations d'éléments définissent la structure des documents valides.
- ▶ Elles spécifient quels **doivent être les enfants** de chaque élément et **l'ordre de ces enfants**.
- ▶ Une déclaration d'élément est de la forme :

<!ELEMENT *nom_element* *type_element* **>**

- ▶ *nom_element* : le nom de l'élément (un nom XML)
- ▶ *type_element* : le type auquel il est associé l'élément; il détermine son **contenu valide**.
- ▶ Le **contenu** d'un élément peut être :
 - ▶ **textuel**
 - ▶ **pur**
 - ▶ **mixtes**



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Elément à contenu textuel

- ▶ les **contenus textuels** sont constitués uniquement de texte formé de caractères, d'entités et de sections littérales.
- ▶ La déclaration d'un élément à contenu *textuel* est de la forme suivante :

`<!ELEMENT nom_element (#PCDATA) >`

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <!DOCTYPE texts [
4      <!ELEMENT texts (#PCDATA)>
5  ]>
6
7  <texts>
8      Du texte simple
9      Une <![CDATA[ Section CDATA avec < et > ]]>
10     Des entités &lt; et &gt;;
11 </texts>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Elément à contenu pur

- ▶ Le contenu d'un élément est ***pur*** lorsqu'il est uniquement constitué d'éléments qui sont ses enfants.
- ▶ Ces enfants peuvent avoir des contenus ***textuels***, ***purs*** ou ***mixtes***.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!DOCTYPE Personne [
4   <!ELEMENT Personne (Nom, Prénom)>
5   <!ELEMENT Prénom (#PCDATA)>
6   <!ELEMENT Nom (#PCDATA)>
7 ]>
8
9 <Personne>
10   <Nom>Brillant Alexandre</Nom>
11   <Prénom>Brillant Alexandre</Prénom>
12 </Personne>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Le **type** d'un élément peut être une **expression rationnelle** qui décrit les suites autorisées d'enfants :

```
<!ELEMENT nom_element regexp >
```

- ▶ **regexp** : est une expression rationnelle construite à partir des noms d'éléments en utilisant les **Indicateurs d'occurrence** :

' , ', ' | ', ' ? ', ' * ' et ' + ' ainsi que les parenthèses ' (' et ') '

Opérateur	Signification
,	Mise en séquence
	Choix
?	0 ou 1 occurrence
*	Répétition d'un nombre quelconque d'occurrences
+	Répétition d'un nombre non nul d'occurrences



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Exemples:

L'élément `elem` doit contenir un élément `elem1`, un élément `elem2` puis un élément `elem3` **dans cet ordre**

```
<!ELEMENT elem (elem1, elem2, elem3)>
```

Exemple d'utilisation valide :

```
<elem>  
  
</elem>
```

```
<!ELEMENT elem (elem1 | elem2 | elem3)*>
```

Exemples d'utilisations non valides :

```
<elem>  
  
    <elem1>( ... )</elem1>  
    <elem3>( ... )</elem3>  
    <elem2>( ... )</elem2>  
  
</elem>
```

car l'ordre des éléments n'est pas respecté

L'élément `elem` doit contenir un seul des éléments `elem1`, `elem2` ou `elem3`

```
<!ELEMENT elem (elem1 | elem2 | elem3)>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Exemples:

L'élément `elem` doit contenir un élément `elem1`, un élément `elem2` puis un élément `elem3` **dans cet ordre**

```
<!ELEMENT elem (elem1, elem2, elem3)>
```

Exemples d'utilisations valides :

```
<elem>
    <elem1>( ... )</elem1>
    <elem2>( ... )</elem2>
    <elem3>( ... )</elem3>
</elem>
```

Exemples d'utilisations **NON** valides :

```
<elem>
    <elem1>( ... )</elem1>
    <elem2>( ... )</elem2>
</elem>
```

Exemples d'utilisations non valides :

```
<elem>
    <elem1>( ... )</elem1>
    <elem3>( ... )</elem3>
    <elem2>( ... )</elem2>
</elem>
```

car l'ordre des éléments n'est pas respecté



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

L'élément `elem` doit contenir un élément `elem1`, un ou zéro élément `elem2` puis un élément `elem3` dans cet ordre

```
<!ELEMENT elem (elem1, elem2?, elem3)>
```

L'élément `elem` doit contenir un élément `elem1`, une suite éventuellement vide d'éléments `elem2` et un élément `elem3` dans cet ordre

```
<!ELEMENT elem (elem1, elem2*, elem3)>
```

L'élément `elem` doit contenir un élément `elem1`, un élément `elem2` ou un élément `elem4` puis un élément `elem3` dans cet ordre

```
<!ELEMENT elem (elem1, (elem2 | elem4), elem3)>
```

L'élément `elem` doit contenir une suite quelconque d'éléments `elem1`, `elem2` ou `elem3`

```
<!ELEMENT elem (elem1 | elem2 | elem3)*>
```

L'élément `elem` doit contenir une suite non vide d'éléments `elem1`, `elem2` ou `elem3`

```
<!ELEMENT elem (elem1 | elem2 | elem3)+>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - Déclaration d'élément

Élément à contenu mixte

- ▶ Ce type d'élément peut contenir du **texte** et des **éléments-enfants**.
- ▶ Il se présente comme une liste de choix avec des **indicateurs d'occurrence**

```
<!ELEMENT element (#PCDATA | element1 | ... | elementN)*>
```

- ▶ indique qu'un élément peut uniquement contenir du `texte` et les éléments `element1, ... , elementN`.
- ▶ Dans une telle déclaration:
 - le mot clé `#PCDATA` doit apparaître en premier.
 - Il n'y a aucun contrôle sur le nombre des éléments et sur leur ordre.
 - Le texte contenu peut se trouver à n'importe quel endroit



IV – Document Type Definition (DTD)

Contenu de la DTD - **Déclaration d'élément**

Élément à contenu vide

- ▶ C'est un élément dont le contenu est nécessairement vide

`<!ELEMENT nom_element EMPTY >`

- ▶ Cet élément peut posséder des attributs

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE texts [
3      <!ELEMENT texts (nom_element)>
4      <!ELEMENT nom_element EMPTY>
5  ]>
6
7  <texts>
8      <nom_element/>
9  </texts>
```



IV – Document Type Definition (DTD)

Contenu de la DTD - **Déclaration d'élément**

- ▶ **Élément à contenu quelconque**

- ▶ Il se déclare par

- ```
<!ELEMENT nom_element ANY>
```

- ▶ Cette déclaration n'impose aucune contrainte sur le contenu de l'élément
    - ▶ Mais le contenu doit être bien formé
    - ▶ Les éléments contenus doivent également être déclarés



## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

---

#### Déclaration d'attribut

- ▶ Pour qu'un document soit valide, tout attribut présent dans un élément doit être déclaré.
- ▶ La déclaration d'attribut est sous forme:

```
<!ATTLIST element attribut type default>
```

- ▶ `element` : nom de l'élément auquel il appartient l'attribut
  - ▶ **`attribut`** : nom de l'attribut (nom XML)
  - ▶ **`type`** : type de l'attribut
  - ▶ **`default`** : valeur par défaut de l'attribut
- ▶ Il est possible de déclarer plusieurs attributs pour un même élément.

```
<!ATTLIST element
 attribut1 type1 default1
 attribut2 type2 default2
 ...
 attributN typeN defaultN
>
```



## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

#### Types des attributs

##### ► Type chaîne de caractères

- C'est le du type d'attribut le plus courant dans les DTDs
- Une chaîne de caractères peut être composée de caractères ou/et des entités analysables.
- Le mot-clef **CDATA** (Character Data) est utilisé pour la déclaration de chaîne de caractère.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE texts [
3 <!-- Déclaration -->
4 <!ELEMENT texts (element)*>
5 <!ELEMENT element (#PCDATA)>
6 <!ATTLIST element attr CDATA "Valeur par défaut">
7 <!ENTITY car "caractères">
8]>
9
10 <texts>
11 <!-- Utilisation -->
12 <element attr="Chaîne de caractères"> (...) </element>
13 <element attr="Chaîne de &car;"> (...) </element>
14 </texts>
```





## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

---

#### Autres Type

- ▶ **(value-1 | value-2 | ... | value-N)**  
La valeur de l'attribut doit être un parmi : **value-1, value-2, ... value-N**.
- ▶ **NMTOKEN**  
La valeur de l'attribut est un **nom** (une suite quelconque de ces caractères sans espace).
- ▶ **NMTOKENS**  
La valeur de l'attribut est une liste de **nom** séparés par des espaces.
- ▶ **ID**  
La valeur de l'attribut est un identifiant unique (nom XML).
- ▶ **IDREF**  
La valeur de l'attribut est une référence vers un ID.
- ▶ **IDREFS**  
La valeur de l'attribut est une liste de références séparées par des espaces.



## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

---

- ▶ Chaque attribut peut être **obligatoire**, **optionnel** ou **fixe** et avoir une valeur **par défaut**.

- ▶ Déclaration d'un attribut avec une valeur par défaut :  
**"valeur" ou 'valeur'**

```
<!ELEMENT elt (...) >
 <!ATTLIST elt attr CDATA "ma_valeur" >
```

- ▶ L'attribut n'est pas obligatoire.
- ▶ S'il est omis dans le fichier XML lors de l'utilisation de l'élément **elt**, il est considéré comme valant **ma\_valeur**.

```
9 <element>(...)</element>
10 <!-- cela est équivalent à -->
11 <element attr= "ma_valeur" ">(...)</element>
```



## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

---

#### ► Déclaration d'un attribut obligatoire : **#REQUIRED**

```
<!ELEMENT elt (...) >
 <!ATTLIST elt attr CDATA #REQUIRED >
```

- L'attribut est **obligatoire**.
- Son absence déclenche une erreur du vérificateur syntaxique.

#### ► Déclaration d'un attribut optionnel : **#IMPLIED**

```
<!ELEMENT elt (...) >
 <!ATTLIST elt attr CDATA #IMPLIED >
```

- L'attribut est **optionnel** et il n'a pas de valeur par défaut.
- Si l'attribut est absent, il n'a pas de valeur.



## IV – Document Type Definition (DTD)

### Contenu de la DTD - Déclaration d'attribut

---

- ▶ **Déclaration d'un attribut avec une valeur fixe : `#FIXED "value"`**

```
<!ELEMENT elt (...)>
 <!ATTLIST elt attr CDATA #FIXED "value" >
```

- ▶ La valeur de l'attribut est **fixée** à la valeur *value* donnée. Si l'attribut est absent, sa valeur est implicitement *value*
- ▶ Si l'attribut est présent, sa valeur doit être *value* pour que le document soit valide.



# Exercices

---



# Schéma XLM