

COURS XML SCHEMA

1) Définition de XML schéma :

Ce langage fournit des nouveautés par rapport au DTD pour mieux contrôler la structure des documents XML.

2) Les apports des schémas :

- Les schémas XML sont décrits en XML.
- L'introduction du "typage des données"
- Le support des espaces de noms
- Un nombre minimum et maximum de présence d'un élément peut être renseigné.
- La notion d'héritage. Les éléments peuvent hériter du contenu et des attributs d'un autre élément.

3) Structure de XML schéma :

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- déclarations d'éléments, d'attributs et de types ici -->
</xsd:schema>!
```

- L'élément racine est xsd:schema.
- Tout élément du langage XML Schema que vous souhaitez utiliser doit commencer par xsd:

4) Liaison du XML schéma dans un document XML:

Pour valider votre document XML à l'aide d'un schéma XML, vous devez ajouter à votre XML ceci :

```
<collection xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance"
xsd:noNamespaceSchemaLocation="Livres.xsd">
...!
</collection>!
```

5) Déclaration d'éléments simples:

- Un **élément simple** est un élément qui ne contient que des chaînes de caractères.

(il ne peut pas contenir d'autres éléments, ni d'attributs)

- Dans un schéma XML, un élément simple se déclare avec la balise `xsd:element`.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">!
<xsd:element name="age" type="xsd:decimal" />!
</xsd:schema>!
```

6) Définir des restrictions :

Un élément simple peut contenir des **restrictions**.

Exemple :

```
<xs:element name="age">
<xs:simpleType>
<xs:restriction base="xsd:int">
<xs:minInclusive value="15"/>
<xs:maxInclusive value="35"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
```

7) Déclaration d'éléments complexes:

- ❑ Un **élément complexe** est un élément qui peut contenir d'autres éléments ou bien des attributs.
- ❑ Dans un schéma XML, un élément complexe se déclare en utilisant la balise `xsd:complexType`.

```
<xsd:element name="personne" />  
<xsd:complexType">  
  Déclaration des sous éléments ou bien attributs...!  
</xsd:complexType>
```

❖ Les éléments vides :

```
xsd:element name="personne" />  
<xsd:complexType>  
  <xsd:attribute name="nom" type="xsd:string" />  
  <xsd:attribute name="prenom" type="xsd:string" />  
</xsd:complexType>
```

- Pour préciser la valeur par défaut a un attribut on met l'attribut **Default**
- La déclaration des attributs doit être placés après la définition des types complexes
- Un attribut est optionnel par défaut.
- Pour que l'attribut soit obligatoire :

```
<xsd:element name="personne" />
<xsd:complexType>
<xsd:attribute name="nom" type="xsd:string" use="required" />
<xsd:attribute name="prenom" type="xsd:string" />
</xsd:complexType>
```

❖ Les éléments qui contiennent d'autres éléments:

- Une séquence d'éléments :

```
<xsd:element name="personne" />
<xsd:complexType>
<xsd:sequence>
<xsd:element name="nom" type="xsd:string" />
<xsd:element name="prenom" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
```

- Une alternative d'éléments :

À la place de la balise `<xsd:séquence>` `</xsd:sequence>` on met la balise `<xsd:choice>` `</xsd:choice>`

❖ Les éléments qui contiennent du texte et des attributs :

```
<xsd:element name="personne" />
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string" >
<xsd:attribute name="nationalité" type="xsd:string" />
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
```

Dans cet exemple, l'élément `personne` contient du texte et un attribut ayant pour nom `nationalité`.

❖ Les éléments qui contiennent du texte et d'autres éléments:

```
<xsd:element name="personne" />
<xsd:complexType mixed="true">
  <xsd:sequence>
    <xsd:element name="nom" type="xsd:string" />
    <xsd:element name="prenom" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

Résultat :

```
<personne>
Je suis <nom>Elouafi </nom> <prenom>Mouad</prenom>.
</personne>
```


8) Déclaration des indicateurs d'occurrences:

Il est possible de contrôler le nombre minimal et maximal d'occurrences d'un élément.

- Utilisation de **minOccurs** et **maxOccurs**.

```
<xsd:element name="personne" />
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="nom" type="xsd:string" />!
    <xsd:element name="prenom" type="xsd:string" minOccurs="1"</
maxOccurs="3">
  </xsd:sequence>
</xsd:complexType>
```

- Les attributs **minOccurs** et **maxOccurs** ont par défaut la valeur 1.
- La valeur **unbounded** peut être associée à maxOccurs. (infini)

**MERCI POUR VOTRE
ATTENTION**