

TD3: Schéma W3C

EX1: Déclarations simples

Objectif: Mise en jambe

Q : Qu'est ce?

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="root" type="xsd:integer"/>
</xsd:schema>
```

Q : Qu'est ce?

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="root">
    <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
        <xsd:maxExclusive value="25"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

Q : Qu'est ce?

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="root">
    <xsd:simpleType>
      <xsd:union>
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100"/>
          </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="300"/>
            <xsd:maxInclusive value="400"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:union>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

Q : Écrivez un XML Schéma qui donne un modèle de document XML ne contenant qu'un élément "AAA". Cet élément ne contient que du texte.

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA" type="xsd:string"/>
</xsd:schema>
```

Q : Nous voulons un élément racine "AAA" ne contenant qu'un élément "BBB" et qu'un élément

"CCC" (ces deux éléments sont de type string, par exemple). L'ordre d'apparition n'est pas important. On utilisera le pattern (patron?) "all" avec ses attributs "minOccurs" et "maxOccurs"

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
    <xsd:complexType mixed="false">
      <xsd:all minOccurs="1" maxOccurs="1">
        <xsd:element name="BBB" type="xsd:string"/>
        <xsd:element name="CCC" type="xsd:string"/>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Nous voulons un élément racine "AAA" ne contenant qu'un élément "BBB" suivis d'un élément "CCC"(ces deux éléments sont de type string, par exemple). L'ordre d'apparition est pas important. Utilisez le pattern (patron?) "sequence"; ses attributs ne sont pas nécessaires, leur valeurs par défaut étant 1.

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
    <xsd:complexType mixed="false">
      <xsd:sequence minOccurs="1" maxOccurs="1">
        <xsd:element name="BBB" type="xsd:string"/>
        <xsd:element name="CCC" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Ici, l'élément "AAA" contient un nombre quelconque d'éléments "BBB" et "CCC" (peut être zéro). On utilisera le pattern "sequence" avec les bonnes valeurs d'attributs. Le but ici est de voir la fonction des attributs "minOccurs" du pattern "sequence" et ceux des éléments fils (inclus dans la séquence). Le document suivant est valide:

```
<AAA xsi:noNamespaceSchemaLocation="correct_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <BBB>111</BBB>
  <CCC>YYY</CCC>
  <BBB>222</BBB>
  <BBB>333</BBB>
  <CCC>ZZZ</CCC>
</AAA>
```

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
    <xsd:complexType mixed="false">
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="BBB" type="xsd:string" minOccurs="0" />
        <xsd:element name="CCC" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Désormais, l'élément "AAA" contient soit "BBB", soit "CCC". On utilisera le pattern "choice".

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
    <xsd:complexType mixed="false">
      <xsd:choice minOccurs="1" maxOccurs="1">
        <xsd:element name="BBB" type="xsd:string"/>
        <xsd:element name="CCC" type="xsd:string"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Nous voulons un élément racine "root" qui ne contient que du texte et un seul attribut ("xx")

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="root">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute name="xx" type="xsd:string" use="required"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Écrivez un XML Schéma qui donne un modèle de document XML ne contenant qu'un élément "AAA". Cet élément racine ne contient que du texte...qui est l'élément par défaut...

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA" type="xsd:string"/>
</xsd:schema>
```

Q : Nous voulons maintenant que cet élément racine "AAA" puisse contenir à la fois du texte et un élément "BBB". Pour cela, il faut fixer l'attribut "mixed" de "complexType" à "vrai". Voyez vous une autre possibilité?

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
    <xsd:complexType mixed="true">
      <xsd:sequence minOccurs="1">
        <xsd:element name="BBB" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Q : Écrivez un XML Schéma qui donne un modèle de document XML ne contenant qu'un élément vide "AAA". Cet élément est défini comme un type complexe, de contenu complexe et est une restriction du type de base "anytype". Voici un document valide pour ce Schéma:

```
<AAA xsi:noNamespaceSchemaLocation="correct_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
```

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:element name="AAA">
```

```
<xsd:complexType>
  <xsd:complexContent>
    <xsd:restriction base="xsd:anyType"/>
  </xsd:complexContent>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

EX2: Déclarations de types

Objectif: Le but est d'écrire un schéma W3C pour les documents recettes de cuisine, comme par exemple le fichier egg.xml

Q : Voici le document egg.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<recette
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="cook.xsd">
  <nom>Oeufs à la coque</nom>
  <portion>4</portion>
  <preparation value="1" unit="min"/>
  <cuisson value="3" unit="min"/>
  <ingredients>
    <ing><nom id="ing1">oeufs</nom><nbre>4</nbre></ing>
  </ingredients>
  <guide>
    <block>Plongez délicatement les <renvoi ref="ing1">oeufs</renvoi>
    dans de l'eau portée à ébullition. </block><block>Faîtes reprendre l'ébullition,
    puis baissez le feu et laissez cuire à petits bouillons pendant
    <cuisson value="3" unit="min"/> minutes. </block><block>Égouttez les
    <renvoi ref="ing1">oeufs</renvoi>, posez-les dans des coquetiers et coupez
    horizontalement la partie supérieure pour les déguster.</block>
  </guide>
</recette>
```

- Écrivez le modèle de l'élément "recette". Il s'agit d'un élément de type complexe constitué d'une séquence d'éléments "nom", "portion", "préparation", "cuisson", "repos", "ingrédients" et "guide". Remarque: il peut ne pas y avoir de temps de repos ou de temps de cuisson.
- Écrivez les modèles des éléments "nom" et "portion". Ensuite, écrivez les modèles des éléments "préparation", "cuisson" et "repos" en utilisant le type "uniteTempsType" (Déclaré plus loin).
- Écrivez le modèle de l'élément "ingrédients": il s'agit d'une séquence non limitée d'éléments "ing" (pour ingrédient). Les éléments "ing" sont de type "iType". De même, on demande d'écrire le modèle de l'élément "guide".
- Déclarez le type "blockType". Cet élément complexe peut contenir une série non ordonnée d'éléments "renvoi" (de type "renvoiType"), "repos" (de type "uniteTempsType"), "strong" (de type "string") et "cuisson" (de type "uniteTempsType").
- Déclarez le type "renvoiType". Il s'agit d'un type dérivé du type de base "string" auquel on associe un attribut "ref", requis, de type "string". Remarquez que "renvoiType" est un type complexe mais qu'il ne contient pas d'éléments fils. Par conséquent, son contenu doit être inscrit dans un élément "xsd:simpleContent".
- Déclarez le type "uniteTempsType". On lui associe deux attributs ("value" et "unit"). Le type "uniteTempsType" est-il simple ou complexe? L'attribut "value" est de type "int" (type de base) mais le type de l'attribut "unit" est une restriction du type de base de "string" (les valeurs de cet attribut sont limitées à "minute" et "heure").
- Déclarez le type "iType". Il s'agit d'un type complexe puisqu'un élément de ce type contient

des éléments ("nom" et, au choix "nbre" et "poids"). L'élément "nom", fils d'un élément de type "iType", ne doit pas être confondu avec l'élément "nom" de "recette". C'est pourquoi on décide ici de préciser son type à l'intérieur même de la déclaration de type de "iType". Remarque: "nom" est de type complexe car on lui associe un attribut. Cependant, il ne contient pas d'éléments fils... on se rapportera à la déclaration de "blockType" pour gérer ce phénomène.

- Enfin, déclarez le type "poidsType". Il s'agit d'un type complexe dérivant du type de base simple "string" (Il est donc simpleContent). On lui associe un attribut "unit" de type (simple, bien évidemment) dérivant de "string" et ne pouvant prendre que les valeurs "gramme" ou "kilogramme".

Correction:

Q : Quelle est la forme du fichier Schéma W3C final?

Correction: `<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">`

```

<xsd:element name="recette">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="nom"/>
      <xsd:element ref="portion"/>
      <xsd:element ref="preparation"/>
      <xsd:element ref="cuisson" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="repos" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="ingredients"/>
      <xsd:element ref="guide"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="nom" type="xsd:string"/>
<xsd:element name="portion" type="xsd:int"/>
<xsd:element name="preparation" type="uniteTempsType"/>
<xsd:element name="cuisson" type="uniteTempsType"/>
<xsd:element name="repos" type="uniteTempsType"/>
<xsd:element name="ingredients">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ing" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ing" type="iType"/>
<xsd:element name="nbre" type="xsd:int"/>
<xsd:element name="poids" type="poidsType"/>
<xsd:element name="guide">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element ref="block" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="block" type="blockType"/>
<xsd:element name="renvoi" type="renvoiType"/>
<xsd:element name="strong" type="xsd:string"/>
<!-- definition des types -->
<xsd:complexType name="uniteTempsType">

```

```

<xsd:attribute name="value" type="xsd:int" use="required"/>
<xsd:attribute name="unit" use="required">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="min|h"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
<xsd:complexType name="iType">
  <xsd:sequence>
    <xsd:element name="nom">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="id" type="xsd:string" use="required"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:choice>
      <xsd:element ref="nbre"/>
      <xsd:element ref="poids"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="poidsType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="unit" use="required">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="gramme|kilogramme"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="blockType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element ref="renvoi" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="repos" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="strong" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="cuisson" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="renvoiType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="ref" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>

```

```
</xsd:complexType>
</xsd:schema>
```

EX3: Des DTD vers les schémas

Objectif: Les schémas permettent de décrire les modèles de données de façon plus précise.

Q : Transposez la DTD livres.dtd en un schéma XSD:

livres.dtd:

```
<!ELEMENT livres (livre*)>
<!ELEMENT livre (titre, auteur+, année, prix)>
<!-- ATTLIST livre edition CDATA #REQUIRED -->
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (prenom, nom, laboratoire?, pays)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT année (#PCDATA)>
<!ELEMENT prix EMPTY>
<!-- ATTLIST prix
monnaie CDATA #REQUIRED
valeur CDATA #REQUIRED -->
<!ELEMENT pays (#PCDATA)>
```

On propose un fichier XML associé:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE livres SYSTEM "livres.dtd">
<livres>
  <livre edition="paperback">
    <titre>XML Weekend Crash Course (with CD-ROM)</titre>
    <auteur>
      <prenom>Kay</prenom>
      <nom>Ethier</nom>
      <pays>USA</pays>
    </auteur>
    <auteur>
      <prenom>Alan</prenom>
      <nom>Houser</nom>
      <pays>USA</pays>
    </auteur>
    <annee>2004</annee>
    <prix monnaie="USD" valeur="17.49"/>
  </livre>
  <livre edition="relie">
    <titre>Data on the Web: From Relations to Semistructured Data and XML</titre>
    <auteur>
      <prenom>Serge</prenom>
      <nom>Abiteboul</nom>
      <laboratoire>INRIA</laboratoire>
      <pays>F</pays>
    </auteur>
    <auteur>
      <prenom>Peter</prenom>
      <nom>Buneman</nom>
      <laboratoire>Upenn</laboratoire>
      <pays>USA</pays>
    </auteur>
```

```
<annee>1999</annee>
<prix monnaie="EUR" valeur="38.43"/>
</livre>
</livres>
```

Q : Que devient le document XML associé?

Correction:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="livres">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="livre" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="livre" type="livreType"/>
  <xsd:element name="titre" type="xsd:string"/>
  <xsd:element name="auteur" type="auteurType"/>
  <xsd:element name="annee">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9]{4}"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="prix" type="prixType"/>
  <xsd:element name="prenom" type="xsd:string"/>
  <xsd:element name="nom" type="xsd:string"/>
  <xsd:element name="laboratoire" type="xsd:string"/>
  <xsd:element name="pays" type="xsd:string"/>
  <xsd:complexType name="livreType">
    <xsd:sequence>
      <xsd:element ref="titre"/>
      <xsd:element ref="auteur" minOccurs="1" maxOccurs="unbounded"/>
      <xsd:element ref="annee"/>
      <xsd:element ref="prix"/>
    </xsd:sequence>
    <xsd:attribute name="edition" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="paperback"/>
          <xsd:enumeration value="relie"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="auteurType">
    <xsd:sequence>
      <xsd:element ref="prenom"/>
      <xsd:element ref="nom"/>
      <xsd:element ref="laboratoire" minOccurs="0"/>
      <xsd:element ref="pays"/>
    </xsd:sequence>
  </xsd:complexType>
```



```
<xsd:complexType name="prixType">
  <xsd:attribute name="valeur" type="xsd:decimal" use="required"/>
  <xsd:attribute name="monnaie" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="USD"/>
        <xsd:enumeration value="EUR"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
</xsd:schema>
```

Que devient le fichier xml?:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<livres
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="book.xsd">
  <livre edition="paperback">
    <titre>XML Weekend Crash Course (with CD-ROM)</titre> ...
```

Q : Modifiez votre schéma XML pour préciser les types utilisés selon les critères suivants:

- Une année est une chaîne de 4 caractères compris dans l'espace 0...9
- Un livre contient exactement un titre, un prix, une année et au moins un auteur. De plus, on associe à des éléments de ce type l'attribut "edition" qui précise si il s'agit d'un paperback ou d'une édition reliée.
- Un prix est un type complexe qui ne dérive pas d'un type simple et ne contient pas d'élément fils. Cependant, on lui associe deux attributs: "valeur" et "monnaie" ("monnaie" a un type qui dérive de "string", son champs de valeur se limite aux chaînes "USD" et "EUR")