

# XML

## TP – 01 (3/3)

### Validation par un Schéma XML

#### Question 1

1. Ecrire un document XML `SCH_Ques-1.xml` valide pour le schéma suivant:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="a" type="xsd:integer"/>
        <xsd:element name="b" type="xsd:integer" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

2. Ecrire un document XML `SCH_Ques-2.xml` valide pour le schéma suivant, mais pas pour le précédent:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="list">
    <xsd:complexType>
      <xsd:sequence maxOccurs="unbounded">
        <xsd:element name="a" type="xsd:integer"/>
        <xsd:element name="b" type="xsd:integer"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

3. Ecrire un document XML `SCH_Ques-3.xml` correspondant au schéma suivant:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="Color">
    <xsd:restriction base="xsd:token">
      <!-- Couleur, en hexadecimal RGB -->
      <xsd:pattern value="[0-9A-Fa-f]{6}"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="Coordinate">
    <xsd:sequence>
      <xsd:element name="x" type="xsd:double"/>
      <xsd:element name="y" type="xsd:double"/>
    </xsd:sequence>
  </xsd:complexType>
```

```
<xsd:element name="polygon">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="color" minOccurs="0" maxOccurs="1" type="Color"/>
      <xsd:element name="point" type="Coordinate" minOccurs="3" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

## Question 2

### Réalisation d'un schéma

Soit un document XML contenant un nombre indéterminé d'éléments sous la forme :

```
<contact titre="..." techno="...">
  <nom>...</nom>
  <prenom>...</prenom>
  <telephone> ...</telephone>
  <email>...</email>
  <email>...</email>
  <!-- ... -->
</contact>
```

1. Construire un schéma nommé **annuaire.xsd** dans lequel les textes seront des chaînes simples **xs:string** en utilisant un **type local anonyme** pour l'élément **contact**.
2. Changer votre Schéma en un autre schéma **annuaire1.xsd** de telle que :
  - a. utiliserez un type complexe et **ContactType** pour définir le type de l'élément **contact**
  - b. L'élément **telephone** et l'attribut **techno** sont en **option**.
  - c. utiliserez un type simple **NumeroType** pour préciser les valeurs acceptables pour le numéro de téléphone : **pattern 0[0-9]{9}**
  - d. utiliserez un type simple **EmailType** pour préciser les valeurs acceptables pour l'email : **pattern [a-z]+@[a-z]+\.[a-z]{2,3}**

## Question 3

1. Écrivez un exemple Qest3.xml validé par le schéma suivant :

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://ma.namespace/perso"
  elementFormDefault="qualified">
  <xsd:element name="doc">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="hello">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="world"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

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

### Question 3 (à rendre dans un délai d'une semaine par mail)

#### Construction de types simples et énumérations

Les **énumérations** (mot-clé `enumeration`) sont utilisées lorsqu'on souhaite reconnaître une valeur dans une liste de valeurs possibles.

Exemple :

```
<xs:simpleType name="couleurType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="bleu"/>
    <xs:enumeration value="blanc"/>
    <xs:enumeration value="rouge"/>
  </xs:restriction>
</xs:simpleType>
```

Dans cet exemple, nous autorisons pour le type `couleurType` les valeurs `bleu`, `blanc`, `rouge`.

1. Créez un schéma `annuaire2.xsd` à partir du schéma `annuaire1.xsd`. Définissez et utilisez les types simples suivants :
  - \* `technoType` : énumération dont les valeurs possibles sont **XML**, **Java**, **Autre**.
  - \* `telType` : liste de **5 entiers** (attention : créez d'abord un type pour la liste d'entiers).
  - \* `emailType` : pattern `[a-z]+@[a-z]+\.[a-z]{2,3}`
2. Validez ce nouveau schéma sur un document de votre conception.

### Question 4 (à rendre dans un délai d'une semaine par mail)

#### Types complexes et groupes

Les groupes sont des agglomérats d'éléments. Ils servent avant tout à stocker des relations entre éléments qui peuvent être présentes dans plusieurs types complexes. On peut, par exemple, imaginer que les éléments auteur et version pourraient se retrouver dans différentes parties d'un document (livre, sections, chapitres...).

Exemple :

```
<xsd:group name="monGroupe">
  <xsd:sequence>
    <xsd:element name="contact" type="xsd:string"/>
    <xsd:element name="note" type="xsd:string"/>
  </xsd:sequence>
```

```
</xsd:group>

<xsd:element name="listel">
  <xsd:complexType maxOccurs="unbounded">
    <xsd:group ref="monGroupe"/>
  </xsd:complexType>
</xsd:element>
```

Nous avons globalement défini le groupe **monGroupe** qui a été intégré à la définition complexe de l'élément **listel**.

1. Réalisez le schéma livre.xsd pour le document livre2.xml.
  - ✗ N'utilisez pas de type complexe anonyme.
  - ✗ Créez et utilisez un groupe représentant une liste d'auteurs (auteursGrp).
  - ✗ Créez et utilisez un groupe d'attributs (avecTitre) représentant un titre.
  - ✗ Faites en sorte que chaque section puisse également contenir une liste d'auteurs (donc en utilisant le groupe).