

XML – séance 3

1. Comptes bancaires

Examinez les fichiers Comptes1.xsd et Comptes2.xsd.

Comparez ces deux fichiers attentivement. Quelles différences constatez-vous ?

Ecrivez des documents XML valide selon l'un puis l'autre.

```
<?xml version = "1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Nom" type="xs:string" />
  <xs:element name="Prenom" type="xs:string" />
  <xs:element name="Visa" type="xs:positiveInteger" />
  <xs:element name="Bancontact" type="xs:positiveInteger" />
  <xs:element name="Proton"><xs:complexType/></xs:element>
  <xs:element name="CptVue">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Compte" />
        <xs:element ref="Visa" minOccurs="0" />
        <xs:element ref="Bancontact"
          minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="Proton" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CptEpargne">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Compte" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="Comptes">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="CptEpargne" />
      <xs:element ref="CptVue" />
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="Client">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Nom" />
      <xs:element ref="Prenom" />
      <xs:element ref="Comptes" />
    </xs:sequence>
    <xs:attribute name="Numero" use="required" type="xs:ID" />
  </xs:complexType>
</xs:element>
<xs:element name="Clients">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Client" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Compte">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="numeroSEPAType">
        <xs:attribute name="Monnaie" default="Eur" type="monnaieType" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="monnaieType">

```

```

    <xs:restriction base="xs:string">
        <xs:enumeration value="Usd" />
        <xs:enumeration value="Eur" />
        <xs:enumeration value="Cad" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="numeroSEPAType">
    <xs:restriction base="xs:string">
        <xs:pattern value="BE\d\d \d{4} \d{4} \d{4}" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Compte2.xsd

```

<?xml version = "1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Clients">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Client" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Nom" type="xs:string" />
                            <xs:element name="Prenom" type="xs:string" />
                            <xs:element name="Comptes">
                                <xs:complexType>
                                    <xs:choice maxOccurs="unbounded">
                                        <xs:element name="CptVue">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element ref="Compte" />
                                                    <xs:element name="Visa" minOccurs="0"
type="xs:positiveInteger"/>
                                                    <xs:element name="Bancontact"

```

```

type="xs:positiveInteger"
minOccurs="0"><xs:complexType/></xs:element>
minOccurs="0" maxOccurs="unbounded" />
<xs:element name="Proton"
  </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CptEpargne">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Compte" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Numero" use="required" type="xs:ID" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Compte">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="numeroSEPAType">
        <xs:attribute name="Monnaie" default="Eur" type="monnaieType" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="monnaieType">
  <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="Usd" />
        <xs:enumeration value="Eur" />
        <xs:enumeration value="Cad" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="numeroSEPAType">
    <xs:restriction base="xs:string">
        <xs:pattern value="BE\d\d \d{4} \d{4} \d{4}" />
    </xs:restriction>
</xs:simpleType></xs:schema>

```

2. Playlists

Regardez le fichier library.xsd :

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="library">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="songlist">
                    <xs:complexType>
                        <xs:sequence>
<xs:element name="song" type="songType" maxOccurs="unbounded" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="playlists" minOccurs="0" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="playlists">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="list" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
<xs:element name="track_ID" maxOccurs="unbounded" type="xs:positiveInteger"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

```

```

<xs:attribute name="name" type="xs:ID" use = "required"/>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="songType">
    <xs:sequence>
        <xs:element name="track_ID" type="xs:integer" />
        <xs:element name="name" type="xs:string" />
        <xs:element name="artist" minOccurs="0" type="xs:string" />
        <xs:element name="album" minOccurs="0" type="xs:string" />
        <xs:element name="genre" minOccurs="0" type="genreType" />
        <xs:element name="comments" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string" >
                        <xs:attribute name="author" type="xs:string"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="genreType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Hip hop"/><xs:enumeration value="Trip hop"/>
        <xs:enumeration value="Rock"/><xs:enumeration value="House" />
        <xs:enumeration value="Metal"/><xs:enumeration value="Disco"/>
        <xs:enumeration value="Pop"/><xs:enumeration value="Electro"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

- a) Quel est l'élément racine de ce fichier library.xsd ?

- b) Ecrivez un document xml valide selon ce schema.

- c) Reprenons library.xsd. Ecrivez un fichier library.dtd qui permettrait de valider les mêmes fichiers xml (càd qui correspond le plus possible aux contraintes du fichier library.xsd).

3. Brocante

Voici un fichier intitulé **brocante.dtd** :

```
<!ELEMENT brocante (habit)*>
<!ELEMENT habit (marque?,etat,prix)>
<!ATTLIST habit age CDATA #REQUIRED>
<!ATTLIST habit type (body|pyjama|pantalon|teeshirt|pull) #REQUIRED>
<!ATTLIST habit unite (mois|ans) "mois">
<!ELEMENT prix (#PCDATA)>
<!ELEMENT marque (#PCDATA)>
<!ELEMENT etat EMPTY>
<!ATTLIST etat est (use|bon|neuf) #REQUIRED>
```

a) Comparaison XSD-DTD

Examinez le fichier **brocante.xsd** présenté ci-dessous :

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema version="1.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="brocante" type="brocanteType" />

  <xsd:complexType name="brocanteType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="habit"
type="habitType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="habitType">
    <xsd:sequence>
      <xsd:element name="marque" type="xsd:string" />
      <xsd:element name="etat" type="etatType" />
      <xsd:element name="prix" type="xsd:decimal" />
    </xsd:sequence>
    <xsd:attribute name="type" type="xsd:string" />
    <xsd:attribute name="age" type="xsd:int" />
    <xsd:attribute name="unite" type="xsd:string" />
  </xsd:complexType>

  <xsd:complexType name="etatType">
    <xsd:attribute name="est" type="xsd:string" />
  </xsd:complexType>

</xsd:schema>
```


Comparez ces fichiers DTD et XSD. On vous demande d'expliquer 3 différences entre les validations que permettront ces fichiers. Argumentez.

Attention :

- Vous ne pouvez pas mettre que la DTD n'est pas du XML et le XSD si.
- Les différences que vous présentez ne peuvent être similaires. Par exemple (fictif !), si la première différence est que le nom doit être de longueur 24 dans l'un et de longueur 33 dans l'autre, votre deuxième différence ne peut être que le prénom est de longueur 30 dans l'un et 23 dans l'autre.

b) Le fichier suivant est-il valide ? OUI - NON'

```
<?xml version="1.0" encoding="UTF-8"?>
<habit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="brocante.xsd">
    <etat est="bon" />
    <prix>5</prix>
</habit>
```

Expliquez **brèvement** pourquoi :

4. Les recettes

Voici un fichier *recettes.xsd*

recettes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema version="1.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="recettes" type="recettesType" />
  <xsd:complexType name="recettesType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="recette" type="recetteType" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="recetteType">
    <xsd:sequence>
      <xsd:element name="entete" type="enteteType" />
      <xsd:element maxOccurs="unbounded" name="ingredients"
        type="ingredientsType" />
      <xsd:element maxOccurs="unbounded" name="instruction"
        type="xsd:string" />
    </xsd:sequence>
    <xsd:attribute name="type" type="typeType" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="ingredientsType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="ingredient"
        type="ingredientType" />
    </xsd:sequence>
    <xsd:attribute name="nbPers" type="xsd:int" use="required"/>
    <xsd:attribute name="pour" type="xsd:string" use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="ingredientType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="qte" type="xsd:double"
          use="optional"/>
        <xsd:attribute name="unite" type="uniteType" use="optional" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
```

```

        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:simpleType name="typeType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="entree"/>
            <xsd:enumeration value="plat"/>
            <xsd:enumeration value="dessert"/>
            <xsd:enumeration value="cocktail"/>
            <xsd:enumeration value="soupe"/>
            <xsd:enumeration value="amuseBouche"/>
            <xsd:enumeration value="sauce"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="uniteType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="cs" />
            <xsd:enumeration value="cc" />
            <xsd:enumeration value="verre" />
            <xsd:enumeration value="verres" />
            <xsd:enumeration value="pincee" />
            <xsd:enumeration value="pinces" />
            <xsd:enumeration value="tige" />
            <xsd:enumeration value="tiges" />
            <xsd:enumeration value="gr" />
            <xsd:enumeration value="kg" />
            <xsd:enumeration value="l" />
            <xsd:enumeration value="cl" />
            <xsd:enumeration value="dl" />
            <xsd:enumeration value="ml" />
            <xsd:enumeration value="gousse" />
            <xsd:enumeration value="gousses" />
            <xsd:enumeration value="feuille" />
            <xsd:enumeration value="feuilles" />
            <xsd:enumeration value="rondelle" />
            <xsd:enumeration value="rondelles" />
            <xsd:enumeration value="cm" />
            <xsd:enumeration value="bouquet" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="enteteType">
        <xsd:all>
            <xsd:element name="titre" type="xsd:string" />
            <xsd:element name="preparation" type="xsd:string"
                minOccurs="0" />
            <xsd:element name="cuisson" type="xsd:string" minOccurs="0" />
            <xsd:element name="remarque" type="xsd:string" minOccurs="0"
        />
            <xsd:element name="marinade" type="xsd:string" minOccurs="0"
        />
        </xsd:all>
    </xsd:complexType>
</xsd:schema>

```

On vous demande de répondre aux questions suivantes :

1. Que signifie xsd présent devant chaque élément du document ?

2. Quel est l'élément racine de ce fichier *recettes.xsd* ?

3. Quels sont les éléments facultatifs ? Quels sont les attributs facultatifs ?

4. Précisez si les affirmations suivantes sont correctes et justifiez :

On peut déclarer les ingrédients après les instructions.	
Il y a toujours un entête	
recettes est l'élément racine de tout document xml qui référencera le document XSD car il s'agit du premier élément déclaré dans le fichier xsd.	
Il y a toujours au moins un ingrédient dans la liste des ingrédients de chaque recette	
Le type d'une recette peut être un plat et une entrée	