

62-31.3 XML

[XML-02-E03]

XSD – Type simple personnalisé

Solution

[Cédric Benoit]

Exercice 1 – Création de types simples personnalisés

Elaborer pour chaque type de donnée ci-dessous un type simple en utilisant (au mieux) les schémas XML.

Regrouper toutes vos définitions dans un fichier XSD.

Elaborer un fichier XML permettant de tester et de vérifier vos définitions.

Exemple :

- Un entier entre 1 et 10 [1 - 10]

```
<xs:simpleType name="entierLimite">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1" />
    <xs:maxInclusive value="10" />
  </xs:restriction>
</xs:simpleType>
```

- Un texte sans espaces comportant exactement 16 caractères

```
<xs:simpleType name="Texte16caracteres">
  <xs:restriction base="xs:NMTOKEN">
    <xs:length value="16"/>
  </xs:restriction>
</xs:simpleType>
```

- Un texte quelconque comportant au plus 32 caractères

```
<xs:simpleType name="TexteMax32caracteres">
  <xs:restriction base="xs:string">
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>
```

- Un texte quelconque qui préserve les espaces multiples mais pas les sauts de ligne

```
<xs:simpleType name="TexteSansSautsDeLigne">
  <xs:restriction base="xs:normalizedString"/>
</xs:simpleType>
```

ou

```
<xs:simpleType name="TexteSansSautsDeLigne2">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="replace"/>
  </xs:restriction>
</xs:simpleType>
```

- Un nombre entier entre -100 et 1000 [-100 - 1000[

```
<xs:simpleType name="Entier-100Et1000">
  <xs:restriction base="xs:short">
    <xs:minInclusive value="-100"/>
    <xs:maxExclusive value="1000"/>
  </xs:restriction>
</xs:simpleType>
```

- Un nombre réel strictement supérieur à 10

```
<xs:simpleType name="ReelSuperieur10">
  <xs:restriction base="xs:float">
    <xs:minExclusive value="10"/>
  </xs:restriction>
</xs:simpleType>
```

- Un nombre entier qui peut prendre les valeurs 1,5,16,21 ou 34

```
<xs:simpleType name="Valeur_1_5_16_21_34">
  <xs:restriction base="xs:byte">
    <xs:enumeration value="1"/>
    <xs:enumeration value="5"/>
    <xs:enumeration value="16"/>
    <xs:enumeration value="21"/>
    <xs:enumeration value="34"/>
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="Valeur_1_5_16_21_34">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="1"/>
    <xs:enumeration value="5"/>
    <xs:enumeration value="16"/>
    <xs:enumeration value="21"/>
    <xs:enumeration value="34"/>
  </xs:restriction>
</xs:simpleType>
```

- Un nombre entier entre 10 et 20 ou entre 40 et 50 [10 - 20] [40 - 50]

```
<xs:simpleType name="EntierEntre10et20">
  <xs:restriction base="xs:byte">
    <xs:minInclusive value="10"/>
    <xs:maxInclusive value="20"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="EntierEntre40et50">
  <xs:restriction base="xs:byte">
    <xs:minInclusive value="40"/>
    <xs:maxInclusive value="50"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="EntierEntre10et20ou40et50">
  <xs:union memberTypes="EntierEntre10et20 EntierEntre40et50"/>
</xs:simpleType>
```

ou

```
<xs:simpleType name="EntierEntre10et20ou40et50Solution2">
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:byte">
        <xs:minInclusive value="10"/>
        <xs:maxInclusive value="20"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:byte">
        <xs:minInclusive value="40"/>
        <xs:maxInclusive value="50"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

ou aussi avec les mêmes exemples ci-dessus, mais remplacer xs:byte par xs:integer

- Un nombre décimal avec 5 chiffres après la virgule

```
<xs:simpleType name="Decimal5ChiffresApresVirgule">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="5"/>
  </xs:restriction>
</xs:simpleType>
```

- Un numéro de série de la forme c-cc-ccc-cccc (c = caractère comme lettre, nombre, espace, etc.)

```
<xs:simpleType name="NumeroSerie">  
    <xs:restriction base="xs:NMTOKEN">  
        <xs:pattern value=".\\-\\.\\-...\\-...."/>  
    </xs:restriction>  
</xs:simpleType>
```

ou si le numéro devrait contenir que des chiffres (ce n'était pas précisé dans la donnée, il y a les possibilités montrées ci-dessous:

```
<xs:simpleType name="NumeroSerie2">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\-\d{2}\-\d{3}\-\d{4}"/>
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="NumeroSerie2">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]\-[0-9]{2}\-[0-9]{3}\-[0-9]{4}"/>
  </xs:restriction>
</xs:simpleType>
```

- Un nombre entier comportant le signe et 4 chiffres

```
<xs:simpleType name="EntierSigneEt4Chiffres">
  <xs:restriction base="xs:short">
    <xs:pattern value="[+][0-9]{4}"/>
  </xs:restriction>
</xs:simpleType>
```

- Un texte parmi les possibilités suivantes : Petit MoyenGrand

```
<xs:simpleType name="PetitMoyenOuGrand">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="Petit"/>
    <xs:enumeration value="Moyen"/>
    <xs:enumeration value="Grand"/>
  </xs:restriction>
</xs:simpleType>
```

- Un texte entre 5 et 9 caractères composé uniquement de voyelles

```
<xs:simpleType name="Voyelles5a9">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="[AEIOUYaeiouy]{5,9}"/>
  </xs:restriction>
</xs:simpleType>
```

</xs:simpleType>

- Une donnée pouvant être soit une date, soit un texte.

<xs:simpleType name="DateText">

<xs:union memberTypes="xs:date xs:string"/>

</xs:simpleType>

Exercice 2 – Adaptation des schémas XSD de l'exercice XML-02-E02

Reprendre les exercices de gestion des clients avec leurs commandes et produits et l'arbre généalogique avec les descendants, réalisés dans l'exercice XML-02-E02 avec leurs document XML et schéma XML (XSD). Pour chacun de ces deux exercices, veuillez analyser et vérifier si un type simple pourrait être créé dans les schémas XML (XSD) correspondant, pour améliorer la gestion de vos données.

Pour chaque document XML, veuillez « jouer » avec les données XML en ajoutant, modifiant, et supprimant des données, afin de vérifier et de valider, si les types de données attribués sont corrects par rapport à ce que vous vouliez pouvoir saisir.

Pour le prochain cours

- Rassembler vos fichiers XML et XSD dans une archive (format zip ou 7z)
- Nommer votre archive selon le modèle : XML-02-E03_classe_nom_prenom
- Mettez l'archive dans le dossier "Devoir" de Cyberlearn correspondant à votre classe (en cas de problème vous pouvez toujours envoyer l'archive par e-mail à l'adresse e-mail : cedric.benoit@he-arc.ch)
- Date limite : avant le début de la prochaine leçon
- Au début de la prochaine leçon, un ou deux étudiants seront pris au hasard et présenteront leur travail en classe.