

62-31.3 XML

XML-02-P03-Creation_Types_Simples

Cédric Benoit

Plan du cours

- Présentation et organisation
- Introduction à XML
- Schémas XML
- **Types de données**
 - Types prédéfinis
 - **Création de types simples**

Points abordés

- Introduction
- Dérivation par Restriction
- A partir des types existants (prédéfinis ou dérivés)
- Avec des facettes (facets) ou contraintes supplémentaires
- Exemples
- ...

Introduction

- Création de nouveaux types (simple).
- Un des mécanismes pour définir un nouveau type en "dérivant" un type existant est la restriction avec des contraintes supplémentaires.
- Exemple :

```
<xs:simpleType name="typeCote">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="gauche"/>  
    <xs:enumeration value="droite"/>  
  </xs:restriction>  
</xs:simpleType>...
```

Restriction (1)

- Ajout de contraintes en limitant les valeurs possibles.
- Balise :
`<xs:restriction...>`
- Le type de base est nommé avec l'attribut *base*.
`<xs:restriction base="type de donnée">...`
`</xs:restriction>`
- Exemple:
`<xs:restriction base="xs:string">...`
`</xs:restriction>`

Restriction (2)

- Les types simples ont des facettes (facets) ou des contraintes.
- Une facette est un aspect définitoire unique contribuant à déterminer l'ensemble de valeurs d'un type simple.
- *length*, *minInclusive* et *maxInclusive* sont des facettes courantes.
- Toutes les facettes d'un type simple définissent l'ensemble de valeurs autorisées pour ce type.

Restriction – Exemple (1)

Exemple de déclaration en une étape. Attention à ...Inclusive et ...Exclusive (Conseiller d'utiliser les mêmes termes).

```
<xs:simpleType name="TypeMonEntier">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="-5"/>  
    <xs:maxExclusive value="10"/>  
  </xs:restriction>  
</xs:simpleType>
```

Valeurs possibles : -5 à 9 (10 n'est pas autorisé)

Restriction – Exemple (2)

Déclaration en deux temps (pas rencontré en pratique)

```
<xs:simpleType name="TypeMonEntier">  
  <xs:restriction>  
    <xs:simpleType>  
      <xs:restriction base="xs:integer">  
        <xs:maxExclusive value="10"/>  
      </xs:restriction>  
    </xs:simpleType>  
    <xs:minInclusive value="-5"/>  
  </xs:restriction>  
</xs:simpleType>
```


Facettes (1)

Les facettes qui vont suivre sont valables pour les types prédéfinis ci-dessous :

ENTITY

language

NMTOKEN

base64Binary

NOTATION

ID

Name

token

hexBinary

QName

IDREF

NCName

anyURI

Facettes (2)

enumeration → permet de définir une liste de valeurs possibles.

Exemple : liste de liens url possibles

```
<xs:simpleType name="TypeSchemaW3C">  
  <xs:restriction base="xs:anyURI">  
    <xs:enumeration value="http://www.w3.org/2000/XMLSchema"/>  
    <xs:enumeration value="http://www.w3.org/2001/XMLSchema"/>  
    <xs:enumeration value="http://www.w3.org/2002/XMLSchema"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes (3)

length → permet de définir une taille fixe en nombre de caractères ou d'octets

Exemple : on désire avoir des mots fixes de 8 caractères

```
<xs:simpleType name="typeGenre">  
  <xs:restriction base="xs:NMTOKEN">  
    <xs:length value="8"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes (4)

maxLength → permet de définir une taille maximale en caractères ou octets

Exemple : image de 1024 octets au maximum

```
<xs:simpleType name="TypeBinaryImage">  
  <xs:restriction base="xs:hexBinary">  
    <xs:maxLength value="1024"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes (5)

minLength → permet de définir une taille minimale en caractères ou octets

Exemple : des noms d'une longueur minimale de 6 caractères

```
<xs:simpleType name="TypeFixName">  
  <xs:restriction base="xs:Name">  
    <xs:minLength value="6"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes (6)

pattern → permet de décrire les valeurs admissibles à l'aide d'un motif ou modèle (expressions régulières qui seront étudiées plus tard).

Exemple : les données de type TypeHttpURI devront commencer par "http://" + nom du site Internet

```
<xs:simpleType name="TypeHttpURI">  
  <xs:restriction base="xs:anyURI">  
    <xs:pattern value="http://.*"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes (7)

whiteSpace → permet de gérer les espaces blancs
x20(espace), #x9(tab), #xA(saut de ligne), #xD(retour
chariot).

- Les valeurs possibles sont :
 - preserve (le processeur XML ne change rien)
 - replace (le processeur XML remplace tabulation, retour de ligne, etc. par des espaces)
 - collapse (le processeur XML supprime les espaces devant et derrière le texte, remplace tous les espaces multiples par un seul espace)

Exemple : ...<xs:whiteSpace value="preserve"/>...

Facettes (8)

Exemple : on veut une remarque de 100 caractères sans espace multiple.

```
<xs:simpleType name="typeRemarque">  
  <xs:restriction base="xs:string">  
    <xs:maxLength value="100"/>  
    <xs:whiteSpace value="collapse"/>  
  </xs:restriction>  
</xs:simpleType>
```


Facettes (9)

Exemple de valeur saisie :

<remarque> Aujourd'hui, il faut
beau </remarque>

Le processeur XML devrait interpréter la valeur de la
manière suivante :

<remarque>Aujourd'hui, il faut beau</remarque>

Facettes – nombres réels (1)

Les facettes qui vont suivre sont valables pour les types prédéfinis suivants :

- double
- float

Facettes – nombres réels (2)

enumeration → permet de définir une liste de valeurs possibles.

Exemple: on désire pouvoir saisir que les nombre 1.000, 5.000 et 9.9999.

```
<xs:simpleType name="typeNombriereel">  
  <xs:restriction base="xs:float">  
    <xs:enumeration value="1.0000"/>  
    <xs:enumeration value="5.0000"/>  
    <xs:enumeration value="9.9999"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – nombres réels (3)

Les valeurs suivantes sont conformes :

`<enumeration>1.0000</enumeration>`

`<enumeration>5.0000</enumeration>`

`<enumeration>9.9999</enumeration>`

Facettes – nombres réels (4)

maxExclusive → permet de définir la valeur maximale (limite exclue)

Exemple : possibilité de saisir des nombre réels jusqu'à 9.999999999999..., mais pas 10.

```
<xs:simpleType name="typeFloatExclusive">  
  <xs:restriction base="xs:float">  
    <xs:maxExclusive value="10"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – nombres réels (5)

maxInclusive → permet de définir la valeur maximale (limite incluse)

Exemple : possibilité de saisir des nombre réels jusqu'à 10 inclu.

```
<xs:simpleType name="typeFloatInclusive">  
  <xs:restriction base="xs:float">  
    <xs:maxInclusive value="10"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – nombres réels (6)

minExclusive → permet de définir la valeur minimale (limite exclue).

Exemple : possibilité de saisir des nombre réels depuis 2.6, en excluant 2.5.

```
<xs:simpleType name="typeFloatExclusive">  
  <xs:restriction base="xs:float">  
    <xs:minExclusive value="2.5"/>  
  </xs:restriction>  
</xs:simpleType>
```

- **Facettes – nombres réels (7)**

minInclusive → permet de définir la valeur minimale (limite incluse).

Exemple : possibilité de saisir des nombre réels depuis 2.5 inclu.

```
<xs:simpleType name="typeFloatInclusive">  
  <xs:restriction base="xs:float">  
    <xs:minInclusive value="2.5"/>  
  </xs:restriction>  
</xs:simpleType>
```


Facettes – nombres réels (8)

pattern → permet de décrire les valeurs admissibles à l'aide d'un motif ou modèle (expressions régulières qui seront étudiées plus tard).

Exemple : possibilité de saisir un nombre réel sans décimal (1) ou avec deux décimales (1.00).

```
<xs:simpleType name="typeFloat">  
  <xs:restriction base="xs:float">  
    <xs:pattern value="[0-9](\.[0-9]{2})?" />  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – dates et heures (1)

Les facettes suivantes sont valables pour les type prédéfinis suivants :

date

dateTime

duration

gDay

gMonth

gMonthDay

gYear

gYearMonth

time

Facettes – dates et heures (2)

enumeration → permet de définir une liste de valeurs possibles.

Exemple: on désire pouvoir saisir une liste d'années précise.

```
<xs:simpleType name="TypeHistoireSuisseModerne">  
  <xs:restriction base="xs:gYear">  
    <xs:enumeration value="1848"/>  
    <xs:enumeration value="1914"/>  
    <xs:enumeration value="1939"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – dates et heures (3)

D'autres facettes décrites dans les nombres réels peuvent être aussi utilisées dans les types prédéfinis liés aux dates et heures.

- minInclusive
- minExclusive
- maxInclusive
- maxExclusive
- pattern

Facettes – numérique (1)

Les facettes suivantes sont valables pour les types prédéfinis suivants :

byte

int

integer

long

short

negativeInteger

nonNegativeInteger

nonPositiveInteger

positiveInteger

unsignedByte

unsignedInt

unsignedLong

unsignedShort

Facettes – numérique (2)

Les types acceptent toutes les facettes des types des nombre réels (xs:float et xs:double).

- enumeration
- pattern
- minInclusive
- maxInclusive
- ...

Facettes – numérique (3)

totalDigits → permet de définir le nombre maximal de chiffres.

Exemple: on désire saisir un nombre à 5 chiffres maximum

```
<xs:simpleType name="TypeTotalChiffre">  
  <xs:restriction base="xs:integer">  
    <xs:totalDigits value="5"/>  
  </xs:restriction>  
</xs:simpleType>
```

Facettes – numérique (4)

Le type prédéfini *decimal* accepte toutes les facettes des nombres entiers et en plus `xs:fractionDigits` : qui permet de définir le nombre maximal de décimal.

Exemple: on désire saisir jusqu'à 3 décimales -> 2.1, 2.12 ou 2.123, mais pas 2.1234.

```
<xs:simpleType name="TypeFractions">  
  <xs:restriction base="xs:decimal">  
    <xs:fractionDigits value="3"/>  
  </xs:restriction>  
</xs:simpleType>
```


Facettes – numérique (5)

Le type prédéfini *boolean* supporte une seule facette qui est `xs:pattern`. Cette facette peut être utilisée pour déterminer la valeur associée au type *boolean* (oui / 1 / vrai ...)

```
<xs:simpleType name="TypeTrueOrFalse">  
  <xs:restriction base="xs:boolean">  
    <xs:pattern value="true"/>  
    <xs:pattern value="false"/>  
  </xs:restriction>  
</xs:simpleType>
```

Exemple d'évolution de restrictions (1)

J'ai un document XML, où je gère des personnes
comme ceci :

```
<Personnes>
```

```
...
```

```
  <Personne genre="Monsieur">
```

```
    <Nom>Dupont</Nom>
```

```
    <Prenom>Jaxques</Prenom>
```

```
  </Personne>
```

```
  <Personne genre="Madame">
```

```
    <Nom>Robert</Nom>
```

```
    <Prenom>Josette</Prenom>
```

```
  </Personne>
```

```
...
```

```
</Personnes>
```

Exemple d'évolution de restrictions (2)

On aimerait pour l'attribut "genre" pouvoir saisir que les valeurs suivantes: : soit Monsieur, Madame ou Mademoiselle.

Actuellement dans le schéma XML (XSD), l'attribut "genre" est déclaré comme ceci:

```
<xs:attribute name="genre" type="xs:string"/>
```

(Si l'attribut "genre" était un élément, <genre>Monsieur</genre> qui se traduit dans le schema XML par `<xs:element name="genre" type="xs:string"/>`, la même manière d'affecter un type simple personnalisé est utilisée)

Question: comment fait-on pour respecter la contrainte de l'attribut "genre" ?

Exemple d'évolution de restrictions (3)

...

```
<xs:simpleType name="typeGenre">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="Mademoiselle"/>  
    <xs:enumeration value="Madame"/>  
    <xs:enumeration value="Monsieur"/>  
  </xs:restriction>  
</xs:simpleType>
```

...

```
<xs:attribute name="genre" type="typeGenre"/>
```

...

Merci pour votre attention !

