

62-31.3 XML

[XML-02-E04]
XSD – Patterns

Solution

[Cédric Benoit]

Exercice 1 – Expressions régulières avec les patterns

Elaborer pour chaque type de donnée ci-dessous un type simple correspondant en utilisant les expressions régulières dans le schéma XML (XSD).

Regrouper toutes vos définitions dans un fichier XSD.

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

1. Une expression régulière permettant de saisir un nombre entier comportant le signe (- et +) et d'une longueur de 4 chiffres.

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

2. Que reconnaît l'expression régulière suivante : `a(ba)*[bc]+` donnez une explication .

ab
ac
abababac

Remarque : le résultat « aac » car implicitement dans le cas des patterns, l'expression ci-dessus implique que l'on commencer par le caractère « a » et terminer par « b » ou « c ». Dans un autre contexte comme Javascript, cela serait différent. Pour exprimer le même genre de contrainte que ci-dessus, il faudrait ajouter les caractères « ^ » au début de l'expression régulière et « \$ » à la fin de l'expression régulière comme suit : `^a(ba)*[bc]+$`

3. L'expression régulière doit permettre de saisir un numéro de série qui correspond à : SERIE-CC.C.CCC (C = chiffre allant de 0 à 9).

```
<xs:simpleType name="Serie">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="SERIE\-[0-9]{2}\.[0-9]\.[0-9]{3}"/>
  </xs:restriction>
</xs:simpleType>
```

4. Définir un type de données permettant de saisir les types de monnaie CHF ou EUR.

```
<xs:simpleType name="Monnaie">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="CHF|EUR"/>
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="Monnaie">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="(CHF|EUR)"/>
  </xs:restriction>
</xs:simpleType>
```

Remarque : dans les deux cas, on peut saisir CHF et EUR comme cela demandé dans la donnée. Mais avec la 1^{ère} solution, il possible de saisir aussi "CHFEUR", "CHFE", "CHFU", etc. Par contre dans la 2^{ème} solution avec les (...), cela garantit qu'il ne sera possible de saisir que CHF et EUR et pas autre chose.

5. L'expression régulière doit permettre de saisir le texte suivant : sans commentaire ou SANS COMMENTAIRE.

```
<xs:simpleType name="Typecommentaire">
  <xs:restriction base="xs:token">
    <xs:pattern value="sans commentaire|SANS COMMENTAIRE"/>
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="Typecommentaire">
  <xs:restriction base="xs:token">
    <xs:pattern value="(sans commentaire|SANS COMMENTAIRE)"/>
  </xs:restriction>
</xs:simpleType>
```

Remarque : même remarque que pour le point 4..

6. L'expression régulière doit permettre de saisir un nombre réel jusqu'à 4 décimales.

```
<xs:simpleType name="Typefloat">
  <xs:restriction base="xs:float">
    <xs:pattern value="[0-9]+(\.[0-9]{1,4})?">
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="Typefloat">
  <xs:restriction base="xs:float">
    <xs:pattern value="[0-9]+(\.{1,4})?">
  </xs:restriction>
</xs:simpleType>
```

Remarque : même si le point est un caractère joker ".", où on peut tout ce qu'on veut (a, b, 1, 2, !, ?, etc.), comme le type de données est un *float*, il ne sera possible de saisir que des chiffres.

7. L'expression régulière doit reconnaître une expression arithmétique simple sans parenthèse. La chaîne doit commencer par un nombre. Ce nombre est suivi d'un opérateur arithmétique simple (+, -, /, *) lui-même suivi d'un autre nombre (n fois).

```
<xs:simpleType name="Typearithmetique">
  <xs:restriction base="xs:NMTOKEN">
    <xs:pattern value="[0-9]+([\+\/\*][0-9]+)"/>
  </xs:restriction>
</xs:simpleType>
```

Remarques : dans un ensemble [...], il faut utiliser plutôt \- dans le cas où l'on veut que le tiret « - » soit considéré comme caractère et non comme intervalle comme de 0 à 9 [0-9] ou de a à z [a-z]. Bien que certains sites cités plus bas ne font pas la différence, alors que d'autres font cette distinction (Voir support de cours)

8. Ecrivez une expression régulière qui permet de saisir des francs avec des centimes en option (exemple 10, 9.95 ou 0.50).

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

ou

```
<xs:simpleType name="Typedecimal">
  <xs:restriction base="xs:decimal">
    <xs:pattern value="+(\.{2})?">/>
  </xs:restriction>
</xs:simpleType>
```

Remarque : même que pour le point 6 avec le nombre réel. Comme le type de données *decimal* n'accepte que des chiffres, on peut aussi mettre le caractères joker ".".

9. Ecrivez une expression régulière qui permet de saisir : lapin chat ou lapin chien.

```
<xs:simpleType name="Typetexte">
  <xs:restriction base="xs:token">
    <xs:pattern value="lapin (chat|chien)">/>
  </xs:restriction>
</xs:simpleType>
```

10. Proposez deux expressions régulières qui permettent de saisir les chiffres de 1 à 999.

```
<xs:simpleType name="TypeInt1a999">
  <xs:restriction base="xs:integer">
    <xs:pattern value="[1-9][0-9]?[0-9]">/>
  </xs:restriction>
</xs:simpleType>
```

ou

```
<xs:simpleType name="TypeInt1a999">
  <xs:restriction base="xs:integer">
    <xs:pattern value="[1-9][0-9]{0,2}">/>
  </xs:restriction>
</xs:simpleType>
```

Remarque : pour exprimer une même contrainte, il peut y avoir différentes expressions régulières possible. Ci-dessus, pour éviter d'écrire deux fois "[0-9]?", il est possible d'utiliser le quantificateur {} comme ceci : [0-9]{0,2}.

11. Proposez une expression régulière permettant de saisir une date sous format : yyyy-mm-dd (année-mois-jour).

Remarque : comme indiqué dans le support de cours GES-DP170221-XML-02-02-Types_bases-XSD, le format date dans W3C doit respecter le standard ISO 8601, ce qui limite l'utilisation du pattern avec le type. Comme mentionner ci-dessus le type date pourrait suffire pour respecter le format mentionné ci-dessus (yyyy-mm-dd). Cependant, comme indiqué dans le support GES-DP170221-XML-02-02-Types_bases-XSD à la diapositive 44, avec le type date on peut indiquer la zone de temps comme : 2001-10-26+02:00 ou une date négative (avant J.-C.) comme. -2001-10-26. Afin d'être plus stricte au niveau de la date, on peut le pattern ci-dessous.

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

12. Ecrivez une expression régulière permettant de saisir une url commençant par : <http://www>.+ suivi de caractères quelconques.

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

13. Quels sont les méta-caractères contenus dans cette expression : [0-9A-Z%!\?\.\\-]

[] → ensemble
- → 0-9 et A-Z
\\ → \ ? \ . \ -

Sur internet, il existe quelques sites pour vérifier vos expressions régulières, dont voici quelques liens ci-dessous :

- <http://www.annuaire-info.com/outil-referencement/expression-reguliere/>
- <http://www.xul.fr/ecmascript/expression-reguliere-demo.php>
- <http://regexpal.com/>

Exercice 2 – Adaptation des schémas XSD de l'exercice XML-02-E03

Reprendre les exercices de gestion des clients avec leurs commandes et produits, l'arbre généalogique avec les descendants, ainsi que l'exercice de gestion des voitures dans un garage réalisés dans l'exercice XML-02-E03 avec leurs document XML et schéma XML (XSD). Pour chacun de ces trois exercices, veuillez analyser et vérifier si un "pattern" pourrait être créé dans les schémas XML (XSD) correspondant, pour améliorer la gestion de vos données.

Par 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-E04_classe_nom_prenom
- Mettez l'archive dans le dossier "Devoirs" 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.