

## Technologie XML

### 1. Une DTD pour des batiments

Nous admettrons pour ce qui suit qu'une construction, un batiment est soit une salle de cours soit un bureau. Une salle et un bureau sont définis par un numéro et un nom de batiment. Une salle peut contenir, mais ce n'est pas obligatoire, le nom d'une formation ; un bureau contient obligatoirement au moins un nom de personne. Voici par exemple une liste fictive de batiments, fichier `batiments1.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>
<constructions>
  <salles>
    <salle numSalle="205" nomBatiment="L">
      <formation>Licence Physique</formation>
    </salle>
    <salle numSalle="217" nomBatiment="G" />
  </salles>
  <bureaux>
    <bureau numBureau="103" nomBatiment="H">
      <personne>Monsieur HUNAULT</personne>
    </bureau>
  </bureaux>
</constructions>
```

Donner pour ce fichier une grammaire DTD « *minimale et raisonnable* ».

## 2. Un peu de XPATH

Indiquez quelles expressions `xmllint` en mode `--shell` permettraient d'obtenir, si c'est possible :

- la liste de tous les numéros de bureaux ;
- le nombre de salles ;
- la liste des numéros de bureaux situés au premier étage d'un bâtiment, sachant que le premier chiffre du numéro de bureau correspond à l'étage ;
- le nombre de constructions en tout (salles + bureaux) ;
- une liste triée par ordre alphabétique des notations résumées de salles et bureaux, comme G217, H103...

## 3. Un autre fichier XML de bâtiments

Donner le contenu du fichier XML nommé `batiments2.xml` qui contient les mêmes informations que le fichier précédent `batiments1.xml` mais structuré selon la grammaire XSD nommée `batiments2.xsd` fournie en annexe.

## 4. Un peu de XSL

On voudrait disposer d'une transformation XSL qui produit une liste triée par ordre alphabétique des notations résumées de salles seulement, comme G217, L205 à partir du document de la question 1 avec indication au passage du contenu du bâtiment (formation).

On désire juste produire un fragment XHTML valide avec ces informations comme présenté sur la page suivante (mais pas toute la page Web). Donner le code XSL correspondant. On pourra s'appuyer sur le fichier `stdWeb2.xsl` afin de ne pas avoir trop de code à écrire sur la copie.

A défaut de fournir le code XSL complet et exact qui effectue la transformation, on pourra fournir un fichier simplifié avec des explications.

Fragment XHTML à produire, on ne sera pas obligé(e) de reproduire exactement l'indentation et les retours à la ligne :

```
<table border="1" class="bgcolor_white collapse"
      cellpadding="20" cellspacing="5">
  <tr>
    <td>G217</td>
    <td></td>          ### ou <td />
  </tr>
  <tr>
    <td>L205</td>
    <td>Licence Physique</td>
  </tr>
</table>
```

## 5. Discutons...

Essayez de répondre à la question suivante :

*Etait-ce vraiment important de migrer la partie traitement de textes de la bureautique classique (Word, Open Office Write...) vers des archives contenant du XML ?*

Votre réponse devra essayer de mettre en évidence votre culture naissante, votre recul et votre esprit de synthèse en matière de modélisation et de traitement de l'information.

Cette réponse devra faire 10 lignes au minimum, sans limite de maximum. On utilisera au moins 3 mots de 4 syllabes ou plus pour « transmettre un contenu rédactionnel fort ».

## ANNEXE : fichier batiments2.xsd

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

  <xs:element name="constructions">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="construction"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="construction">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nomBatiment"/>
        <xs:sequence minOccurs="0">
          <xs:element ref="numBureau"/>
          <xs:element ref="personne"/>
        </xs:sequence>
        <xs:element minOccurs="0" ref="numSalle"/>
        <xs:element minOccurs="0" ref="formation"/>
      </xs:sequence>
      <xs:attribute name="type" use="required" type="xs:string"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="nomBatiment" type="xs:string"/>
  <xs:element name="numBureau" type="xs:integer"/>
  <xs:element name="personne" type="xs:string"/>
  <xs:element name="numSalle" type="xs:integer"/>
  <xs:element name="formation" type="xs:string"/>

</xs:schema>
```

# CORRIGÉ RAPIDE

## 1. Une DTD pour des batiments

```
<!ELEMENT constructions (salles,bureaux) >
<!ELEMENT salles        (salle)*      >
<!ELEMENT bureaux       (bureau)*     >

<!ELEMENT salle         (formation)?  >
<!ATTLIST salle         nomBatiment  NMTOKEN #REQUIRED >
<!ATTLIST salle         numSalle     CDATA   #REQUIRED >

<!ELEMENT bureau        (personne)+    >
<!ATTLIST bureau        nomBatiment  NMTOKEN #REQUIRED >
<!ATTLIST bureau        numBureau    CDATA   #REQUIRED >

<!ELEMENT formation     (#PCDATA)      >
<!ELEMENT personne      (#PCDATA)>    >
```

## 2. Un peu de XPATH

```
ls //bureau/@numBureau           # la liste de tous les numéros de bureaux
xpath count(//salle)             # le nombre de salles

ls //bureau/@numBureau[starts-with(.,"1")] # bureaux situés au premier étage
xpath count(//salle | //bureau)     # le nombre de constructions en tout
```

ce n'est pas possible d'avoir une liste triée par ordre alphabétique des notations résumées de salles et bureaux, comme G217, H103 avec xmllint en mode shell.

### 3. Un autre fichier XML de batiments

```
<?xml version="1.0" ?>
<constructions>

  <construction type="salle">
    <nomBatiment>G</nomBatiment>
    <numSalle>217</numSalle>
  </construction>

  <construction type="salle">
    <nomBatiment>L</nomBatiment>
    <numSalle>205</numSalle>
    <formation>Licence Physique</formation>
  </construction>

  <construction type="bureau">
    <nomBatiment>H</nomBatiment>
    <numBureau>103</numBureau>
    <personne>Monsieur HUNAULT</personne>
  </construction>

</constructions>
```

## 4. Un peu de XSL

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- batiments.xml -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:import href="stdWeb2.xml" />

<xsl:template match="/">
  <xsl:call-template name="debutTableau" />
  <xsl:call-template name="sdl" />
  <xsl:apply-templates select="//salle">
    <xsl:sort select="@nomBatiment" />
    <xsl:sort select="@numSalle" type="numeric" />
  </xsl:apply-templates>
  <xsl:call-template name="sdl" />
  <xsl:call-template name="finTableau" />
</xsl:template>

<xsl:template match="salle">
  <tr>
    <td>
      <xsl:value-of select="./@nomBatiment" />
      <xsl:value-of select="./@numSalle" />
    </td>
    <td>
      <xsl:value-of select="./formation" />
    </td>
  </tr>
</xsl:template>

</xsl:stylesheet>
```