

Utilisation de XML, DTD, XSLT et JavaScript

### Les outils utilisés

- ★ `xmllint` pour vérifier et valider votre document XML :

- ◇ pour vérifier que votre document XML est correct :

```
xterm
$ xmllint mon_fichier.xml
```

- ◇ pour vérifier que votre document est « valide » par rapport à un DTD :

```
xterm
$ xmllint mon_fichier.xml -valid
```

- ★ `xmllint` pour parcourir l'arborescence d'un document XML :

```
xterm
$ xmllint --shell mon_fichier.xml
/ >
```

Les commandes sont :

- ◇ `base` : retourne l'URL du document XML qu'on est en train d'inspecter ;
- ◇ `du` : affiche la structure du document ;
- ◇ `ls expXPath` : affiche les enfants des noeuds trouvés par l'expression `expXPath` (sans leur contenu !)
- ◇ `cat expXPath` : affiche le contenu de l'ensemble des noeuds trouvés par l'expression `expXPath` ;
- ◇ `cd expXPath` : change le noeud contexte vers le noeud trouvé par l'expressions `expXPath` *attention, expXPath doit retourner un seul noeud* ;
- ◇ `pwd` : affiche le chemin de la racine qui mène au noeud contexte.

- ★ `xsltproc` : pour appliquer un fichier XSLT sur un document XML :

```
xterm
$ xsltproc -o transforme.html mon_xslts.xsl mon_fichier.xml
```

Ici, on applique un document *xsl* sur un document *XML* et qui va produire un document *HTML*.

```
xterm
$ xsltproc mon_document.xml -o transforme.html
```

Ici, le document *XML* contient une référence sur le fichier *XSL* associé.

- ★ Firefox pour afficher vos documents XML, CSS, XSL, en demandant l'affichage de la « console d'erreurs »

### XML et DTD

- 1 – On veut créer un document de gestion de contact en XML regroupant pour chaque personne :

- Nom
- Prénom
- Sexe
- Civilité
- Adresse électronique principale

#### Questions :

- a. Comment allez vous gérer les différents attributs d'une personne ?  
Sous forme d'« attributs » ou de « balises » ?
- b. Écrivez un tel fichier au format XML et vérifiez avec `xmllint` qu'il est bien formé (vous rentrerez 5 personnes dans votre fichier).

- 2 – On veut maintenant étendre le fichier précédent pour pouvoir tenir compte des différents « contacts » d'une personne avec son identifiant :

- ☐ pour chaque service de chat (Skype, Facetime, Telegram, *etc.*);
- ☐ pour chaque réseau social (Twitter, Facebook, Instagram, *etc.*);
- ☐ pour chaque messagerie globale (GMail, Hotmail, *etc.*).

**Questions :**

- a. Comment allez vous gérer les différents contacts de la personne, sachant qu'une personne n'utilise pas tous ces contacts et que d'autre part on veut faciliter leur gestion ?
- b. Améliorez votre fichier XML précédent ;
- c. Créez un DTD pour ce format de document XML ;
- d. Validez votre fichier par rapport à cette DTD avec `xmllint`.

■ ■ ■ **XML, XSL**

- 3 – Dans cet exercice, on va travailler sur un document XML contenant le programme de la TNT.

Vous devez récupérer le fichier `tnt.xml`. *Attention, le fichier prend plus de 15Mo décompressé.*

```
xterm
$ wget https://p-fb.net/licence3/compilation2/files/tnt.zip
$ unzip tnt.zip
```

**Manipulations :**

- a. Examinez le contenu à l'aide de « `gedit` » ;
- b. Parcourez la structure du document XML à l'aide de « `xmllint --shell tnt.xml` ».

Essayez les commandes :

```
xterm
ls
cd tv
ls
cd programme[1]
ls
cat title
ls //title
ls //programme[@channel='C1.telerama.fr']/title
```

À quoi correspond `C1.telerama.fr` ?

- c. Appliquez le fichier `xsl` suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8" />
<xsl:template match="/tv">
  <html>
    <body>
      <h1> Liste des programmes </h1>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
<xsl:template match="programme[@channel='C1.telerama.fr']">
  <p>
    <HR/>
    <xsl:value-of select="title"/>
    <xsl:value-of select="desc"/>
  </p>
</xsl:template>
</xsl:stylesheet>
```

Que fait-il ?

d. Appliquez le fichier xsl suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="UTF-8" />
<xsl:template match="/tv">
  <html>
    <body>
      <h1> Liste des programmes </h1>
      <xsl:for-each select="programme[@channel='C1.telerama.fr']">
        <p>
          <HR/>
          <xsl:value-of select="title"/>
          <xsl:value-of select="desc"/>
        </p>
      </xsl:for-each>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Quelle différence avec le précédent ?

e. On ne veut afficher que les « magazine » :

- ◇ comment est exprimé la catégorie d'un programme dans le fichier XML ?
- ◇ donnez deux méthodes différentes pour n'afficher que les « programmes » de catégorie « magazine » :
  - \* avec « xsl:if » ;
  - \* directement avec XPath ;

f. On voudrez afficher l'image donnée dans l'élément `icon`, c-à-d mettre sa valeur dans une balise html `IMG` :

```
1 
```

g. Améliorez l'affichage de tous ces éléments.

## ■ ■ ■ XML, CSS et XSL

4 – On veut afficher le contenu du fichier des questions 1 et 2 dans un navigateur Web.

### Serveur Web temporaire

*Pour mettre à disposition le contenu du répertoire courant en accès HTTP dans votre navigateur, vous lancerez dans le répertoire contenant les fichiers la commande suivante :*

```
python3 -m http.server
```

*Puis, dans votre navigateur Web, vous entrerez l'URL suivante : `http://localhost:8000` pour accéder à vos fichiers.*

- a. Donnez une version simple avec l'utilisation d'un CSS dédié ;
- b. Écrivez un fichier au format XSL pour afficher le contenu de ce fichier sous forme de tableau HTML.
- c. Modifiez votre fichier XSL pour trier les contacts suivant le nom (le `data-type` à utiliser est « "text" »).

## ■ ■ ■ XML, AJAX et JQuery

Soit le contenu du fichier « livres.xml » :

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <liste_livres titre="Science Fiction">
3   <livre>
4     <titre isbn="2290055956">Le cycle des robots</titre>
5     <auteur>Isaac Asimov</auteur>
6   </livre>
7   <livre>
8     <titre isbn="2290312983">Le monde des non-A</titre>
9     <auteur>Van Vogt</auteur>
10  </livre>
11  <livre>
12    <titre isbn="284172039X">La huitième couleur</titre>
13    <auteur>Terry Pratchett</auteur>
14  </livre>
15 </liste_livres>
```

et le fichier HTML « affiche\_livres.html » :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>JQuery XML</title>
5     <script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jque
ry/1.11.0/jquery.js"></script>
6   </head>
7   <body>
8     <h2> Liste des livres </h2>
9     <div id="contenu"></div>
10    <input type="button" id="requete" value="Ajax !">
11
12    <script type="text/javascript">
13      // traite un livre
14      function traiter_un_livre(index, valeur) {
15        var auteur = $(valeur).find('auteur').text();
16        var isbn = $(valeur).find('titre').attr('isbn');
17        var titre = $(valeur).find('titre').text();
18
19        $('#liste_livres').append('<li><strong>' + titre + '</strong>, <em>' + auteur + '</em>,
'+ isbn + '</li>');
20      }
21      // ajoute une liste et traite chaque livre du document XML
22      function traiter_liste_livres(donnees)
23      {
24        $('#contenu').append('<ul id="liste_livres"></ul>').append('<div id="contenu">');
25        $(donnees).find('livre').each(traiter_un_livre);
26      }
27      // Requete AJAX declenchee lors de l'appui sur le bouton
28      $('#requete').click(function() {
29        $.ajax({
30          type: 'GET',
31          url: 'livres.xml',
32          dataType: 'xml',
33          success: function(donnees) { traiter_liste_livres(donnees); }
34        });
35      });
36    </script>
37  </body>
38 </html>
```

### 5 – Questions :

- Testez le comportement du fichier HTML réalisant la requête AJAX et effectuant un affichage du contenu XML.  
*Vous construirez les fichiers livres.xml et affiche\_livres.html dans un répertoire et utiliserez la commande Python donnée plus haut pour partager ce contenu dans votre navigateur.*
- Vous adapterez le fichier HTML pour traiter les données de l'exercice 2 avec un affichage sous forme de table et en ajoutant une icône pour l'identification de la nature du contact.