

1.

Les feuilles de style (CSS - CascadingStyleSheet).

Principe.

- On peut mettre en forme un document XML grâce à une feuille de style CSS, de la même façon qu'on le fait pour une page HTML. Cette méthode est plutôt simple, mais manque de souplesse.

Méthode.

- Dans le fichier XML, on ajoute une référence vers le fichier CSS :

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="fichier.css"?>
<racine>
  ...
</racine>
```

Exemple.

L'exemple suivant est basé sur le fichier `famille.xml`:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="css1.css" type="text/css"?>
<racine>
  <enfant>
    <nom>Loïc</nom>
    <lien>garçon</lien>
    <date>07/11/83</date>
    <data>Le petit qui me dépasse d'une tête.</data>
  </enfant>
  <enfant>
    <nom>Marine</nom>
    <lien>fille</lien>
    <date>20/12/85</date>
    <data>La petite fille chérie à son papa.</data>
  </enfant>
</racine>
```

Voici la feuille de style associée (`css1.css`) :

```
<style type="text/css">
racine , enfant {}
nom {
display: block;
width: 250px;
font-size: 16pt ;
font-family: arial ;
font-weight: bold;
background-color: teal;
color: white;
padding-left: 10px;
}
lien {
display: block;
font-size: 12pt;
padding-left: 10px;
}
date {
display: block;
font-size: 12pt;
color: red ;
font-weight: bold;
padding-left: 10px;
}
data {
display: block;
font-size: 11pt ;
font-style: italic;
font-family: arial ;
padding-left: 10px;
}
</style>
```

Et voici le résultat:

| |
|--|
| Loïc |
| garçon |
| 07/11/83 |
| <i>Le petit qui me dépasse d'une tête.</i> |
| Marine |
| filles |
| 20/12/85 |
| <i>La petite fille chérie à son papa.</i> |

Exercice : Utilisez le fichier *utilisateurs.xml* et créez le fichier *css* afin d'afficher :

| |
|--|
| PASTORE |
| Sébastien |
| spastore2 at free.fr |
| 1973-02-25 |
| DUBUISSON |
| Christophe |
| chrsitophe.dubuisson at laposte.net |
| 1 |
| BONVALOT |
| Romain |
| nikanoru_du_84 at hotmail.com |
| 2 |

2. Les feuilles de style XSL : beaucoup plus performantes que les CSS.

- **XSLT** (sous-partie du XSL qui nous intéresse ici), signifie **eXtended Stylesheet Language Transformation**. **XSLT** permet en effet de transformer un document XML en document HTML (par exemple), pour qu'il soit lisible dans un navigateur. Pour réaliser cette transformation, on dispose de règles et d'instructions complexes proches d'un langage de programmation. XSL offre donc une grande liberté, notamment de trier ou de filtrer les balises que l'on souhaite afficher.

Exemple simple :

Exemple d'un document XML lié à une feuille de style XSL simple :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="bouteille.xsl"?>
<bouteille>
  <marque>Cristaline</marque>
  <composition>calcium 71mg/l, magnésium 5,5mg/l, chlorure 20mg/l, nitrate 1mg/l,
traces de fer.</composition>
  <source>
    <ville>St-Cyr la Source</ville>
    <departement>Loiret</departement>
  </source>
  <code_barre>3274080005003</code_barre>
  <contenance>150cl</contenance>
  <ph>7,45</ph>
</bouteille>
```

Et voici la feuille de style XSL associée :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <head>
      <title>Exemple de sortie HTML</title>
      <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    </head>
    <body>
      <h1>Bouteille de marque <xsl:value-of select="bouteille/marque" /></h1>
      <h2>Composition:</h2>
      <p><xsl:value-of select="bouteille/composition" /></p>
      <h2>Lieu d'origine:</h2>
      <p>Ville de <b><xsl:value-of select="bouteille/source/ville" /></b>, dans le
département <b><xsl:value-of select="bouteille/source/departement" /></b></p>
      <h2>Autres informations</h2>
      <ul>
        <li>Contenance: <xsl:value-of select="bouteille/contenance" /></li>
        <li>pH: <xsl:value-of select="bouteille/ph" /></li>
      </ul>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

L'affichage sera :

Bouteille de marque Cristaline

Composition:

calcium 71mg/l, magnésium 5,5mg/l, chlorure 20mg/l, nitrate 1mg/l, traces de fer.

Lieu d'origine:

Ville de St-Cyr la Source, dans le département Loiret

Autres informations

- Contenance: 150cl

| | |
|---|--|
| <?xml version="1.0"?> | |
| | Le XSL est dérivé du XML. Il est normal que le document XSL commence par la déclaration de document XML, soit <?xml version="1.0"?>. |
| <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> | |
| | <p>La seconde ligne déclare que le document est du XSL <i>extensible stylesheet</i>.</p> <p>L'attribut xmlns fait référence au "namespace" utilisé. Le namespace officiel du W3C est xmlns:xsl="http://www.w3.org/1999/XSL/Transform. Pour la petite histoire 1999 fait référence à l'année d'apparition du concept XSL. Le xmlns (incorrect) de Microsoft IE soit xmlns:xsl="http://www.w3.org/TR/WD-xsl"> est dû au fait que le XSL a été implanté dans Internet Explorer 5.0 alors qu'il n'était encore qu'en cours d'élaboration [workingdraft] par le W3C.</p> |
| <xsl:template match="/"> | |
| | <p>Voilà une balise template et son attribut match.</p> <p>Cette balise template va déterminer un gabarit dans lequel on va transformer des éléments du fichier XML sous une forme que le navigateur pourra afficher.</p> <p>Les éléments du fichier XML sont déterminés par l'attribut match="/". Le slash / entre guillemets signale que sont concernées toutes les balises XML du document associé à partir de la racine [root].</p> |
| <html> <body> | |
| | Début de la partie Html qui servira de support pour l'affichage du document dans le navigateur. Attention, balises en minuscules ! |
| Diverses balises Html et XSL... Par exemple : <xsl:value-of select="chemin d'accès/élément"/> | |
| | La balise <xsl:value-of> sera fréquemment utilisée car elle permet de sélectionner un élément du fichier XML associé pour le traiter dans le fichier XSL. Dans l'attribut select, on détermine le chemin d'accès vers la balise XML souhaitée (puisque le XML est structuré) comme le chemin d'accès de répertoire en sous-répertoire vers un dossier. Attention, on utilise bien ici le "forward slash" soit / . |
| </body> </html> | |
| | Fin de la partie en Html. |
| </xsl:template> | |
| | La fermeture de la balise de template. |
| </xsl:stylesheet> | |
| | Le document XSL se termine obligatoirement par la fermeture de la balise de déclaration de document XSL. |
| Attention ! Pour que ce fichier XSL soit d'une quelconque utilité, il faut encore faire référence, dans le fichier XML au fichier XSL. On ajoutera donc dans le fichier XML : | |
| <?xml-stylesheet type="text/xsl" href="nom_du_fichier_xsl.xml"?> | |
| | Cette balise indique au navigateur qu'une feuille de style [stylesheet] est associée au fichier XML et qu'il doit aller chercher le fichier de style à l'adresse indiquée par l'attribut href. |

Exercice : Créer le fichier XML suivant :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<recette>

    <entete>
        <auteur>Casimir</auteur>
        <titre>Recette du Gloubi-Boulga</titre>
        <remarque>Pour une personne</remarque>
    </entete>

    <procedure>
        Remplir un saladier avec de la confiture de fraises, du chocolat râpé,
        des bananes écrasées, de la moutarde forte, des saucisses de Toulouse
        écrasées tièdes mais crues. Mélanger vigoureusement jusqu'à obtenir une
        bouillie marron-clair. Il est normal qu'il y ait des grumeaux. Les
        proportions sont environ égales pour tous les ingrédients, mais il est
        possible de varier selon les goûts de chacun.
    </procedure>

</recette>
```

Créer une feuille de style XSL permettant à partir de cette fiche recette de produire une page HTML qui :

- a pour titre le contenu de l'élément `titre` ;
- commence par un titre `<h1>` ayant comme contenu le contenu de l'élément `titre` ;
- donne ensuite le nom de l'auteur de la recette ;
- affiche ensuite le mot **Remarque** : puis le contenu de l'élément `remarque` ;
- affiche **Procédure** en niveau `<h2>` ;
- dans un paragraphe, présente la procédure à suivre.

Titre : Recette du Gloubi-Boulga

Auteur:Casimir

Remarque:Pour une personne

Procédure

Remplir un saladier avec de la confiture de fraises, du chocolat râpé, des bananes écrasées, de la moutarde forte, des saucisses de Toulouse écrasées tièdes mais crues. Mélanger vigoureusement jusqu'à obtenir une bouillie marron-clair. Il est normal qu'il y ait des grumeaux. Les proportions sont environ égales pour tous les ingrédients, mais il est possible de varier selon les goûts de chacun.

Exemple avec boucle

a. Introduction

Il arrive que la structure du document XML ne soit pas linéaire, mais fondée par exemple sur des boucles, ou bien comporte un nombre indéterminé de fois un même élément ; c'est d'ailleurs le plus souvent le cas.

On peut ainsi reprendre l'exemple de la bouteille d'eau, qui se présente sous la forme du fichier XML suivant :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="bouteille1.xsl"?>
<bouteille>
  <marque>Cristaline</marque>
  <composition>
    <ion_positif>calcium 71mg/l</ion_positif>
    <ion_negatif>nitrate 1mg/l</ion_negatif>
    <ion_positif>magnésium 5,5mg/l</ion_positif>
    <ion_negatif>chlorure 20mg/l</ion_negatif>
    <autres_materiaux>fer</autres_materiaux>
  </composition>
  <source>
    <ville>St-Cyr la Source</ville>
    <departement>Loiret</departement>
  </source>
  <code_barre>3274080005003</code_barre>
  <contenance>150cl</contenance>
  <ph>7,45</ph>
</bouteille>
```

Cette fois-ci, il faut tenir compte d'un nombre *a priori* indéterminé d'éléments `ion_positif`, par exemple. Il suffit pour cela d'introduire dans la feuille de style un élément `xsl:for-each` avec un attribut `select` permettant d'indiquer les nœuds sur lesquels s'effectue la boucle :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
    <head>
      <title>Exemple de sortie HTML</title>
      <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    </head>
    <body>
      <h1>Bouteille de marque <xsl:value-of select="bouteille/marque" /></h1>
      <h2>Composition:</h2>
      <h3>Ions positifs</h3>
      <ul>
        <xsl:for-each select="bouteille/composition/ion_positif">
          <li><xsl:value-of select="." /></li>
        </xsl:for-each>
      </ul>
      <h3>Ions négatifs</h3>
      <ul>
        <xsl:for-each select="bouteille/composition/ion_negatif">
          <li><xsl:value-of select="." /></li>
        </xsl:for-each>
      </ul>
    </body>
  </html>
</template>
</xsl:stylesheet>
```

```

</ul>
<h3>Autres matériaux</h3>
<ul>
  <xsl:for-each select="//autres_materiaux">
    <li><xsl:value-of select="." /></li>
  </xsl:for-each>
</ul>
<h2>Lieu d'origine</h2>
<p>Ville de <b><xsl:value-of select="bouteille/source/ville" /></b>, dans le
département <b><xsl:value-of select="bouteille/source/departement" /></b></p>
<h2>Autres informations</h2>
<ul>
  <li>Contenance: <xsl:value-of select="bouteille/contenance" /></li>
  <li>pH: <xsl:value-of select="bouteille/ph" /></li>
</ul>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Noter :

1. le réarrangement des entrées : les ions positifs sont de toute manière affichés *avant* les ions négatifs.
2. la sélection des « autres matériaux » avec `select="//autres_materiaux"`. Cela est une des manières de faire référence à un élément dans l'arbre des éléments. Nous verrons la liste de ces sélecteurs plus tard.

Bouteille de marque Cristaline

Composition:

Ions positifs

- calcium 71mg/l
- magnésium 5,5mg/l

Ions négatifs

- nitrate 1mg/l
- chlorure 20mg/l

Autres matériaux

- fer

Lieu d'origine

Ville de St-Cyr la Source, dans le département Loiret

Autres informations

Exercice : Créer le fichier XML suivant :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<recette>
  <entete>
    <auteur>Casimir</auteur>
    <titre>Recette du Gloubi-Boulga</titre>
    <remarque>Pour une personne</remarque>
  </entete>
  <procedure>
    <texte>
      Remplir un saladier avec les ingrédients...
    </texte>
    <liste>
      <item>confiture de fraises</item>
      <item>chocolat râpé</item>
      <item>bananes écrasées</item>
      <item>moutarde forte</item>
      <item>saucisses de Toulouse écrasées tièdes mais crues</item>>
    </liste>
    <texte>
      Mélanger vigoureusement jusqu'à obtenir une bouillie marron-
clair.
      Il est normal qu'il y ait des grumeaux.
    </texte>
    <texte>
      Les proportions sont environ égales pour tous les ingrédients,
mais
      il est possible de varier selon les goûts de chacun.
    </texte>
  </procedure>
</recette>
```

Reprendre l'exercice précédent, mais cette fois-ci en affichant dans la procédure une liste des ingrédients, *puis* les opérations à pratiquer.

Remarque : Il est également possible de réarranger par ordre alphabétique la liste des ingrédients, mais cela n'est possible qu'en appelant des commandes XSL évoluées.

Recette du Gloubi-Boulga

Auteur:Casimir

Remarque:Pour une personne

Procédure

Ingrédients:

- confiture de fraises
- chocolat râpé
- bananes écrasées
- moutarde forte
- saucisses de Toulouse écrasées tièdes mais crues

Opérations:

1. Remplir un saladier avec les ingrédients...
2. Mélanger vigoureusement jusqu'à obtenir une bouillie marron-clair. Il est normal qu'il y ait des grumeaux.
3. Les proportions sont environ égales pour tous les ingrédients, mais il est possible de varier selon les goûts de chacun.

Trier avec le XSL

Le langage XSL permet en quelques mots de trier des données du fichier XML associé en ordre croissant ou décroissant. Ainsi, il suffit d'ajouter L'attribut `order = "ascending|descending"` permet d'obtenir un ordre croissant ou décroissant.. Et c'est tout !

Exemple : Soit le fichier xml suivant

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<compilation>
<mp3>
<titre>Foule sentimentale</titre>
<artiste>Alain Souchon</artiste>
</mp3>
<mp3>
<titre>Solaar pleure</titre>
<artiste>MC Solaar</artiste>
</mp3>
<mp3>
<titre>Le baiser</titre>
<artiste>Alain Souchon</artiste>
</mp3>
<mp3>
<titre>Pourtant</titre>
<artiste>Vanessa Paradis</artiste>
</mp3>
<mp3>
<titre>Chambre avec vue</titre>
<artiste>Henri Salvador</artiste>
</mp3>
</compilation>
```

Nous allons trier notre compilation de mp3 en XML en ordre alphabétique croissant du nom des artistes. Et pour changer un peu, on permute les colonnes "titre" et "artiste" pour bien montrer que le XSL affiche les données du fichier XML selon le fichier Html (ou autre) qu'il contient.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <body>
    <table border="1" cellspacing="0" cellpadding="3">
      <tr bgcolor="#FFFF00">
        <td>Artiste</td>
        <td>Titre</td>
      </tr>
      <xsl:for-each select="compilation/mp3" >
```

```

        <xsl:sort select="artiste" order="ascending" />
        <tr>
            <td>
                <xsl:value-of select="artiste"/>
            </td>
            <td>
                <xsl:value-of select="titre"/>
            </td>
        </tr>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

On enregistre le fichier sous le nom xsl_order avec une extension .xsl. Testez.

Exercice : reprenez le fichier utilisateurs.xml et affichez les utilisateurs par nom.