Mise en forme des documents XML avec CSS.

La mise en forme d'un document XML avec CSS ressemble beaucoup à la mise en forme d'un document HTML à la différence près que les balises du document HTML sont cette fois, les éléments du document XML.
On crée un fichier avec l'extension ".css" et on y fait référence dans le document xml avec la déclaration :

<?xml-stylesheet href="fichier.css" type="text/css"?> précisant où se trouvent les styles

Ci-dessous, un exemple très simple :

En voici le listing :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet href="css.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>
```

et la feuille css : (on remarque la propriété **display:** qui permet avec la valeur "block" de grouper les informations (cf. css cours de 1ere)

```
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;
}
```

L'inconvénient de cette mise en forme est que l'on ne sait pas accéder aux "attributs" des éléments mais uniquement à leur valeur. (Ce qui explique qu'il n'y a pas d'attribut dans l'exemple).
EXERCICE : mettre en forme (avec une feuille CSS) le fichier "Bottin.xml" (labo1). Il faut donc, pour avoir accès à toutes les données, modifier sa structure (transformer les attributs contenus dans le nœud "adresse"' en nœud "enfant" d'adresse)

Mise en forme des documents XML avec XSL (eXtensible StyleSheet Language).

Une feuille de style (ici nommée "exemple.xsl") est avant tout un document de type xml. Il se présentera comme suit avec l'espace de nom xmlns=http://www.w3.org/1999/XSL/Transform adéquat:

```
<?xml version="1.0"?>                                          ➔ c'est un fichier xml
<xsl:stylesheet  xmlns="http://www.w3.org/1999/XSL/Transform"> ➔ la référence du langage
<xsl:output method="html" encoding="ISO-8859-1"/>              ➔ precise le format de sortie

CONTENU du fichier xsl

</ xsl:stylesheet>                                             ➔ fermeture balise stylesheet
```

A noter sur la troisième ligne `<xsl:output method="html>` qui permet de préciser le type de transformation, ici "html" et le type d'encodage (on aurait pu avoir UTF-8 au lieu d'ISO (//ANSI))

Dans le document "exemple.xml" contenant les données, on fera référence à la feuille de style avec la déclaration suivante:

```
<?xml version="1.0"?>
<?xml-stylesheet  href="exemple.xsl"  type="text/xsl"?>

<racine unique du document>

CONTENU

</racine unique du document>
```

Le traitement d'un document XML peut s'effectuer de trois façons:

- Par un processeur XSL installé sur l'ordinateur client (inclus dans le navigateur Internet)
- Par un processeur XSL installé sur le serveur (si vous voulez tester avec Chrome, il est nécessaire de travailler comme cela, avec Apache (inclus dans WAMP) comme serveur Web, par exemple.
- Par un logiciel spécifique installé sur le serveur

Nous envisageons ici la première solution, ce qui demande à l'ordinateur client de posséder impérativement un navigateur compatible avec les technologies XML (Firefox, Edge).

Le processeur XSL permettra soit de transformer un fichier XML en HTML ou en un autre fichier XML (tout est possible en fait) (XSL – T pour transformation) et d'ajouter au fichier de sortie HTML, un lien classique vers un fichier CSS.

L'espace de nom impose que tous les éléments d'une feuille de style commence par **<xsl:** pour ce qui est de la transformation et par **<fo:** pour ce qui est du formatage (utilisé dans le domaine de l'édition/impression).

Une feuille de style va consister en la définition d'une structure imbriquée de gabarits qui va permettre la mise en forme des éléments du document XML apparaissant dans la définition des gabarits.

On va illustrer l'utilisation des feuilles de style sur l'exemple "Bottin" légèrement modifié. On va utiliser dans ce premier exemple la notion de gabarit, permettant de mettre en forme un nœud particulier. Notez qu'un gabarit a +/- le même comportement qu'une boucle For Each càd qu'il traite tous les nœuds.

Le fichier "ex_XML_XSL.xml" :

```
<?xml version="1.0" encoding="ISO-8859-1"  ?>
<?xml-stylesheet  href="ex_XML_XSL.xsl"  type="text/xsl" ?>
<Bottin region="Hainaut"  annee="2006" >
<!--début du contenu du bottin-->

      <Abonne id="01">
            <Nom>Cuvelier</Nom>
            <Prenom>Charles</Prenom>
            <Prefixe>064</Prefixe>
            <Tel>263542</Tel>
            <Adresse designation="rue" nom="du puits" numero="56"></Adresse>
            <Ville>Nivelles</Ville>
            <CodePostal>7160</CodePostal>
      </Abonne>

      <Abonne id="02">
            <Nom>Dupuis</Nom>
            <Prenom>Béatrice</Prenom>
            <Prefixe>065</Prefixe>
            <Tel>362456</Tel>
            <Adresse designation="avenue" nom="De Gaule" numero="248"></Adresse>
            <Ville>Mons</Ville>
            <CodePostal>7000</CodePostal>
      </Abonne>

      <Abonne id="03">
      <Nom>Bossart</Nom>
      <Prenom>Willy</Prenom>
      <Prefixe>067</Prefixe>
      <Tel>340552</Tel>
      <Adresse designation="boulevard" nom="Sanders" numero="87"></Adresse>
      <Ville>VilleDieu</Ville>
      <CodePostal>7060</CodePostal>
      </Abonne>
</Bottin>
<!--fin du contenu du bottin-->
```

Le fichier "ex_XML_XSL.xsl" :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="ISO-8859-1"/>
<!-- création d'un gabarit sur le noeud "Bottin"-->
<xsl:template match="Bottin">
 <!-- Dès que l'on trouve le noeud "Bottin", on commence le document HTML avec <Body>-->
 <html>
      <body>
     <xsl:apply-templates select="Abonne"/> <!-- imbrication des gabarits correspondant à "Abonne"-->
      </body>
 </html>
 </xsl:template>
<!-- création d'un gabarit sur le noeud Abonne-->
 <xsl:template match="Abonne">
       <!-- Dans ce qui suit   équivaut à l'espace et &#47; équivaut à / -->
     <h1 align="center">
          <xsl:value-of select="@id"/> <xsl:value-of select="Nom"/> <xsl:value-of select="Prenom"/>
                              <!-- @ permet d'accéder aux attributs du nœud courant-->
          <!-- <xsl:value-of select="Nom"/> permet d'accéder à la valeur de l'enfant noté "enfant" du noeud courant "Abonne"-->
          <!-- il s'agit d'expressions XPath qui est un langage qui permet de définir des chemins de localisation  http://fr.wikipedia.org/wiki/XPath-->
     </h1>
     <p align="center"> <font size="5" color="red">
         TELEPHONE:<xsl:value-of select="Prefixe"/>&#47;<xsl:value-of select="Tel"/>
     </font></p>
     <h3 align="center">
          <xsl:apply-templates select="Adresse"/> <!-- imbrication des gabarits correspondant à "Adresse"-->
          <xsl:value-of select="CodePostal"/> <xsl:value-of select="Ville"/>
     </h3>
     <HR/>  <!-- trace une ligne entre chaque abonne-->
 </xsl:template>

<!-- création d'un gabarit sur le noeud Adresse-->
<xsl:template match="Adresse">
      <h3 align="center">
       <xsl:value-of select="@designation"/> 
       <xsl:value-of select="@nom"/> 
       N° <xsl:value-of select="@numero"/>
      </h3>
 </xsl:template>

</xsl:stylesheet>
```

Le résultat :

01 Cuvelier Charles

TELEPHONE:064/263542

rue du puits N° 56

7160 Nivelles

---

02 Dupuis Béatrice

TELEPHONE:065/362456

avenue De Gaule N° 248

7000 Mons

---

03 Bossart Willy

TELEPHONE:067/340552

boulevard Sanders N° 87

7060 VilleDieu

Exercices:

1) Modifier le fichier XSL ci-dessus afin que l'attribut "région" et la valeur de l'année présents dans l'élément "Bottin" apparaissent dans le titre de la fenêtre du navigateur (balise <Title> en html)

2) Pour le fichier "stage.xml" généré précédemment en VBA, concevoir les mises en formes suivantes:

- Premier niveau de difficulté : mise en forme inspirée de l'exemple ci-dessus. (En "vrac" avec une mise ne forme de votre choix – ne pas trop sophistiquer, le but est de comprendre les bases du xsl)

- Deuxième niveau de difficulté : mise en forme en utilisant un tableau (HTML) le plus fidèlement inspiré de l'annexe 1.0. (cf. labo1 fichier Labo_XML01_stage_-1-.rtf – conseil : fusionner des cellule HTML pour obtenir le résultat voulu.