

Le Standard XML -XSLT -

Ichrak MEHREZ
(m_ichrak@hotmail.fr)

Introduction

- **XSLT** ou **eXtensible Stylesheet Language Transformations**
- Une technologie qui permet de *transformer les informations* d'un document XML vers un autre type de document comme un autre document XML ou encore une page web
- XSLT s'appuie sur une autre technologie (pour sélectionner des informations à transformer : **XPath**.
- Un **document XSLT** est associé à un **document XML** afin de créer un nouveau document d'une nature différente ou identique

Structure d'un document XSLT

- L'extension portée par les documents XSLT est **.xsl**
- **Prologue**: la première ligne d'un document XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
```

- **Le corps**: l'élément racine est **<xsl:stylesheet />**

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">  
</xsl:stylesheet>
```

- ❑ version: numéro de version utilisée (1 ou 2)
- ❑ xmlns:xsl nous permet de déclarer un **espace de noms**.
 - ➔ toutes les balises utilisées dans un document XSLT doivent être préfixées par **xsl:**

Balise output

- Placée directement après l'élément racine, permet de décrire le document produit à l'issue des différentes transformations.
- L'attribut *method* : préciser le *type* du document produit
 - ❑ **xml** : un document XML.
 - ❑ **html** : un document HTML.
 - ❑ **text** : un document texte.
 - ❑ **xhtml** (version 2) : un document xHTML.
- L'attribut *encoding* : préciser l'*encodage* du document produit à l'issue des transformations.
 - ❑ UTF-8.
 - ❑ ISO-8859-1.
 - ❑ etc.

Balise output

- L'attribut *indent* permet d'indiquer si l'on souhaite que le document produit à l'issue des transformations soit *indenté* ou non.
 - ❑ Yes.
 - ❑ No.
- L'attribut *doctype-public* permet de renseigner l'identifiant public de la DTD.
- L'attribut *doctype-system* permet de renseigner l'URL de la DTD.

Balise output

- Un exemple nous permettant de produire un document HTML 4

```
<!-- le prologue -->
<?xml version="1.0" encoding="UTF-8" ?>

<!-- l'élément racine -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <!-- l'élément output -->
  <xsl:output
    method="html"
    encoding="UTF-8"
    doctype-public="-//W3C//DTD HTML 4.01//EN"
    doctype-system="http://www.w3.org/TR/html4/strict.dtd"
    indent="yes" ></xsl:output>

  <!-- reste du document XSLT -->

</xsl:stylesheet>
```

Référencer un document XSLT

- Le référencement d'un document XSLT se fait au niveau du document XML dont les informations seront utilisées au cours de la transformation (sous le prologue)

```
<?xml-stylesheet type="text/xsl" href="mon_document.xsl" ?>
```

- L'attribut **type**: permet de définir le *type* du document que nous souhaitons référencer.
- L'attribut **href**: permet d'indiquer l'*URI* du document que l'on souhaite référencer.

Les templates

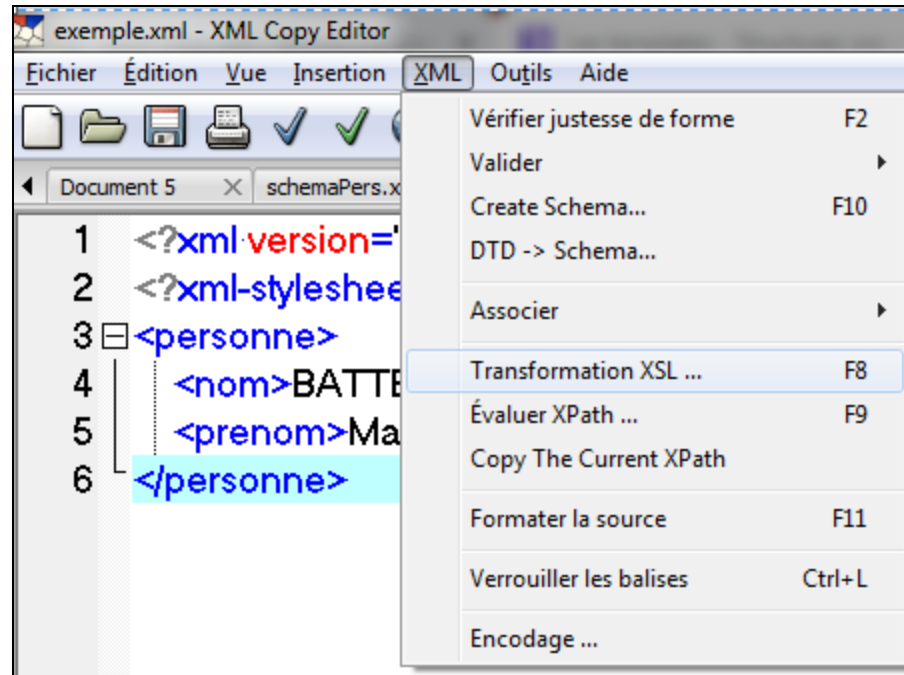
- Le **corps** d'un document XSLT est composé d'un ensemble de **templates**.
- Un **template** est défini par la balise **<xsl:template />** à laquelle plusieurs attributs peuvent être associés:
 - L'attribut **match** permet de renseigner une **expression XPath**.
 - L'attribut **name** est le nom donné au **template**, permettant de l'identifier de manière unique. Cet attribut est important puisqu'il n'ai pas rare qu'un template en appelle un autre.

Un premier exemple

```
<?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"
    doctype-public="-//W3C//DTD HTML 4.01//EN"
    doctype-system="http://www.w3.org/TR/html4/strict.dtd"
    indent="yes"></xsl:output>

  <xsl:template match="/">
    <html>
      <head>
        <title>Mon premier document XSLT</title>
      </head>
      <body>
        <p>Bonjour !</p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Un premier exemple



Un premier exemple

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Mon premier document XSLT</title>
  </head>
  <body>
    <p>Bonjour!</p>
  </body>
</html>
```

Les fonctions: value-of

- La fonction `<xsl:value-of />` ("valeur de") pour extraire la *valeur d'un élément XML* ou la *valeur de ses attributs*.
- Cette fonction possède un attribut *select* auquel il convient de renseigner une expression XPath

```
<xsl:value-of select="expression XPath" />
```

Les fonctions: value-of

- Afficher le type du numéro de téléphone et le numéro en lui même de Marie POPPINS

```
<xsl:template match="/">
  <html>
    <head>
      <title>Test de la fonction value-of</title>
    </head>
    <body>
      <p>Type du numéro : <xsl:value-of select="repertoire/personne[nom='POPPINS']/telephones/telephone/@type">
      </xsl:value-of>
      </p>
      <p>Numéro : <xsl:value-of select="repertoire/personne[nom='POPPINS']/telephones/telephone">
      </xsl:value-of>
      </p>
    </body>
  </html>
</xsl:template>
```

Les fonctions: for-each

- La fonction `<xsl:for-each />`, ("pour chaque") est une fonction qui permet de boucler sur un ensemble d'éléments.
- La fonction `<xsl:for-each/>` possède un attribut `select` auquel il convient de renseigner une expression XPath

```
<xsl:for-each select="expression XPath" />
```

Les fonctions: for-each

- Afficher pour chaque adresse e-mail de John DOE son type et son contenu

```
<xsl:template match="/">
<html>
  <head>
    <title>Test de la fonction for-each</title>
  </head>
  <body>
    <xsl:for-each select="repertoire/personne[nom='DOE']/emails/email">
      <p>Type de l'adresse e-mail : <xsl:value-of select="@type">
      </xsl:value-of>
      </p>
      <p>adresse e-mail : <xsl:value-of select=".">
      </xsl:value-of>
      </p>
    </xsl:for-each>
  </body>
</html>
</xsl:template>
```