

# FOP

## Formatting Objects Processor

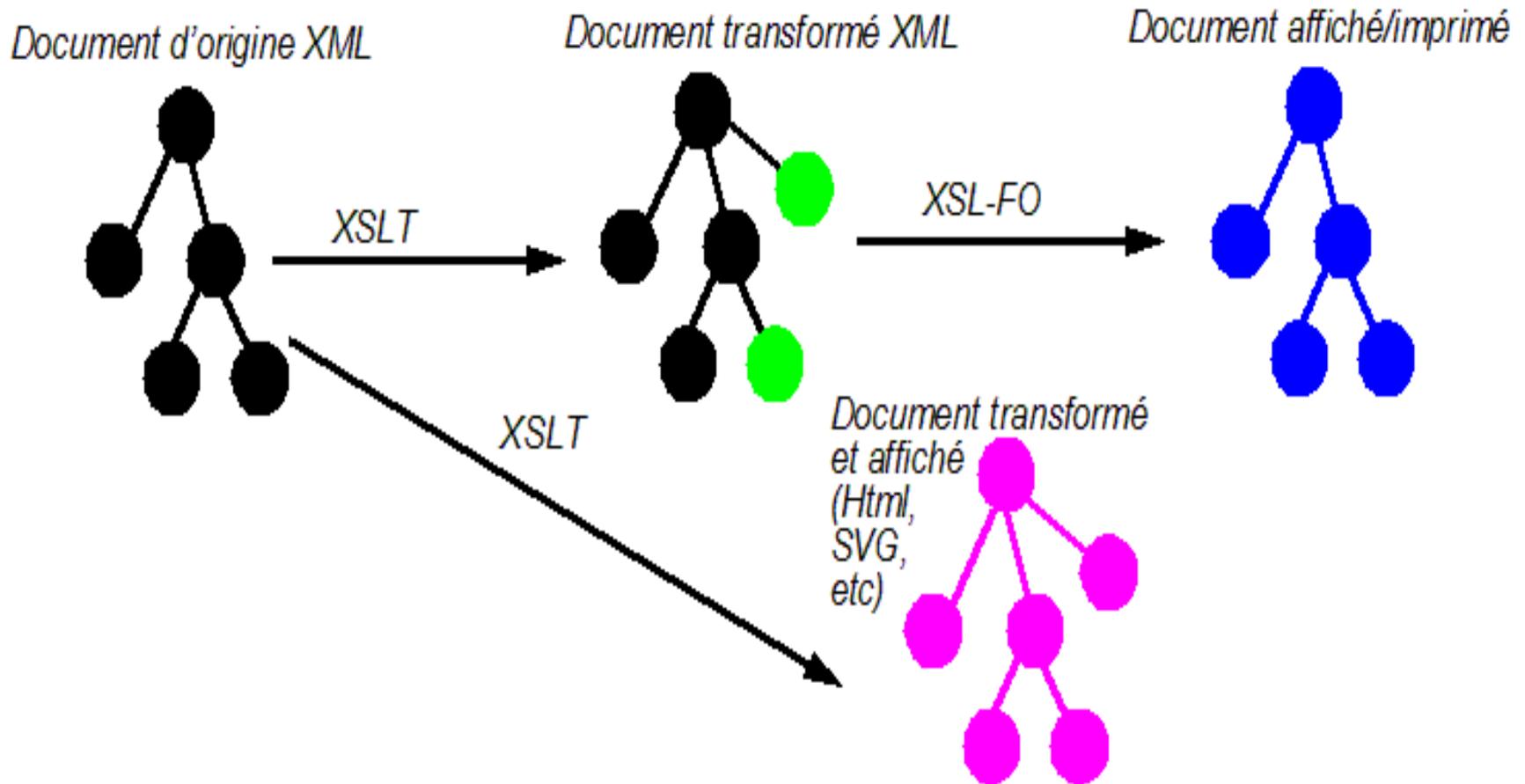
Génération de documents  
imprimables

# XSL :

XSLT : langage de transformation

Xpath : langage de navigation

XSL-FO : Vocabulaire de mise en forme de documents (audio, écran, papier)



- FOP est un projet apache
- Il permet de générer des documents imprimables (pdf, rft, ps, xml, ...)
- Le site de référence est [xml.apache.org/fop/](http://xml.apache.org/fop/)
- Les fichiers pdf de ce site sont créés en utilisant Apache FOP

Le principe de fop est le suivant :

1. un fichier xml est transformé en un fichier xml-fo par une transformation xslt

2. le fichier xml-fo est transformé en un document imprimable

- Le fichier xml-fo est un fichier XML défini avec des balises fo:???. La racine est fo:root.

- On peut générer le fichier formaté de deux façons

1) `Fop -xml fic.xml -xsl fic.xsl -xxx fic.xxx`

2) `Fop -fo fic.fo -xxx fic.xxx`

où xxx peut valoir pdf, rtf, svg, ps, ... .

- Les moteurs fop travaillent encore avec des moteurs xslt 1.0.
- Astuce pour utiliser un moteur xslt 2.0  
vous faites l'opération en deux temps
  - `java -jar saxon9he.jar foo.xml foo.xsl > foo.fo`
  - `fop -fo foo.fo -pdf foo.pdf (par exemple)`
- Pour rendre transparent, vous pouvez écrire un script (nommé fop2 par exemple) qui reçoit quatre paramètres : le nom du fichier xml, le nom du fichier xsl, l'extension pour préciser le type du fichier de sortie et le nom du fichier de sortie.
- Sous Unix, les lignes principales du script fop2 pourraient être :

```
java -jar saxon9he.jar {$1} {$2} > {$1}.fo
fop -fo {$1}.fo -{$3} {$4}
rm {$1}.fo
```

- Les nœuds (ou leurs ancêtres) du fichier xsl doivent préciser les espaces de noms utiles.

`xmlns:xsl="http://www.w3.org/1999/XSL/Transform"`  
car on utilise le langage xsl.

`xmlns:fo="http://www.w3.org/1999/XSL/Format"` car  
on utilise le langage fo.

`xmlns:svg="http://www.w3.org/2000/svg"` car on peut  
utiliser le langage svg pour les éléments graphiques  
de l'impression.

`xmlns:fox="http://xml.apache.org/fop/extensions"` pour  
certaines extensions de fo.

- Le formatage des pages

```
<fo:layout-master-set>  
  <fo:simple-page-master  
    master-name="simpleA4"  
    page-height="29.7cm"  
    page-width="21cm"  
    margin-top="1cm"  
    margin-bottom="2cm"  
    margin-left="2cm"  
    margin-right="2cm">  
    <fo:region-body margin-top="1cm"  
      margin-bottom="1.5cm"/>  
    <fo:region-before extent="1cm"/>  
    <fo:region-after extent="1.cm"/>  
  </fo:simple-page-master>  
</fo:layout-master-set>
```

- La structure pour la définition des différentes pages

```
<fo:page-sequence master-reference="simpleA4">  
  <fo:static-content flow-name="xsl-region-  
    before">  
    ...  
  </fo:static-content>  
  <fo:static-content flow-name="xsl-region-after">  
    ...  
  </fo:static-content>  
  <fo:flow flow-name="xsl-region-body">  
    ...  
  </fo:flow>  
</fo:page-sequence>
```

- Le contenu pour la définition des différentes pages.

- Les balises `fo:block` et `fo:inline`

- `<!-- écriture d'un bloc : passage à la ligne -->`

- `<fo:block font-family="Times Roman" id="cartouche" text-align="center" line-height="24pt" background-color="{ $couleur}" color="white" space-after.optimum="6pt" font-size="18pt">`

- `...`  
`</fo:block>`

- `<!-- écriture d'un morceau de ligne -->`

- `<fo:inline>`

- `...`  
`</fo:inline>`

- On peut définir un ensemble de blocs et/ou de morceaux de lignes en définissant un noeud `fo:wrapper`. Les sous noeuds héritent des attributs graphique du noeud wrapper.

- Les navigations basiques externe et interne.

<!-- lien vers l'extérieur : xxx est une url -->

```
<fo:basic-link color="{ $couleur}" external-destination=«  
xxx">
```

```
    <fo:inline>
```

```
    ...
```

```
    </fo:inline>
```

```
</fo:basic-link>
```

<!-- lien vers un élément du document : xxx est l'id de l'élément -->

```
<fo:basic-link color="blue" internal-destination="xxx">
```

```
    ...
```

```
</fo:basic-link>
```

- Les navigations avec une table d'index.
  - On crée un arbre de bookmarks avec la balise `<fo:bookmark-tree>`.
  - Dans ce nœud, on crée autant de nœuds `<fo:bookmark internal-destination="???">` qu'il y a de bookmarks.
  - Dans chaque nœud `<fo:bookmark>`, on insère un nœud `<fo:bookmark-title>` pour le libellé.
  - Chaque nœud `<fo:bookmark>` peut lui-même contenir des nœuds `<fo:bookmark>`.

# Gérer des tableaux

- Créer un nœud `fo:table`
- Dans ce nœud, créer autant de nœuds `fo:table-column` qu'il y a de colonnes pour préciser leur largeur.
- Dans ce même nœud, créer un nœud `fo:table-body` qui contient autant de nœuds `fo:table-row` qu'il y a de lignes.
- Chaque nœud `fo:table-row` contient autant de nœuds `fo:table-cell` (non vide) qu'il y a de colonnes dans chaque ligne. On remplit ce nœud avec des nœuds `fo:block`.