

# FOP

## Formatting Objects Processor

# Génération de documents imprimables

- Les bookmarks
- Les listes
- Les images
- Les graphiques (SVG)

- 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 listes d'items

- Créer un nœud **fo:list-block**
- Dans ce nœud, créer autant de nœuds **fo:list-item** qu'il y a d'items.
- Un nœud **fo:list-item** contient :
  - un nœud **fo:list-item-label** pour préciser le repérage de l'item. C'est un **fo:block** avec le caractère dingbat.
  - un nœud **fo:list-item-body** pour préciser le contenu de l'item. C'est un ensemble de **fo:block**.

```

<xsl:template match="session-materials">
  Matériaux de cours:
  <fo:list-block space-before.optimum="6pt">
<xsl:apply-templates select="session-material"/>
</fo:list-block>
</xsl:template>
<xsl:template match="session-material">
  <fo:list-item space-before.optimum="8pt">
    <fo:list-item-label end-indent="label-end()">
      <fo:block>&#x2022;</fo:block>
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      <fo:block text-align="justify">
        <xsl:value-of select="@material-title"/>
      </fo:block>
      <fo:block text-align="justify" font-size="10pt">
        (URL:<xsl:value-of select="@material-url"/>)
        <xsl:value-of select="@material-comment"/>
      </fo:block>
    </fo:list-item-body>
  </fo:list-item>
</xsl:template>

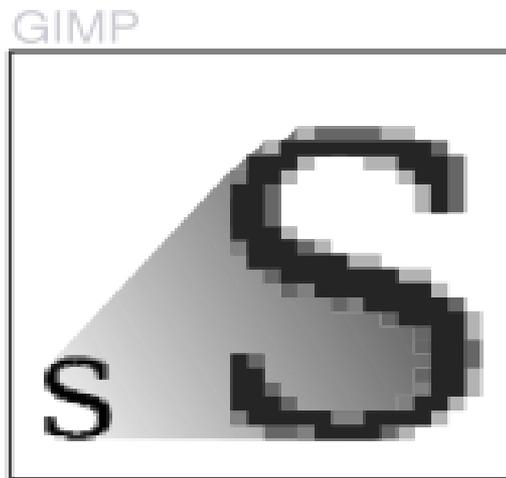
```

# Gérer des images

- Pour insérer une image dans le document, il faut insérer dans un nœud `fo:block`, un nœud `fo:external-graphic` qui a un attribut `src` pour donner l'url de l'image.
- Un exemple  
`<fo:external-graphic src="eisti.gif"/>`

## Le format **SVG**

- Scalable Vector Graphics : format de données conçu pour décrire des ensembles de graphiques vectoriel.
- Ce format est défini en **XML**.
- Il est spécifié par le World Wide Web Consortium.



**BITMAP**  
.jpeg .gif .png



**OUTLINE**  
.svg

- Pour insérer un objet SVG dans un document imprimable, on utilise un objet de flux de type `fo:instream-foreign-object`.
- Puis, on définit un arbre `svg:svg` en précisant la taille du graphique. Un arbre `svg:svg` est aussi utilisable en html.

```
<fo:instream-foreign-object>  
  <svg:svg width="???" height="???">  
  ...  
  </svg:svg>  
</fo:instream-foreign-object>
```

- Pour afficher du texte, on définit un nœud `svg:text`
- Les attributs sont
  - font-size pour la police
  - fill pour la couleur de la police
  - x et y en pixels pour positionner le début du texte

```
<svg:text font-size="15pt" x="10" y="10"  
fill="black">
```

...

```
</svg:text>
```

- Pour insérer un lien, on définit un nœud `svg:a`
- Les attributs sont
  - `xlink:href` pour l'url du lien
  - `alt` pour afficher un message quand la souris passe sur le lien

```
<svg:a xlink:href="monLien.fr"  
  alt="message">
```

...

```
</svg:a>
```

- Pour tracer un segment, on définit un nœud `svg:line`
- Les attributs sont
  - x1 et y1 pour l'origine du segment
  - x2 et y2 pour l'extrémité du segment
  - style pour le style de la ligne (couleur, épaisseur, type du trait)

```
<svg:line x1="0" y1="0" x2="10" y2="30"  
style="stroke:blue;stroke-width:1"/>
```

- Pour tracer un cercle, on définit un nœud `svg:circle`
- Les attributs sont
  - `cx` et `cy` pour les coordonnées du centre
  - `r` pour le rayon
  - `stroke` pour le style du trait
  - `stroke-width` pour la largeur du trait
  - `fill` pour la couleur de remplissage du disque

```
<svg:circle cx="10" cy="20" r="15px"  
stroke="black" stroke-width="1"  
fill="white"/>
```

- Pour tracer une polyligne, on définit un nœud `svg:path`
- Les attributs sont
  - `d` pour définir les points de la polyligne
  - `fill` pour le remplissage de la figure inscrite
  - `stroke` pour la couleur du trait
  - `stroke-width` pour la largeur du trait
  - `marker-start` pour définir un marqueur de début de ligne.
  - `marker-end` pour définir un marqueur de fin de ligne.
- L'attribut `d` est de la forme  
`d="M x1 y1 L x2 y2 L ... z"`
- Exemple  
`<svg:path d="M 0 5 L 15 20 L 30 10 L 0 5 z" fill="white" stroke="blue" stroke-width="1" marker-end="url(#Triangle)"/>`

# Des marqueurs réutilisables

- On peut définir à l'intérieur d'un nœud `svg:defs` un ensemble de marqueurs utilisables dans toute la source xsl.
- Un marqueur est associé à une figure
- Un marqueur est défini par un nœud de type `svg:marker`.
- Des attributs sont :
  - `id` pour le référencer
  - `viewBox` pour définir sa partie circonscrite
  - `refX` et `refY` pour définir l'emplacement du marqueur dans la figure marquée.
- Son contenu permet de définir la forme du marqueur.

# Un exemple de marqueur réutilisable

```
<svg:defs>  
  <!-- un marqueur en forme de triangle -->  
  <svg:marker id="Triangle"  
    viewBox="0 0 20 20" refX="0" refY="10"  
    markerUnits="strokeWidth"  
    markerWidth="8" markerHeight="6"  
    orient="auto">  
    <svg:path d="M 0 0 L 20 10 L 0 20 z" />  
  </svg:marker>  
</svg:defs>
```

- Pour insérer une image, on définit un nœud `svg:image`
- Les attributs sont
  - `x` et `y` pour définir l'emplacement de l'image dans la graphique
  - `width` et `height` pour les dimensions de l'image
  - `xlink:href` pour l'url de l'image
  - `marker-end` pour définir un marqueur de fin de ligne.
- On peut insérer un nœud `svg:title` pour avoir un titre.
- Exemple

```
<svg:image x='50' y='150' width='600'  
height='195' xlink:href='Images/fleche_bas.jpg'>  
  <svg:title>Fleche Bas</svg:title>  
</svg:image>
```