

# XSL-FO

## Extensible Stylesheet Language - Formatting Objects

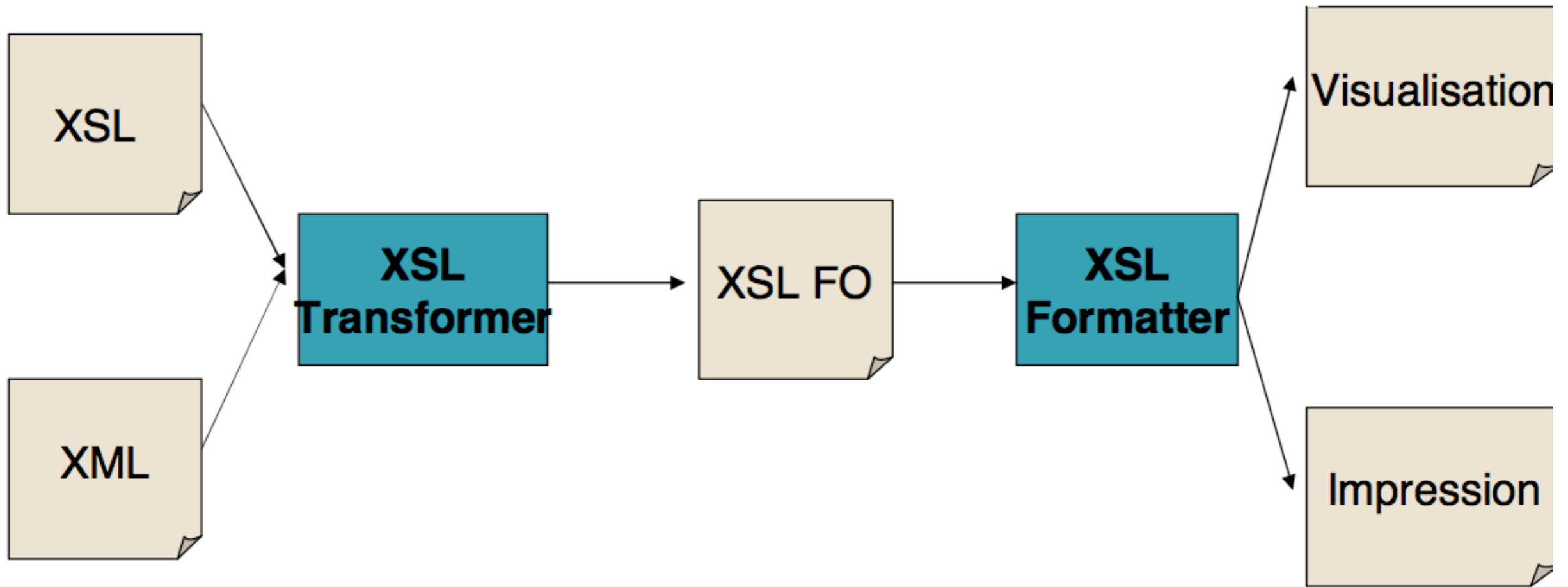
Pierre-Alain Reynier

<http://www.lif.univ-mrs.fr/~preynier/XML/>

# XSL-FO

- XSL-FO fait partie du standard XSL :  
<http://www.w3.org/TR/xsl/#fo-section>
- Offre un format de présentation permettant de générer des documents au format complexe tels que PDF, RTF, PS
- XSL-FO emploie un document résultant d'une transformation XSLT

# XSL-FO : vue globale



# XSL-FO

- Un document XSL-FO est un doc. XML
- L'espace de noms associé est :  
<http://www.w3.org/1999/XSL/Format>
- Le préfixe “fo” est utilisé en général pour le désigner
- Un document XSL-FO est constitué :
  - d'instructions de mise en page
  - d'instructions d'affichage de contenu

# Structure d'un document

**Root** (fo:root)

**Mise en page** (fo:layout-master-set)

**Contenu du document**

(fo:page-sequence)

# Structure d'un document

```
<fo:root xmlns="http://www.w3.org/1999/XSL/Format">  
  <fo:layout-master-set>  
    <fo:simple-page-master master-name="A4">  
      <!--Mise en page -->  
    </fo:simple-page-master>  
  </fo:layout-master-set>  
  
  <fo:page-sequence master-reference="A4">  
    <!--Contenu de la page -->  
  </fo:page-sequence>  
</fo:root>
```

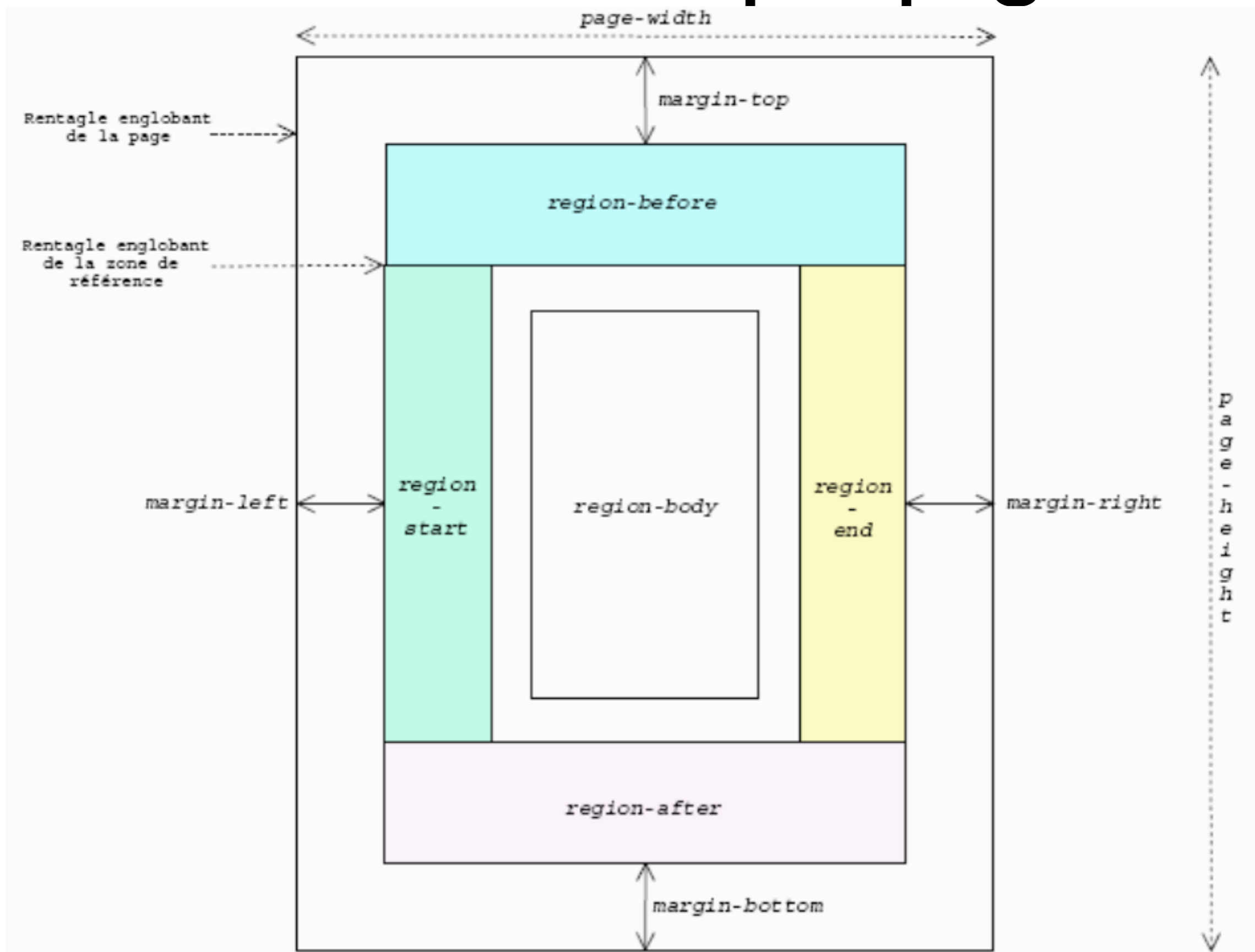
# Mise en page

# Mise en page

- Première partie d'un doc XSL-FO
- **fo:layout-master-set** : définit les mises en page utilisées dans le document.
- Une mise en page est définie par une des deux instructions :
  - **simple-page-master** : décrit la mise en page d'une page
  - **page-sequence-master** : décrit la mise en page de plusieurs pages



# Modèle de “simple page”



# Modèle de simple page

```
<fo:simple-page-master master-name="A4"  
  page-width="21cm"  
  page-height="29.7cm"  
  margin-bottom="2cm"  
  margin-top="2cm"  
  margin-left="2.5cm"  
  margin-right="2.5cm">  
  
</fo:simple-page-master>
```

# Découpage en régions

- Une page est découpée en régions :  
region-body, region-before, region-after,  
region-start, region-end.
- La taille d'une région est définie par  
l'attribut extent.
- Seule la taille de la région centrale ne peut  
pas être précisée.
- Les régions situées autour ont une taille  
mais pas de marge.

# Modèle de simple page

```
<fo:simple-page-master master-name="A4"  
  page-width="21cm"  
  page-height="29.7cm"  
  margin-bottom="2cm"  
  margin-left="2.5cm"  
  margin-right="2.5cm">  
  
  <fo:region-body margin-top="3cm"/>  
  <fo:region-start extent="2cm" name="entete"/>  
  <fo:region-before extent="2cm"/>  
  <fo:region-after extent="2cm"/>  
  <fo:region-end extent="2cm"/>  
  
</fo:simple-page-master>
```

# Découpage en régions

- Nous utiliserons ensuite les noms des régions pour préciser où placer le texte.
- En principe chaque région doit être nommée. Toutefois, il existe des noms par défaut :
  - xsl-region-body
  - xsl-region-before
  - xsl-region-after
  - xsl-region-start

# Pagination complexe

- **fo:page-sequence-master** permet de définir une pagination pour un ensemble de pages
- Cette instruction peut contenir :
  - des définitions de pages simples avec : **single-page-master-reference**
  - des définitions de présentations portant chacune sur un ensemble de pages avec :
    - **repeatable-page-master-reference**
    - **repeatable-page-master-alternatives**

# Pagination complexe

- `fo:single-page-master-reference` permet de formater une seule page
- Cette instruction utilise une référence vers la valeur de l'attribut `master-name` d'un élément `fo:simple-page-master`

# Pagination complexe

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="page"
      page-height="297mm" page-width="210mm"
      margin-top="10mm" margin-bottom="10mm"
      margin-left="20mm" margin-right="20mm">
      <fo:region-before extent="30mm" />
      <fo:region-body margin-top="20mm"/>
      <fo:region-after extent="15mm"/>
    </fo:simple-page-master>
    <fo:page-sequence-master master-name="simple">
      <fo:single-page-master-reference master-reference="page"/>
    </fo:page-sequence-master>
  </fo:layout-master-set>
  .....
</fo:root>
```



# Pagination complexe

- `fo:repeatable-page-master-reference` définit une mise en page pour un ensemble de pages simples
- L'attribut `maximum-repeats` définit le nombre de pages constituant cet ensemble
- La valeur `no-limit` indique un nombre de pages potentiellement infini.

# Pagination complexe

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">  
<fo:layout-master-set>  
  <fo:simple-page-master master-name="page"  
  ....>  
</fo:simple-page-master>  
  
  <fo:page-sequence-master master-name="plusieurs">  
    <fo:repeatable-page-master-reference  
      master-reference="page"  
      maximum-repeats="no-limit"/>  
  </fo:page-sequence-master>  
</fo:layout-master-set>  
  
....  
</fo:root>
```

# Pagination complexe

- **fo:repeatable-page-master-alternatives**  
définit différentes mises en pages pour un ensemble de pages
- Le modèle à appliquer dépend de certaines conditions relatives à la page produite :
  - position de la page dans la séquence de pages,
  - numéro de la page,
  - page blanche ou non.

# Pagination complexe

- Les différents modèles de pages de `fo:repeatable-page-master-alternatives` sont introduits par :  
`fo:conditional-page-master-reference`
- Cet élément possède les attributs :
  - `page-position`
  - `odd-or-even`
  - `blank-or-not-blank`
  - `master-reference`

# Pagination complexe

- page-position peut prendre 4 valeurs :
  - first : première page
  - last : dernière page
  - rest : toute autre page
  - any : n'importe quelle page
- odd-or-even peut prendre 3 valeurs :
  - odd : page impaire
  - even : page paire
  - any : n'importe quelle page

# Pagination complexe

- blank-or-not-blank peut prendre 3 valeurs :
  - blank : la page ne contient aucune donnée
  - not-blank : la page contient au moins une donnée
  - any : n'importe quelle page
- master-reference fait référence à la valeur de l'attribut master-name d'un fo:simple-page-master

# Pagination complexe

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
<fo:layout-master-set>
  <fo:simple-page-master master-name="normale" ... >
  ... </fo:simple-page-master>
  <fo:simple-page-master master-name="impaire" ...>
  ... </fo:simple-page-master>
  <fo:simple-page-master master-name="paire" ...>
  ... </fo:simple-page-master>
  <fo:page-sequence-master master-name="pair-impair">
  <fo:repeatable-page-master-alternatives maximum-repeats="no-limit">
    <fo:conditional-page-master-reference master-reference="normale"
    page-position="first"/>
    <fo:conditional-page-master-reference master-reference="impaire"
    odd-or-even="odd"/>
    <fo:conditional-page-master-reference master-reference="paire"
    odd-or-even="even"/>
    <fo:conditional-page-master-reference master-reference="normale"
    page-position="last"/>
  </fo:repeatable-page-master-alternatives>
  </fo:page-sequence-master>
</fo:layout-master-set>
....
</fo:root>
```

# Contenu du document



# Contenu du document

- Seconde partie d'un doc XSL-FO
- Défini à l'aide de l'instruction **fo:page-sequence** (pour un ens. de pages)
- **fo:page-sequence** contient les éléments **fo:title**, **fo:block** et **fo:static-content** :
  - un seul **fo:flow** est autorisé dans un élément **fo:page-sequence**
  - plusieurs **fo:static-content** sont autorisés

# Contenu du document : **static**

**fo:static-content** : contenu défini pour toutes les pages

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

```
<fo:layout-master-set>
```

```
<fo:simple-page-master master-name="page"  
page-height="297mm" page-width="210mm"  
margin-top="10mm" margin-bottom="10mm"  
margin-left="20mm" margin-right="20mm">
```

```
<fo:region-before extent="30mm" />
```

```
<fo:region-body margin-top="20mm"/>
```

```
<fo:region-after extent="15mm"/>
```

```
</fo:simple-page-master>
```

```
<fo:layout-master-set>
```

```
<fo:page-sequence master-reference="page">
```

```
<fo:static-content flow-name="xsl-region-before">
```

```
<fo:block> En tête </fo:block>
```

```
</fo:static-content>
```

```
</fo:page-sequence>
```

```
</fo:root>
```

# Contenu du document : flow

**fo:flow** : définit le contenu (glissant) des pages

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

```
<fo:layout-master-set>
```

```
<fo:simple-page-master master-name="page"  
page-height="297mm" page-width="210mm"  
margin-top="10mm" margin-bottom="10mm"  
margin-left="20mm" margin-right="20mm">
```

```
<fo:region-before extent="30mm" />
```

```
<fo:region-body margin-top="20mm"/>
```

```
<fo:region-after extent="15mm"/>
```

```
</fo:simple-page-master>
```

```
<fo:layout-master-set>
```

```
<fo:page-sequence master-reference="page">
```

```
<fo:flow flow-name="xsl-region-body">
```

```
<fo:block> Contenu </fo:block>
```

```
</fo:flow>
```

```
</fo:page-sequence>
```

```
</fo:root>
```

# Contenu du document : **block**

- **fo:block** permet de formater des blocs de texte et d'images.
- Un bloc peut représenter un ensemble de lignes ou une partie de ligne
- Un bloc possède des attributs permettant de spécifier sa fonte, sa couleur, la taille de la fonte...
- Un bloc vide permet de simuler un saut de ligne

# Contenu du document : **block**

Un exemple simple :

```
<fo:block  
  color="#FFFFFF"  
  background-color="green"  
  border-color="red"  
  border-style="solid">  
  Contenu  
</fo:block>
```

# Contenu du document : **block**

**fo:inline** permet de définir les mises en forme sur une partie de block :

```
<fo:block>  
  <fo:inline  
    color="red">C</fo:inline>ontenu  
</fo:block>
```

# Contenu du document : listes

Les listes à puces sont représentées avec :  
`fo:list-block`

```
<fo:list-block>  
  <fo:list-item>  
    <fo:list-item-label>...</fo:list-item-label >  
    <fo:list-item-body>...</fo:list-item-body>  
  </fo:list-item> ...  
</fo:list-block>
```

# Contenu du document : **tableaux**

- **fo:table-and-caption** permet de définir un tableau et sa légende.
- **fo:table-caption** définit la légende
- **fo:table** définit le tableau
- Les dimensions de chaque colonne doivent d'abord être précisées avec **fo:table-column**
- Un tableau comprend trois parties **table-header**, **table-body** et **table-footer**



# Contenu du document : **tableaux**

- Chaque partie est découpée en lignes avec **fo:table-row**
- Chaque ligne est découpée en cellules avec **fo:table-cell**
- L'attribut **table-layout** (de **fo:table**) définit comment faire le rendu du tableau :
  - **fixed** : la taille des colonnes est définie à partir de tailles précisées
  - **auto** : la taille des colonnes est définie à partir des données présentes dans les colonnes

# Contenu du document : tableaux

```
<table>
  <table-column column-width="( 210mm -2 * 15mm ) -2in"/>
  <table-column column-width="1 in"/>
  <table-column column-width="1 in"/>
  <table-body>
    <table-row>
      <table-cell>
        <block font-size="19pt"> Exemple </block>
        <block font-size="10pt" > Module 1 </block>
      </table-cell>
      <table-cell>
        <block text-align="start"> Exemple 2 </block>
      </table-cell>
    </table-row>
  </table-body>
</table>
```

# Contenu du document : tableaux

```
<fo:table-cell padding="6pt" border="1pt solid blue"
  background-color="silver" number-columns-
  spanned="3"[4]>
  <fo:block text-align="center" font-weight="bold">
    Header
  </fo:block>
</fo:table-cell>
```

# Contenu du document : **images**

Il est possible d'insérer des images à l'aide de **fo:external-graphic**

```
<fo:external-graphic  
  src="/Users/reynier/images/logo.jpg"  
  width="99px" height="109px"/>
```

# Contenu du document : **images**

On peut également inclure du code SVG :

**fo:instream-foreign-object**

```
<fo:block>
  <fo:instream-foreign-object>
    <svg:svg width="400pt" height="200pt">
      <svg:rect style="fill:none;stroke:blue"
        x="20pt" y="20pt"
        width="100pt" height="100pt"/>
      <svg:line x1="20pt" y1="20pt" x2="120pt" y2="120pt"/>
      <svg:line x1="120pt" y1="20pt" x2="20pt" y2="120pt"/>
      <svg:text x="20pt" y="150pt">Hello SVG!</svg:text>
    </svg:svg>
  </fo:instream-foreign-object>
</fo:block>
```

# Contenu du document : liens

- Des liens entre documents peuvent être insérés grâce à `fo:basic-link`
- Les liens peuvent être internes ou externes suivant les attributs `external-destination` ou `internal-destination` :
  - lien interne : vers une partie du même document. L'ancre de destination est positionnée par l'attribut "id" d'un élément `fo:block`
  - lien externe : vers un autre document
- Les propriétés graphiques de ces liens peuvent être spécifiées dans la balise `fo:basic-link`

# Contenu du document : liens

```
<fo:flow flow-name="xsl-region-body">
  <fo:block>
    <fo:basic-link
      external-destination="http://www.lif.univ-mrs.fr"
      color="blue"> http://www.lif.univ-mrs.fr
    </fo:basic-link>
    <fo:basic-link
      internal-destination="Ancre1"> Vers l'ancre 1
    </fo:basic-link>
  </fo:block>
<fo:block id="Ancre1"> L'ancre est ici </fo:block>
</fo:flow>
```

# Une dernière remarque

- On peut obtenir le numéro de la page :  
<fo:page-number/>
- On peut faire des notes de bas de page :

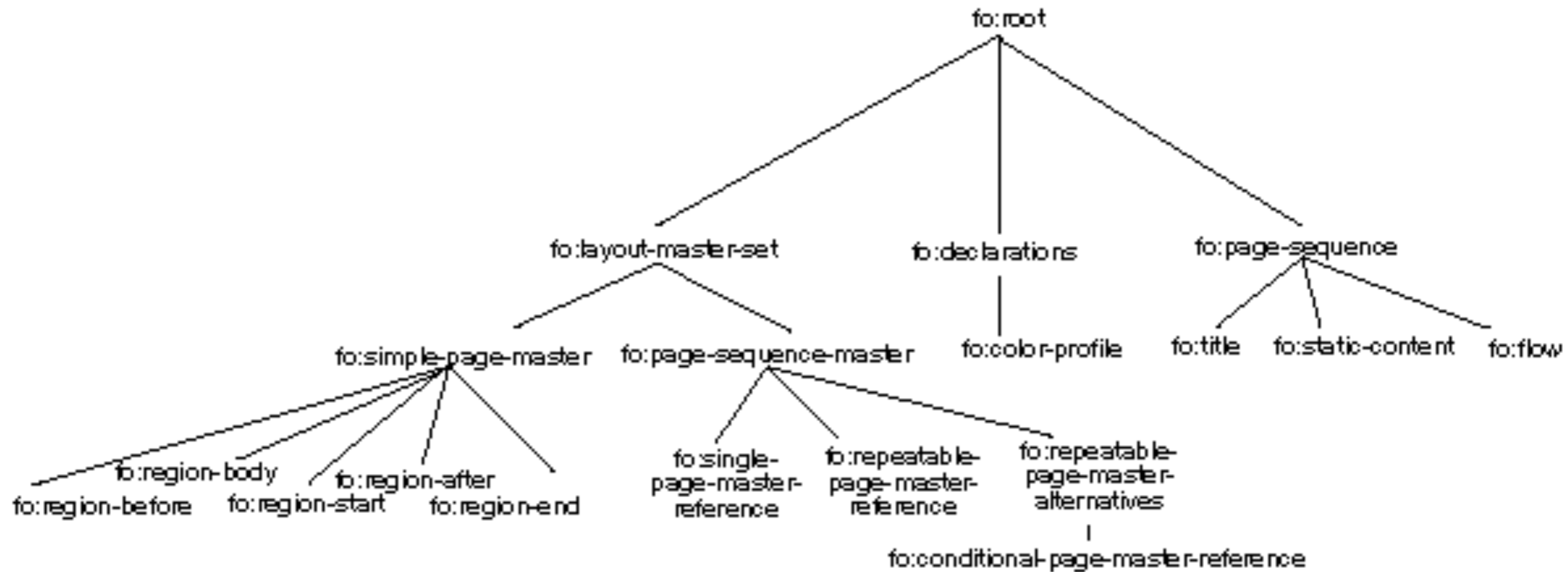
```
<fo:footnote>  
  <fo:inline>(1)</fo:inline>  
  <fo:footnote-body>  
    (1) En effet...  
  </fo:footnote-body>  
</fo:footnote>
```

et bien d'autres choses encore....



# Conclusion et Mise en œuvre

# Arborescence des éltis XSL-FO



# Pour aller plus loin

- La page W3C contenant la recommandation :

<http://www.w3.org/TR/xsl/#fo-section>

- Une référence très complète :

<http://www.ibiblio.org/xml/books/bible2/chapters/ch18.html>

# Chaîne complète

- Pour obtenir un document PDF à partir d'un document XML, il faut :
  - 1 - Définir une transformation XSLT transformant le document XML en un document XSL-FO
  - 2 - Convertir le document XSL-FO au format choisi (ici PDF) en utilisant un formateur XSL-FO

# Chaîne complète : XSLT (I)

```
<xsl:stylesheet version="1.0"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:xsd="http://www.w3.org/1999/XSL/Transform" >
<xsl:output method="xml"/>
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
      <fo:simple-page-master master-name="page"
        page-height="297mm" page-width="210mm"
        margin-top="10mm" margin-bottom="10mm"
        margin-left="20mm" margin-right="20mm">
        <fo:region-before extent="30mm" />
        <fo:region-body margin-top="20mm"/>
        <fo:region-after extent="15mm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
```

# Chaîne complète : XSLT (2)

```
<fo:page-sequence master-reference="page">  
  <fo:flow flow-name="xsl:region-body">  
    <xsl:apply-templatesselect="//personne" />  
  </fo:flow>  
</fo:page-sequence>  
</fo:root>  
</xsl:template>
```

```
<xsl:template match="personne" >  
  <fo:block>  
    <xsl:value-of select="concat(nom, ' ',prenom)" />  
  </fo:block>  
</xsl:template>
```

# Chaîne complète : XSL-FO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="page"
      page-height="297mm" page-width="210mm"
      margin-top="10mm" margin-bottom="10mm"
      margin-left="20mm" margin-right="20mm">
      <fo:region-before extent="30mm" />
      <fo:region-body margin-top="20mm"/>
      <fo:region-after extent="15mm"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="page">
    <fo:flow flow-name="xsl:region-body">
      <fo:block> Zola Emile</fo:block>
      <fo:block> Dickens Charles </fo:block> ...
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

# Quelques processeurs FO

- Apache Group : FOP  
Formatting Object Processor  
Génère du PDF : <http://www.apache.org>
- JFOR (Open Source)  
Génère du RTF <http://www.jfor.org>
- Antenna House  
XSL Formatter <http://www.antennahouse.com>
- RenderX  
Génère du PDF : <http://www.renderx.com>

et d'autres....