

# Modélisation et interopérabilité :

## Semaine 47, cours 9

Benoît Valiron <[benoit.valiron@monoidal.net](mailto:benoit.valiron@monoidal.net)>

<http://inf356.monoidal.net/>

Quelques transparents de la semaine dernière pour fixer les choses...

# XPath

- Chemin absolu, chemin relatif vis à vis d'un noeud contextuel
  - `eltenant`
  - `@att`
  - `text()`
  - `elt1|elt2`
  - `//`
  - `..`
  - `.`
  - `*`
- Notion d'axes : `child`, descendant, ancestor, siblings
- Types d'expressions : booléen, nombre, chaîne de caractères, ensemble de noeuds

# Exemple

```
<film>
  <date>2009</date>
  <titre>The lost Valley</titre>
  <casting>
    <acteur>Alice</acteur>
    <acteur>Bob</acteur>
    <acteur lang="fr">
      Camille
    </acteur>
  </casting>
</film>
```

Titre du film correspondant au  
noeud contextuel :

```
../../titre
```

```
../preceding-sibling::titre
```

```
ancestor::film/titre
```

Acteur parlant français

```
../*[@lang="fr"]
```

# XSLT : Transformations XML

# XSLT

- Description d'une transformation d'un document XML.
- Basé sur des structures appelées **modèles (template)**.
- Document source –(feuille de style)→ document résultat.
- Espace de noms :  
<http://www.w3.org/1999/XSL/Transform>
- Extension : `.xsl`

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

  <output method="xml"/>      <----- IMPORTANT POUR
                                PRODUIRE DU XML

  <template match='...'>
    ...
  </template>

</stylesheet>
```

- Un modèle décrit une règle, c'est-à-dire une action à effectuer pour chaque noeud décrit par l'expression XPath contenue dans l'attribut `match`
- Il y a un "template" par défaut qui affiche tout le texte d'un noeud donné.
- L'attribut `priority` prend une valeur décimale.
- Par défaut, le modèle avec le `match` le plus précis a la priorité maximale.

# Dans un template, on trouve...

- `<value-of select="exp" />`
- `<for-each select="exp">`  
    `...`  
    `</for-each>`
- `<if test="exp">`  
    `...`  
    `</if>`
- `<choose>`  
    `<when test="exp1">...</when>`  
    `<when test="exp1">...</when>`  
    `<when test="exp1">...</when>`  
    `...`  
    `<otherwise>...</otherwise>`  
    `</choose>`
- `<text>...</text>`
- `<variable name="nom"`  
    `select="exp" />`
- `<variable name="nom">`  
    `...`  
    `</variable>`
- `<copy-of select="exp" />`
- `<element name="nom">`  
    `...`  
    `</element>`
- `<attribute name="nom">`  
    `...`  
    `</attribute>`



```
<variable name='nom' select='exp' />
```

- Associe à **nom** le résultat de l'expression XPath **exp**.
- Réutilisable avec **\$name** dans **<value-of ... />**

```
<liste>
  <quoi>Benoît</quoi>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre>
    <titre>Q. C.</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

```
<stylesheet version='1.0' xmlns='... '>
  <template match="/liste">
    <variable name='nom' select='quoi' />
    <for-each select='livre'>
      <value-of select='$nom' /> :
      <value-of select='.' />
    </for-each>
  </template>
</stylesheet>
```

**Réponse :**

Livre : XML

Livre : Quantum Computation

# Exemple

```
<lib>
  <livre id="a">
    <titre>XML</titre>
  </livre>
  <livre id="b">
    <titre>Quantum Computation</titre>
  </livre>
  <emprunt ref="a">Valiron</emprunt>
  <emprunt ref="b">Turing</emprunt>
</lib>
```

Réponse :

XML

**doc.xsl :**

```
<stylesheet version="1.0"
  xmlns="http://www.w3.org/1999/XSL/Transform">
  <template match="/">
    <variable name='x' select='/lib/emprunt[.="Valiron"]/@ref' />
    <value-of select='/lib/livre[@id=$x]/titre' />
  </template>
</stylesheet>
```

# Exemple

```
<stylesheet version='1.0'
```

```
  xmlns='http://www.w3.org/1999/XSL/Transform'>
```

```
<template match="/">
```

```
  <for-each select='//liste/gare'>
```

```
    <variable name='x' select='@id' />
```

```
    <value-of select='.' /> : <for-each select='//trajet'>
```

```
      <value-of select='horaire[@gare=$x]/@t' />
```

```
      <text> </text>
```

```
    </for-each>
```

```
    <text>
```

```
</text>
```

```
  </for-each>
```

```
</template>
```

```
</stylesheet>
```

**doc.xml :**

```
<trains>
```

```
  <liste>
```

```
    <gare id='g'>Grenoble</gare>
```

```
    <gare id='l'>Lyon</gare>
```

```
    <gare id='c'>Chambéry</gare>
```

```
  </liste>
```

```
  <trajet>
```

```
    <horaire gare='g' t='10:01' />
```

```
    <horaire gare='l' t='10:22' />
```

```
    <horaire gare='c' t='11:07' />
```

```
  </trajet>
```

```
  <trajet>
```

```
    <horaire gare='g' t='15:10' />
```

```
    <horaire gare='l' t='16:22' />
```

```
    <horaire gare='c' t='16:49' />
```

```
  </trajet>
```

```
</trains>
```

**Réponse :**

Grenoble : 10:01 15:10

Lyon : 10:22 16:22

Chambéry : 11:07 16:49

# <apply-templates select='...' />

- Part du noeud contextuel et applique le modèle le plus adapté à l'ensemble de noeuds décrit par **select**.

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre>
    <titre>Q. C.</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

```
<stylesheet version="1.0" xmlns="...">
  <template match="*">
    Un noeud <apply-templates select='*' />
  </template>
</stylesheet>
```

→ Écrit 10 fois "Un noeud"

# <apply-templates select='...' />

- Part du noeud contextuel et applique le modèle le plus adapté à l'ensemble de noeuds décrit par **select**.

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre>
    <titre>Q. C.</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

```
<stylesheet version="1.0" xmlns="...">
  <template match="*">
    Noeud : <apply-templates
              select='*|text()' />
  </template>
  <template match="text()">
    <value-of select='.' />
  </template>
</stylesheet>
```

Écrit :

```
Noeud : liste
Noeud : livre
Noeud : titre
Noeud : auteur
Noeud : pub
Noeud : livre
...
```

# <apply-templates />

- Dans ce cas, `select` est `*|text()`

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre>
    <titre>Q. C.</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

```
<stylesheet version="1.0" xmlns="...">
  <template match="*">
    Un noeud : <apply-templates />
  </template>
  <template match="text()">
    <value-of select=".'" />
  </template>
</stylesheet>
```

# Modèle par défaut

```
<stylesheet version="1.0" xmlns="...">
```

```
</stylesheet>
```

**=**

```
<stylesheet version="1.0" xmlns="...">
```

```
  <template match="/|*">
```

```
    <apply-templates />
```

```
  </template>
```

```
  <template match="text()|@"*>
```

```
    <value-of select=".'" />
```

```
  </template>
```

```
</stylesheet>
```

# Par défaut :

- On commence à la racine.
  - Si on rencontre un élément :
    - On applique le modèle par défaut à ses fils
  - Si on rencontre du texte :
    - On l'écrit dans le document de sortie
- Donc par défaut, on passe au travers de tout le document et on écrit le texte qu'on rencontre.



# Exemple

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

Réponse :

```
XML
Chagnon
2007
```

```
Quantum Computation
Nielsen
Chuang
2001
```

**doc.xsl :**

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

</xsl:stylesheet>
```

# Exemple

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

**doc.xsl :**

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

  <xsl:template match="/*">
    <xsl:apply-templates />
  </xsl:template>

  <xsl:template match="text()|@*">
    <xsl:value-of select=".'" />
  </xsl:template>

</xsl:stylesheet>
```

**Réponse :**

XML  
Chagnon  
2007

Quantum Computation  
Nielsen  
Chuang  
2001

# Exemple 2

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

On  
réécrit...

**doc.xsl :**

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

  <xsl:template match="/liste/livre[@genre='web']">
    Trouvé
  </xsl:template>

</xsl:stylesheet>
```

# Exemple 2

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

Réponse :

Trouvé

Quantum Computation  
Nielsen  
Chuang  
2001

**doc.xsl :**

```
<xsl:stylesheet version="1.0" xmlns:xsl="...">
  <xsl:template match="/liste/livre[@genre='web']">
    Trouvé
  </xsl:template>
  <xsl:template match="/*"><xsl:apply-templates /></xsl:template>
  <xsl:template match="text()|@*">
    <xsl:value-of select=".'" />
  </xsl:template>
</xsl:stylesheet>
```

# Exemple 2

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

Réponse :

Trouvé

**doc.xsl :**

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

  <xsl:template match="/liste/livre[@genre='web']">
    Trouvé
  </xsl:template>

  <template match="text()|@*"></template>

</xsl:stylesheet>
```

# Réécriture de <for-each ...>

```
<stylesheet version='1.0'
  xmlns='http://www.w3.org/1999/XSL/Transform'>
  <template match="/">
    <for-each select='/liste/livre'>
      <value-of select='titre' /> : <for-each select='auteur'>
        <value-of select='.' />
        <text> </text>
      </for-each>
    <text>
  </text>
  </for-each>
</template>
</stylesheet>
```

**doc.xml :**

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

**Réponse :**

XML : Chagnon

Quantum Computation : Nielsen Chuang

# Réécriture de <for-each ...>

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

  <template match="livre">
    <value-of select='titre' /> : <apply-templates select='auteur' />
    <text>
</text>
  </template>

  <template match="auteur">
    <value-of select="." /><text> </text>
  </template>

</stylesheet>
```

**doc.xml :**

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```

# Réécriture de <for-each ...>

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

  <template match="livre">
    <value-of select='titre' /> : <apply-templates/>
    <text>
</text>
</template>

  <template match="auteur">
    <value-of select="." /><text> </text>
</template>

  <template match="text()"></template>
</stylesheet>
```

**doc.xml :**

```
<liste>
  <livre genre="web">
    <titre>XML</titre>
    <auteur>Chagnon</auteur>
    <pub>2007</pub>
  </livre>
  <livre genre="quantique">
    <titre>Quantum Computation</titre>
    <auteur>Nielsen</auteur>
    <auteur>Chuang</auteur>
    <pub>2001</pub>
  </livre>
</liste>
```



# mode="nom"

- Attribut pris par <template> et <apply-templates>
- Utile quand deux mêmes éléments doivent avoir un formatage différent.
- Il y a un mode générique, quand l'attribut n'est pas utilisé.
- Le processeur XSLT commence sans mode

# Exemple

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

```
  <template match="/">  
    aaa  
  </template>
```

```
</stylesheet>
```

**doc.xml**

```
<livre>  
  <auteur>Nielsen</auteur>  
  <titre>Q. C.</titre>  
</livre>
```

**Réponse :**

aaa

# Exemple

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

```
  <template match="/" mode="abc">  
    aaa  
  </template>
```

```
</stylesheet>
```

**doc.xml**

```
<livre>  
  <auteur>Nielsen</auteur>  
  <titre>Q. C.</titre>  
</livre>
```

**Réponse :**

Nielsen  
Q. C.

# Exemple

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

  <template match="/">
    <apply-templates select='/' mode='a' />
  </template>

  <template match="/" mode='a'>
    <apply-templates select='/' mode='b' />
    <apply-templates select='/' mode='c' />
  </template>

  <template match="/" mode='b'>
    Ici <apply-templates select='/' mode='c' />
  </template>

  <template match="/" mode='c'>
    Là
  </template>

</stylesheet>
```

## doc.xml

```
<livre>
  <auteur>Nielsen</auteur>
  <titre>Q. C.</titre>
</livre>
```

Réponse :

Ici  
Là  
  
Là

# Exemple

```
<x:stylesheet version='1.0'
  xmlns:x='http://www.w3.org/1999/XSL/Transform'
  xmlns='http://www.w3.org/1999/xhtml'>
  <x:output method='xml' />

  <x:template match="/article">
    <html>
      <x:apply-templates select='title' mode='head' />
      <body>
        <x:apply-templates select='section' mode='body' />
      </body>
    </html>
  </x:template>
  <x:template match='title' mode='head'>
    <head><title><x:value-of select='.' /></title></head>
  </x:template>
  <x:template match='title' mode='body'>
    <h1><x:value-of select='.' /></h1>
  </x:template>
  <x:template match='para' mode='body'>
    <p><x:value-of select='.' /></p>
  </x:template>

</x:stylesheet>
```

## doc.xml

```
<article>
<title>Mon beau titre</title>
<section>
<title>Section 1</title>
<para>
Blah Blah
</para>
</section>
</article>
```

# Exemple

```
<x:stylesheet version='1.0'
  xmlns:x='http://www.w3.org/1999/XSL/Transform'
  xmlns='http://www.w3.org/1999/xhtml'>
  <x:output method='xml' />

  <x:template match="/article">
    <html>
      <x:apply-templates select='title' mode='head' />
      <body>
        <x:apply-templates select='section' mode='body' />
      </body>
    </html>
  </x:template>
  <x:template match='title' mode='head'>
    <head><title><x:value-of select='.' /></title></head>
  </x:template>
  <x:template match='title' mode='body'>
    <h1><x:value-of select='.' /></h1>
  </x:template>
  <x:template match='para' mode='body'>
    <p><x:value-of select='.' /></p>
  </x:template>

</x:stylesheet>
```

doc.xml

```
<article>
<title>Mon beau titre</title>
<section>
<title>Section 1</title>
<para>
Blah Blah
</para>
</section>
</article>
```

## Résultat

```
<html xmlns="...xhtml">
<head>
<title>Mon beau titre</title>
</head>
<body>
<h1>Section 1</h1>
<p>
Blah Blah
</p>
</body></html>
```

# Arguments pour les modèles

- `<template match='...'>`  
  `<param name='nom'>valeur défaut</param>`  
  ...  
  `</template>`
- `<apply-templates select='...'>`  
  `<with-param name='nom'>`  
  une valeur  
  `</with-param>`  
  `</apply-template>`
- Valeur de nom accessible avec `$nom`

# Exemple

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

<template match="/">
  <for-each select='//liste/gare'>
    <variable name='x' select='@id' />
    <value-of select='.' /> : <for-each select='//trajet'>
      <value-of select='horaire[@gare=$x]/@t' />
      <text> </text>
    </for-each>
    <text>
  </text>
  </for-each>
</template>
</stylesheet>
```

**doc.xml :**

```
<trains>
  <liste>
    <gare id='g'>Grenoble</gare>
    <gare id='l'>Lyon</gare>
    <gare id='c'>Chambéry</gare>
  </liste>
  <trajet>
    <horaire gare='g' t='10:01' />
    <horaire gare='l' t='10:22' />
    <horaire gare='c' t='11:07' />
  </trajet>
  <trajet>
    <horaire gare='g' t='15:10' />
    <horaire gare='l' t='16:22' />
    <horaire gare='c' t='16:49' />
  </trajet>
</trains>
```

**Réponse :**

```
Grenoble : 10:01 15:10
Lyon : 10:22 16:22
Chambéry : 11:07 16:49
```



# Exemple

```
<stylesheet version='1.0'
  xmlns='http://www.w3.org/1999/XSL/Transform'>
  <template match="liste/gare">
    <value-of select='.' /> : <apply-templates select='//trajet' mode='h'>
      <with-param name='x'><value-of select='@id' /></with-param>
    </apply-templates>
  </template>

  <template match='trajet' mode='h'>
    <param name="x">g</param>
    <value-of select='horaire[@gare=$x]/@t' />
    <text> </text>
  </template>
</stylesheet>
```

**doc.xml :**

```
<trains>
  <liste>
    <gare id='g'>Grenoble</gare>
    <gare id='l'>Lyon</gare>
    <gare id='c'>Chambéry</gare>
  </liste>
  <trajet>
    <horaire gare='g' t='10:01' />
    <horaire gare='l' t='10:22' />
    <horaire gare='c' t='11:07' />
  </trajet>
  <trajet>
    <horaire gare='g' t='15:10' />
    <horaire gare='l' t='16:22' />
    <horaire gare='c' t='16:49' />
  </trajet>
</trains>
```

**Réponse :**

Grenoble : 10:01 15:10  
Lyon : 10:22 16:22  
Chambéry : 11:07 16:49

# Exemple

```
<stylesheet version='1.0'
  xmlns='http://www.w3.org/1999/XSL/Transform'>
  <template match="liste/gare">
    <value-of select='.'/> : <apply-templates select='//trajet' mode='h'/>
  </template>

  <template match='trajet' mode='h'>
    <param name="x">g</param>
    <value-of select='horaire[@gare=$x]/@t'/>
    <text> </text>
  </template>
</stylesheet>
```

**doc.xml :**

```
<trains>
  <liste>
    <gare id='g'>Grenoble</gare>
    <gare id='l'>Lyon</gare>
    <gare id='c'>Chambéry</gare>
  </liste>
  <trajet>
    <horaire gare='g' t='10:01'/>
    <horaire gare='l' t='10:22'/>
    <horaire gare='c' t='11:07'/>
  </trajet>
  <trajet>
    <horaire gare='g' t='15:10'/>
    <horaire gare='l' t='16:22'/>
    <horaire gare='c' t='16:49'/>
  </trajet>
</trains>
```

**Réponse :**

```
Grenoble : 10:01 15:10
Lyon : 10:01 15:10
Chambéry : 10:01 15:10
```

# Exemple

```
<stylesheet version='1.0'
  xmlns='http://www.w3.org/1999/XSL/Transform'>
  <template match="liste/gare">
    <value-of select='.' /> : <apply-templates select='//trajet'>
      <with-param name='x'><value-of select='@id' /></with-param>
    </apply-templates>
  </template>

  <template match='trajet'>
    <param name="x">g</param>
    <value-of select='horaire[@gare=$x]/@t' />
    <text> </text>
  </template>
</stylesheet>
```

**doc.xml :**

```
<trains>
  <liste>
    <gare id='g'>Grenoble</gare>
    <gare id='l'>Lyon</gare>
    <gare id='c'>Chambéry</gare>
  </liste>
  <trajet>
    <horaire gare='g' t='10:01' />
    <horaire gare='l' t='10:22' />
    <horaire gare='c' t='11:07' />
  </trajet>
  <trajet>
    <horaire gare='g' t='15:10' />
    <horaire gare='l' t='16:22' />
    <horaire gare='c' t='16:49' />
  </trajet>
</trains>
```

**Réponse :**

Grenoble : 10:01 15:10  
Lyon : 10:22 16:22  
Chambéry : 11:07 16:49

10:01  
15:10