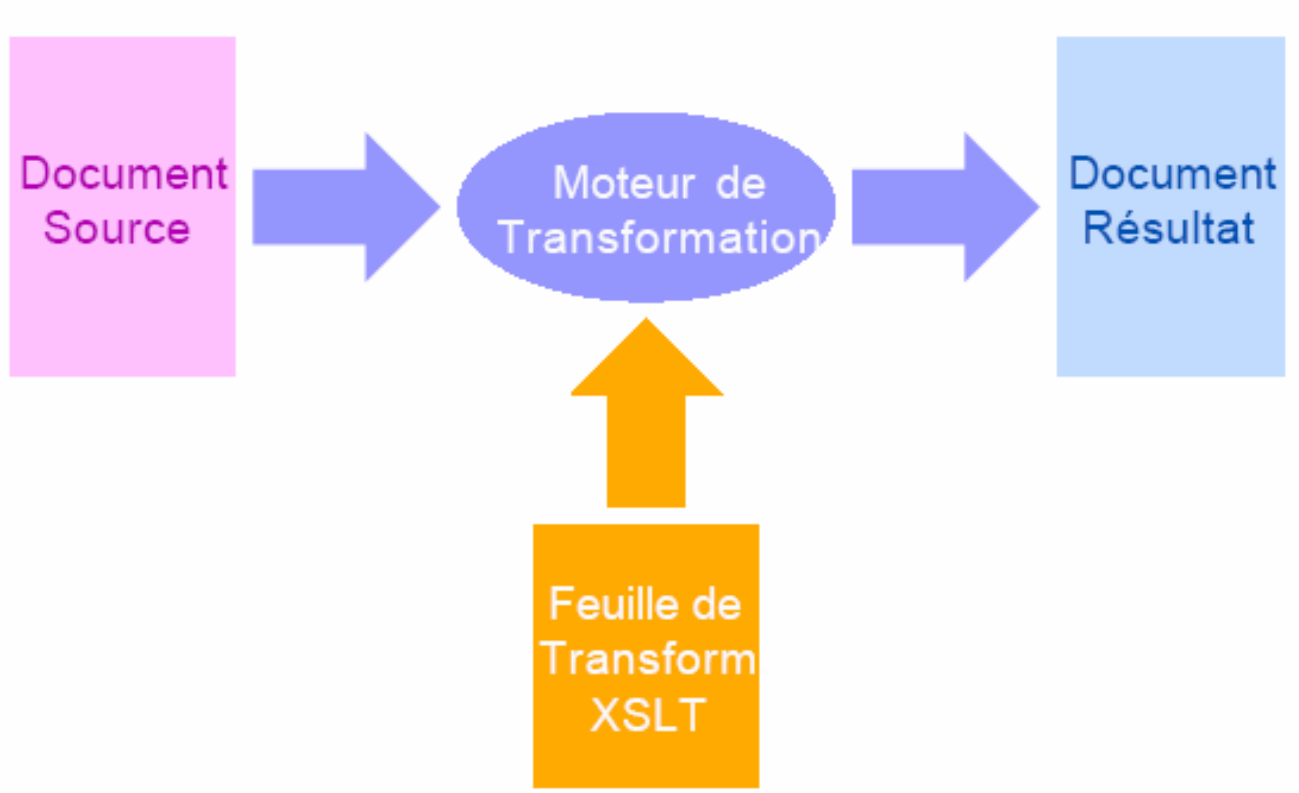
XSL
Langage de transformation de XML

Concepts fondateurs

http://www.mulberrytech.com/quickref/xpath2.pdf

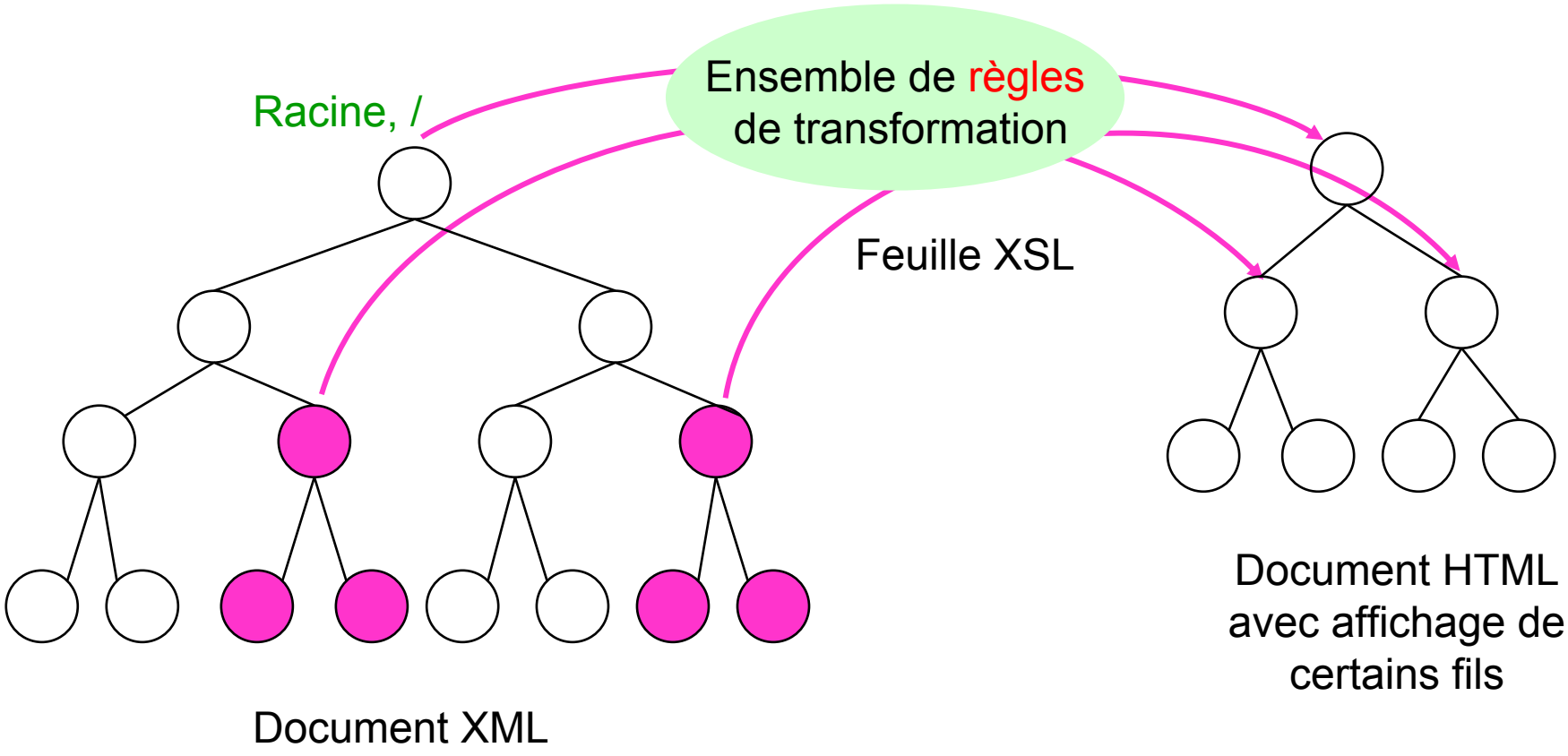
L’idée


Document Source
Moteur de Transformation
Document Résultat
Feuille de Transform XSLT

XSL

La feuille de style ou de transformation

- Ensemble de règles



Le langage XSLT

- Format d'une règle de transformation

```
<xsl:template match='un motif'>
  [action]
</xsl:template>
```

- Un **motif** (**XPath**) qui identifie le/les nœud(s) XML du document qui est/sont concerné(s) par la règle et sur le(s) quel(s) il faut appliquer une action
- Une **action** qui effectue la transformation et/ou spécifie les caractéristiques de la présentation
  - Nous allons voir les actions de type :
    - Sélection de nœuds (Xpath)
    - Réécriture de ces noeuds

Le langage XSL

Voici un premier document XSL

<?xml version="1.0"?>	Normal car XSL est dérivé de XML
<xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">	Le doc est du XSL *extensible stylesheet*. xmlns fait réf au "namespace" utilisé
<xsl:template match="/">	Détermine un gabarit dans lequel on va transformer des éléments du fichier XML. L'attribut match="/" signale que sont concernées toutes les balises XML du document associé à partir de la racine
<html> <body>	Début de la partie Html qui servira de support pour l'affichage du document
Diverses balises Html et XSL... Par exemple : <xsl:value-of select="chemin d'accès/élément" />	<xsl:value-of> sera très utilisée : permet de sélectionner un élément du fichier XML associé pour le traiter dans le fichier XSL. Dans l'attribut select, on détermine le chemin d'accès vers la balise XML souhaitée (puisque le XML est structuré) comme le chemin d'accès de répertoire en sous-répertoire vers un dossier
</body> </html>	
</xsl:template>	
</xsl:stylesheet>	

Le langage XSLT

Voici un premier document XSLT

Attention !

- Pour que ce fichier XSL soit d'une quelconque utilité, il faut encore faire référence, dans le fichier XML, au fichier XSL (feuille de style)
- On ajoutera donc dans le fichier XML :

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

Cette balise indique au navigateur qu'une feuille de style [stylesheet] est associée au fichier XML et qu'il doit aller chercher le fichier de style à l'adresse indiquée par l'attribut href

XSL

Premier exemple : exemple1.xml

- Le fichier XML

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="example1.xsl"?>
<source>
    <title>XSL</title>
    <author>John Smith</author>
</source>
```

XSL

Premier exemple : exemple1.xml

- Le fichier XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
    <h1>
        <xsl:value-of select="//title"/>
    </h1>
    <h2>
        <xsl:value-of select="//author"/>
    </h2>
</xsl:template>
</xsl:stylesheet>
```

Le langage XSLT

Exemple de document : xsldemo.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="xsldemo.xsl"?>
<compilation>
    <mp3>
        <titre>Foule sentimentale</titre><artiste>Alain Souchon</artiste>
    </mp3>
    <mp3>
        <titre>Solaar pleure</titre> <artiste>MC Solaar</artiste>
    </mp3>
    <mp3>
        <titre>Le baiser</titre> <artiste>Alain Souchon</artiste>
    </mp3>
    <mp3>
        <titre>Pourtant</titre><artiste>Vanessa Paradis</artiste> </mp3>
    <mp3>
        <titre>Chambre avec vue</titre> <artiste>Henri Salvador</artiste>
    </mp3>
</compilation>
```

Action : value-of, Selection : select

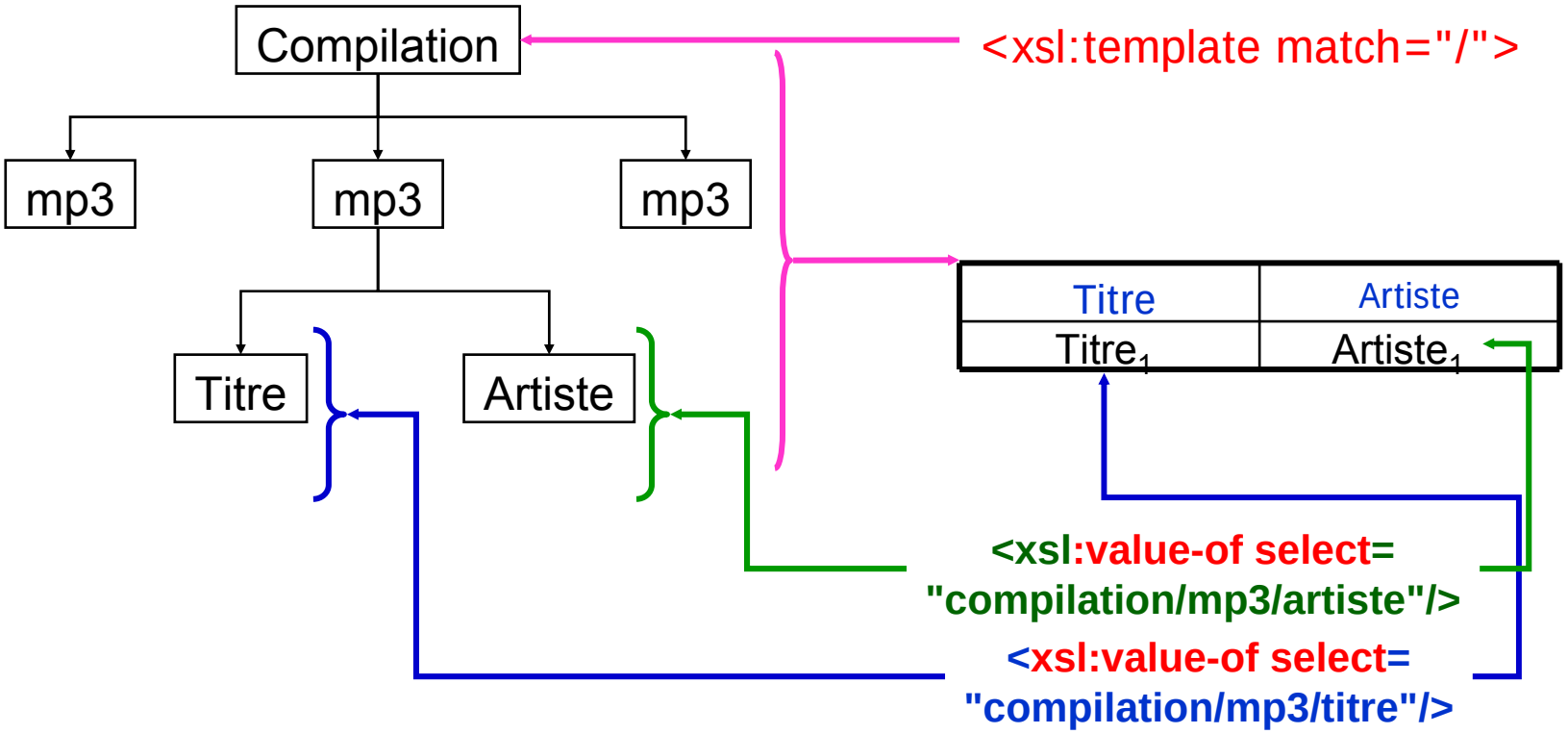
Exemple : xsldemo.xsl

```
<?xml version='1.0'?>
    <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
    <xsl:template match="/">
      <html>
      <body>
      <table border="1" cellspacing="0" cellpadding="3">
            <tr bgcolor="#FFFF00">
                  <td>Titre</td>
                  <td>Artiste</td>
            </tr>
            <tr>
                  <td><xsl:value-of select="compilation/mp3/titre"/></td>
                  <td><xsl:value-of select="compilation/mp3/artiste"/></td>
            </tr>
      </table>
    </body>
    </html>
    </xsl:template>
    </xsl:stylesheet>
```

Action : value-of, Sélection : select

Exemple : xsldemo.xsl

- Fonctionnement : sélectionne la première occurrence



Action : value-of, Sélection : select

Exemple : xsldemo.xsl

- Le résultat

Réaliser la transformation par l'interface XSLT

Sélection de plusieurs éléments

Exemple : xsldemo.xsl

- Afficher toutes les données : xsl:for-each
  - xsl:for-each [pour chaque] avec comme attribut select="compilation/mp3"
  - Exemple :

Sélection de plusieurs éléments
xsl:for-each

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
    <html>
       <body>
       <table border="1" cellspacing="0" cellpadding="3">
              <tr bgcolor="#FFFF00">
                     <td>Titre</td>
                     <td>Artiste</td>
              </tr>
              <xsl:for-each select="compilation/mp3">
                     <tr>
                            <td><xsl:value-of select="titre"/></td>
                            <td><xsl:value-of select="artiste"/></td>
                     </tr>
              </xsl:for-each>
    </table>
    </body>
    </html>
</xsl:template>
</xsl:stylesheet>
```

Sélection de plusieurs éléments

Exemple : xsldemo2.xsl

- Résultat

Titre	Artiste
Foule sentimentale	Alain Souchon
Solaar pleure	MC Solaar
Le baiser	Alain Souchon
Pourtant	Vanessa Paradis
Chambre avec vue	Henri Salvador

Afficher le .xml par l'interface XSLT

Sélection et tri

- order-by
  - permet de trier les éléments sélectionnés en ordre croissant ou décroissant
    - order-by="+balise"
      - pour trier en ordre croissant
    - order-by="-balise"
      - pour trier en ordre décroissant

Sélection et tri

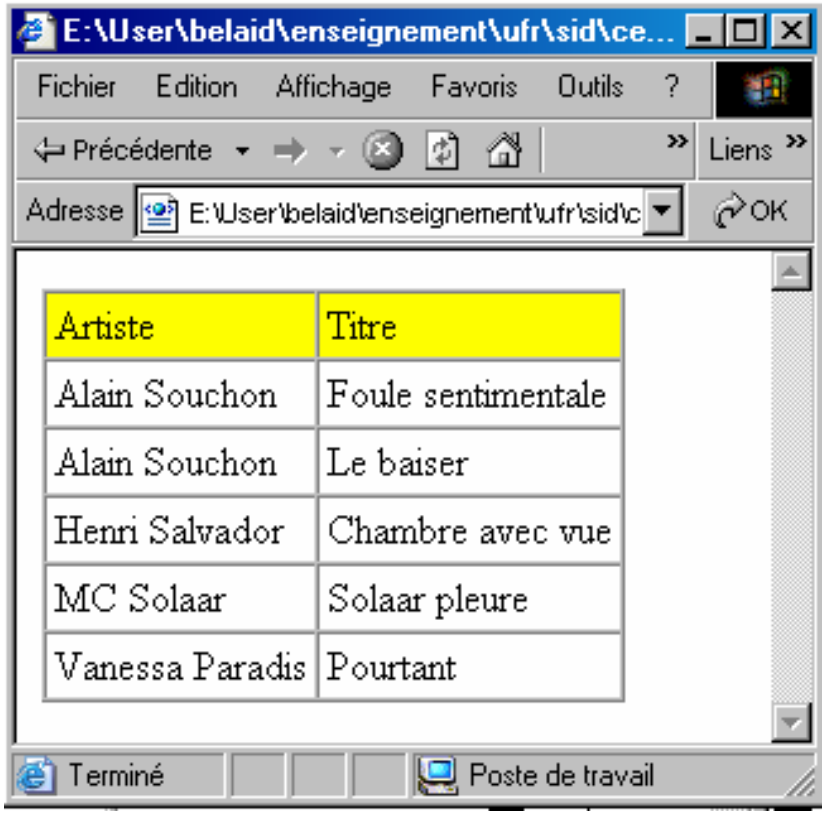
Exemple : xsldemo3.xsl

```
…
<xsl:template match="/">
    <html>
    <body>
        <table border="1" cellspacing="0" cellpadding="3">
            <tr bgcolor="#FFFF00">
                <td>Artiste</td>
                <td>Titre</td>
            </tr>
            <xsl:for-each select="compilation/mp3" order-by="+artiste">
                <tr>
                    <td><xsl:value-of select="artiste"/></td>
                    <td><xsl:value-of select="titre"/></td>
                </tr>
            </xsl:for-each>
    </table>
    …
```

Sélection et tri

Exemple : xsldemo3.xsl

- Résultat



Sélection avec filtrage

Prédicat :

- XSLT permet de filtrer les données du fichier XML associé selon des critères comme égal, pas égal, plus grand que, plus petit que
- On utilise pour cela le prédicat : balise=valeur dans la sélection

  select="chemin_d'accès[balise='xxx']"
- Les opérateurs possibles sont :

  = pour égal

  != pour différent (non égal)

  > pour plus grand que

  < pour plus petit que

Exemple

- Sélectionner toutes les chansons d'un artiste : Alain Souchon

➔ select="compilation/mp3[artiste='Alain Souchon']"

Sélection avec filtrage

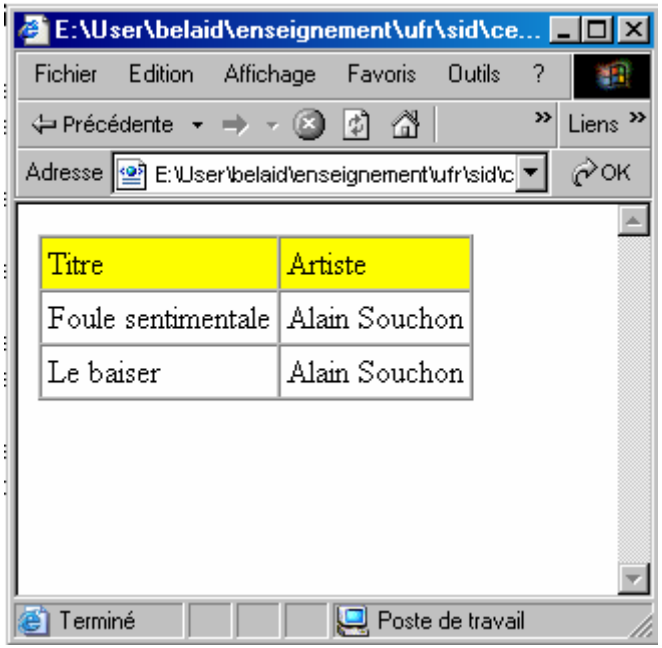
Exemple : xsldemo4.xsl

```
<html>
    <body>
    <table border="1" cellspacing="0" cellpadding="3">
        <tr bgcolor="#FFFF00">
            <td>Titre</td>
            <td>Artiste</td>
        </tr>
        <xsl:for-each select="compilation/mp3[artiste='Alain Souchon']">
        <tr>
            <td><xsl:value-of select="titre"/></td>
            <td><xsl:value-of select="artiste"/></td>
        </tr>
    </xsl:for-each>
    </table>
    </body>
</html>
```

Sélection avec filtrage

Exemple : xsldemo4.xsl

- Résultat



Choix sur la sélection

- xsl:if
  - permet d'appliquer un test conditionnel sur les éléments sélectionnés

    <xsl:if test="condition-booléenne"> Instructions... </xsl:if>
  - Si le test conditionnel est vérifié alors le processeur XSL exécutera les instructions contenues à l'intérieur des marqueurs, sinon il les ignorera et passera aux instructions suivantes
- Exemple de fonction simple

  …

  <xsl:for-each select="catalog/cd">

  <xsl:if test="price > 10"> ←CD dont le prix est ≥ 10 €

Choix sur la sélection

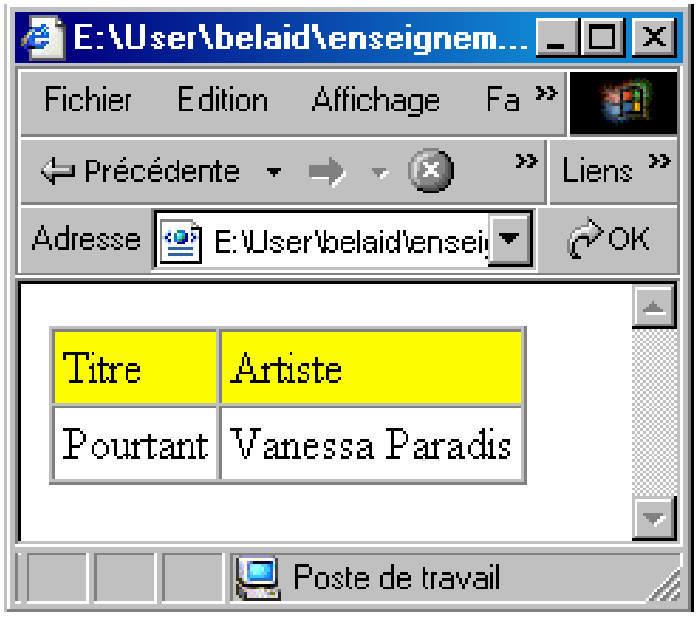
Exemple : xsldemo5.xsl

```
…
    <xsl:template match="/">
    <html>
    <body>
      <table border="1" cellspacing="0" cellpadding="3">
                <tr bgcolor="#FFFF00">
                        <td>Titre</td>
                        <td>Artiste</td>
                </tr>
                <xsl:for-each select="compilation/mp3">
                        <xsl:if test=".[artiste='Vanessa Paradis']">
                           <tr>
                                <td><xsl:value-of select="titre"/></td>
                                <td><xsl:value-of select="artiste"/></td>
                           </tr>
                        </xsl:if>
                </xsl:for-each>
    </table>
    …
```

Choix sur la sélection

Exemple : xsldemo5.xsl

- Résultat



Choix sur la sélection
Exemple : xsldemo5.xsl

- Autres exemples d'attributs (à voir plus loin)
  - <xsl:if test="not(dateRevision)">
    - /* si la valeur du nom n'est pas égale à celle de dateRevision
  - <xsl:if test="position()=2">...
    - /* si la position du nœud courant est 2 */
  - <xsl:if test="name(.)='nom'">…
    - /* si le nom du nœud courant = 'nom' */
  - <xsl:if test="@id">...
    - /* s'il existe un enfant de nom = au nom référencé par id */

Exprimer plusieurs choix sur la sélection

- <xml:choose>
  - avec : <xsl:when> et <xsl:otherwise>

```
...
 <xsl:choose>
   <xsl:when test="quelque-chose">
     [action]
   </xsl:when>
   <xsl:when test="autre-chose">
     [action]
   </xsl:when>
     ...
   <xsl:otherwise>
     [action]
   </xsl:otherwise>
 </xsl:choose>
...
```

Exprimer plusieurs choix sur la sélection : exemple

<xsl:choose>	
condition vérifiée	**<xsl:when test=".[artiste='Alain Souchon']">** <tr bgcolor="#00FF00"> <td><xsl:value-of select="titre"/></td> <td><xsl:value-of select="artiste"/></td> </tr> **</xsl:when>**
sinon	<xsl:otherwise> <tr> <td><xsl:value-of select="titre"/></td> <td><xsl:value-of select="artiste"/></td> </tr> </xsl:otherwise>
</xsl:choose>	

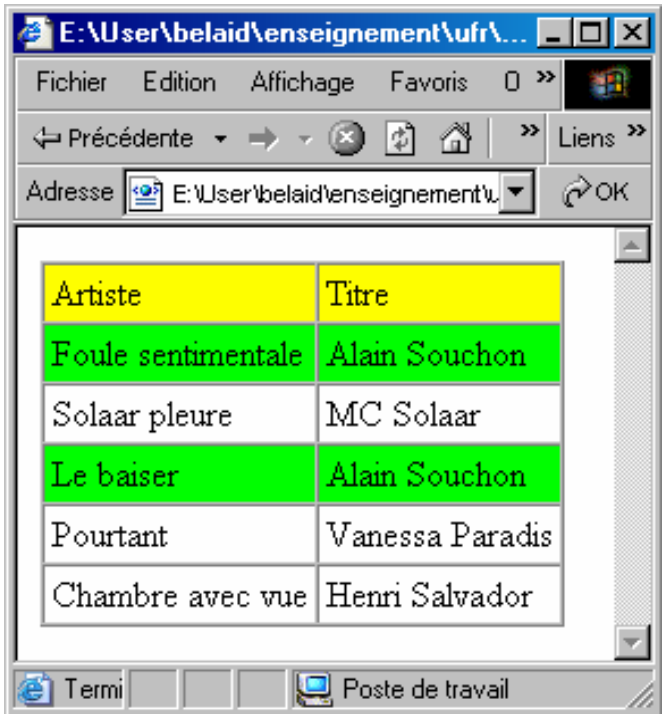
Exprimer plusieurs choix sur la sélection

Exemple : xsldemo6.xsl

```
…
   <table border="1" cellspacing="0" cellpadding="3">
   <tr bgcolor="#FFFF00">
   <td>Artiste</td>
   <td>Titre</td>
   </tr>
   <xsl:for-each select="compilation/mp3">
   <xsl:choose>
            <xsl:when test=".[artiste='Alain Souchon']">
                     <tr bgcolor="#00FF00">
                              <td><xsl:value-of select="titre"/></td>
                              <td><xsl:value-of select="artiste"/></td>
                     </tr>
            </xsl:when>
            <xsl:otherwise>
                     <tr>
                              <td><xsl:value-of select="titre"/></td>
                              <td><xsl:value-of select="artiste"/></td>
                     </tr>
            </xsl:otherwise>
   </xsl:choose>
   </xsl:for-each>
   </table>…
```

Exprimer plusieurs choix sur la sélection

Résultat : **xsldemo6.xsl**



Le langage Xpath

Définition

- Xpath est le langage de parcours de l'arbre XML et de désignation des nœuds
- La désignation peut se faire de plusieurs manières :
  - Par le nom (par le chemin)
  - Par une propriété (prédicat le qualifiant)

Le langage Xpath

Désignation par le nom

- Le nœud de départ ou racine est donné par :

  "/"

  - Si le chemin commence par '/', alors il représente un chemin absolu vers l'élément requis
- Le nœud courant est donné par :

  "."
- Le nœud de nom "x"
  - Désignation directe :
    - //x
  - Désignation par le chemin ou le niveau:
    - /nom1/nom2/…/x
    - //nom1/nom2/…/x
    - /*/*/*/nom

Exemples

Utiliser l'interface XSLT, avec le process Xpath sur cet exemple ou sur bibliotheque1.xml

- AAA : sélectionne l'élément racine AAA

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC/>
</AAA>
```

On également écrire une feuille de style avec comme règle :

```
<xsl:template match="/">
 <xsl:value-of select="AAA"/>
</xsl:template>
```

Xpath : recherche par le nom

Xpath : /bibliotheque/livre

- /AAA/CCC
  - Sélectionne tous les éléments CCC qui sont enfants de l'élément racine AAA

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC/>
</AAA>>
```

- /AAA/DDD/BBB
  - Sélectionne tous les éléments BBB qui sont enfants de DDD, qui sont enfants de l'élément racine AAA

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC/>
</AAA>
```

Xpath : recherche par le nom

'//' : tous les éléments du document qui correspondent au critère qui suit sont sélectionnés : //livre

- //BBB
  - Sélectionne tous les éléments BBB

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC>
        <DDD>
            <BBB/>
            <BBB/>
        </DDD>
    </CCC>
</AAA>
```

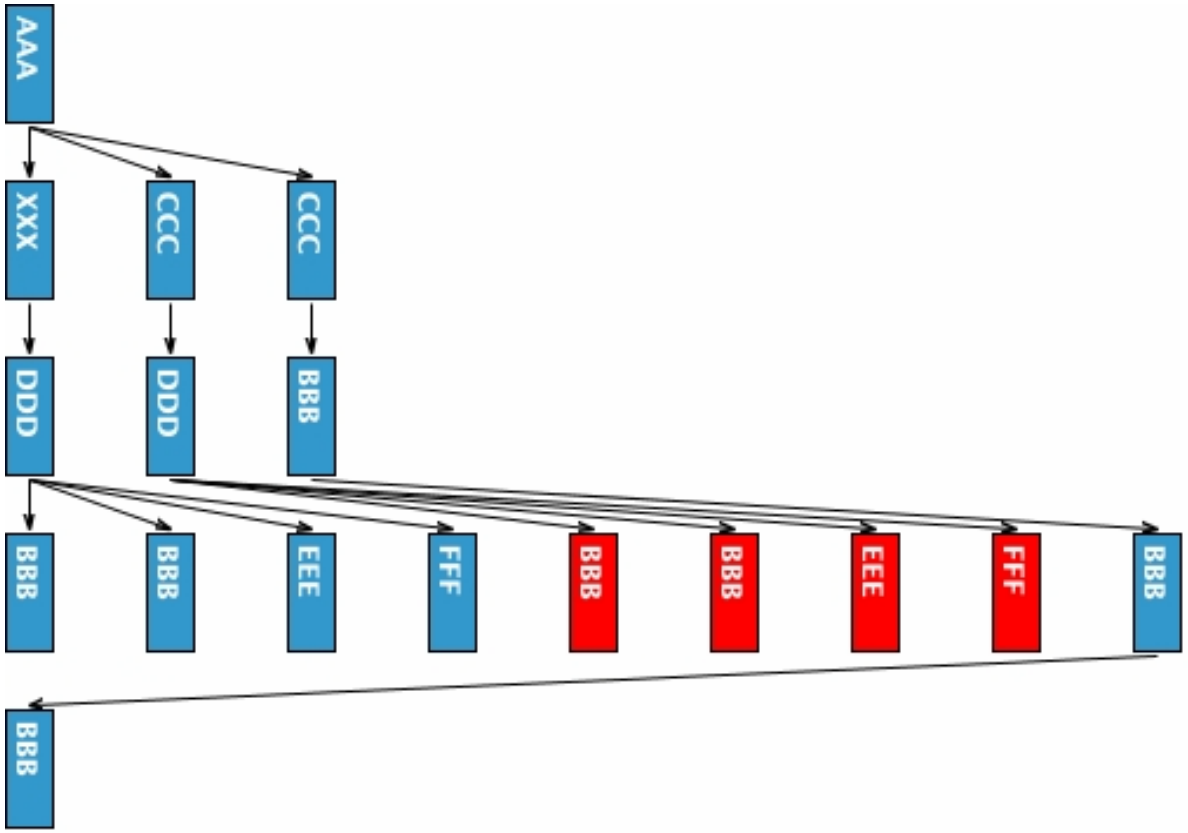
- //DDD/BBB
  - Sélectionne tous les éléments BBB qui sont enfants de DDD

```
<AAA>
    <BBB/>
    <CCC/>
    <BBB/>
    <DDD>
        <BBB/>
    </DDD>
    <CCC>
        <DDD>
            <BBB/>
            <BBB/>
        </DDD>
    </CCC>
</AAA>
```

Xpath : recherche par le nom

L'étoile * sélectionne tous les éléments localisés par ce qui la précède dans le chemin : /bibliotheque/livre/*

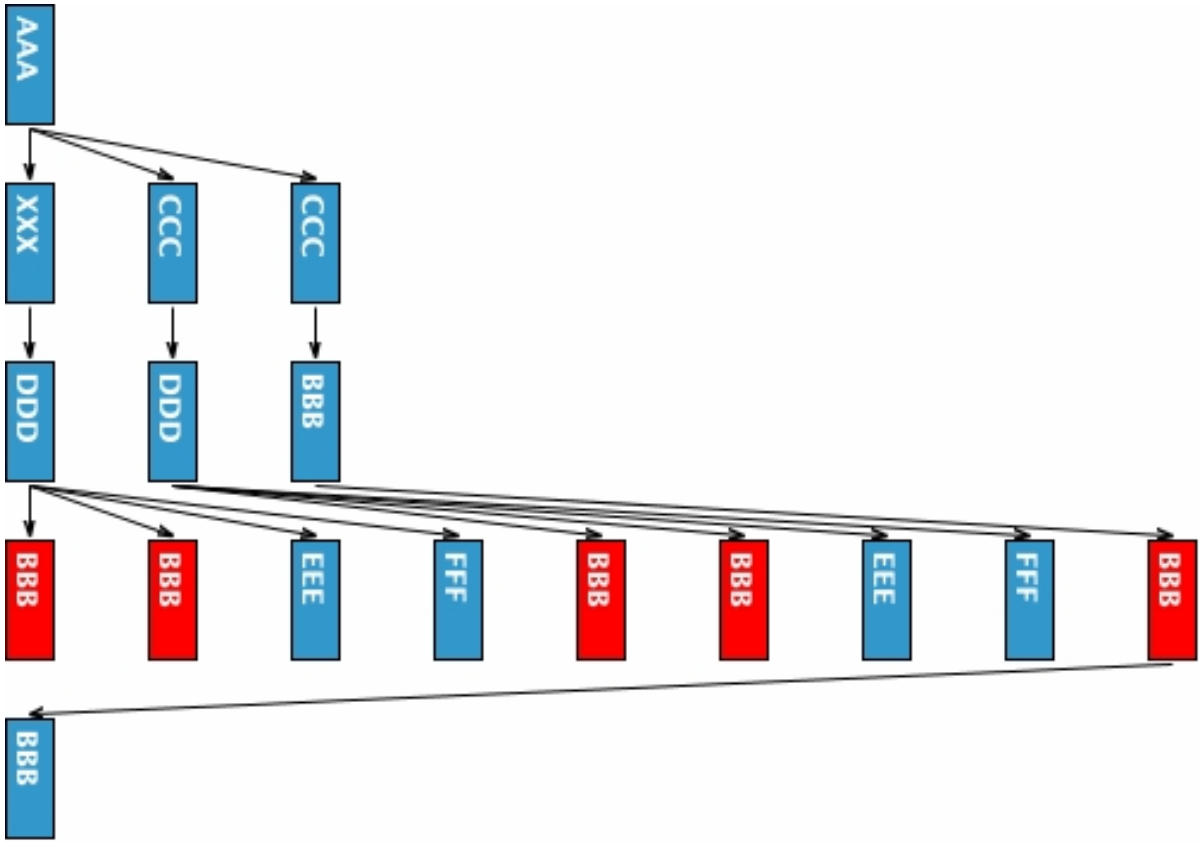
- /AAA/CCC/DDD/*
  - Sélectionne tous les éléments inclus dans les éléments /AAA/CCC/DDD



Xpath : recherche par le nom

/*/*/titre

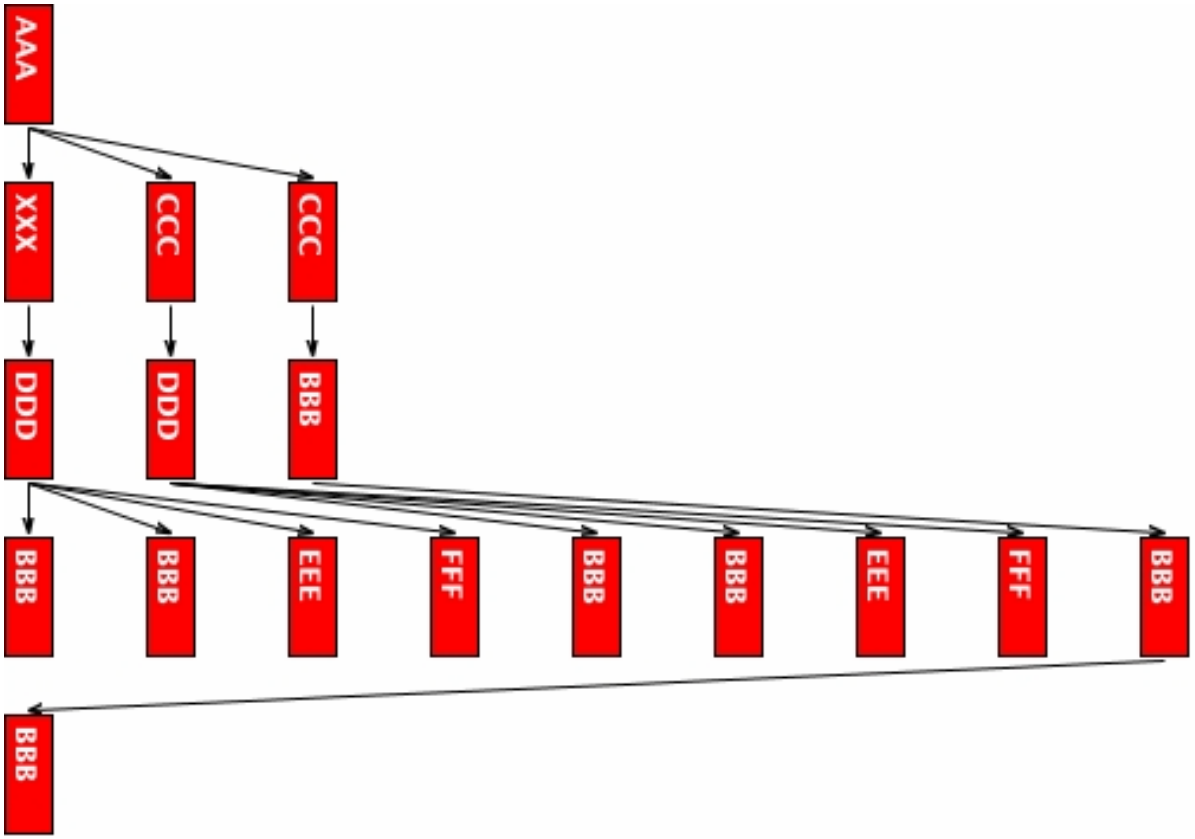
- /*/*/*/BBB
  - Sélectionne tous les éléments BBB qui ont trois ancêtres



Xpath : recherche par le nom

//*

- //*
  - Sélectionne tous les éléments



Xpath : recherche par le prédicat

- Fonctions de sélection
  - element**[n]**
    - sélectionne le n^{ième} élément *element* dans le nœud courant
  - element**[elt]**
    - sélectionne dans le noeud courant, l'élément *element* qui a comme élément fils *elt*
  - **[elt=**"valeur"**]**
    - sélectionne dans le nœud courant, l'élément ayant pour fils un nœud *elt* qui a une valeur égale à *valeur*
  - element**[@attribut]**
    - sélectionne dans le nœud courant, l'élément *element* qui possède un attribut *attribut*
  - **[@attribut=**'valeur']
    - sélectionne dans le nœud courant, l'élément dont l'attribut *attribut* a une valeur égale à *valeur*

Xpath : recherche par le prédicat

/bibliotheque/livre[2]

Un nombre entre crochets donne la position d'un élément dans le jeu sélectionné. La fonction last sélectionne le dernier élément du jeu

- /AAA/BBB[1]
  - Sélectionne le premier élément BBB, fils de l'élément racine AAA

```
<AAA>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
</AAA>
```

- /AAA/BBB[last()]
  - Sélectionne le dernier élément BBB, fils de l'élément racine AAA

```
<AAA>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
</AAA>
```

Xpath : le prédicat est un attribut

Les attributs sont spécifiés par le préfixe @

- //@id
  - Sélectionne tous les attributs id

```
<AAA>
    <BBB id = "b1"/>
    <BBB id = "b2"/>
    <BBB name = "bbb"/>
    <BBB/>
</AAA>
```

- //BBB[@id]
  - Sélectionne tous les BBB qui ont un attribut id

```
<AAA>
    <BBB id = "b1"/>
    <BBB id = "b2"/>
    <BBB name = "bbb"/>
    <BBB/>
</AAA>
```

Xpath : le prédicat est un attribut

Les attributs sont spécifiés par le préfixe @

- //BBB[@name]
  - Sélectionne tous BBB qui ont un attribut name

```
<AAA>
    <BBB id = "b1"/>
    <BBB id = "b2"/>
    <BBB name = "bbb"/>
    <BBB/>
</AAA>
```

- //BBB[@*]
  - Sélectionne tous BBB qui ont un attribut

```
<AAA>
    <BBB id = "b1"/>
    <BBB id = "b2"/>
    <BBB name = "bbb"/>
    <BBB/>
</AAA>
```

Xpath : le prédicat est un attribut

- //BBB[not(@*)]
  - Sélectionne tous les BBB qui n'ont pas d'attribut

```
<AAA>
    <BBB id = "b1"/>
    <BBB id = "b2"/>
    <BBB name = "bbb"/>
    <BBB/>
</AAA>
```

Xpath : le prédicat est un attribut

Les valeurs d'attributs peuvent être utilisées comme critère de sélection

- //BBB[@id='b1']
  - Sélectionne tous les éléments BBB ayant un attribut id dont la valeur est b1

```
<AAA>
    <BBB id = "b1"/>
    <BBB name = " bbb "/>
    <BBB name = "bbb"/>
</AAA>
```

- //BBB[@name='bbb']
  - Sélectionne tous les éléments BBB ayant un attribut name dont la valeur est bbb

```
<AAA>
    <BBB id = "b1"/>
    <BBB name = " bbb "/>
    <BBB name = "bbb"/>
</AAA>
```

Xpath : le prédicat est un attribut

La fonction normalize-space supprime les espaces de début et de fin puis remplace les séquences d'espaces blancs par un seul espace

- //BBB[normalize-space(@name)='bbb']
  - Sélectionne tous les éléments BBB ayant un attribut name dont la valeur est bbb. Les espaces de début et de fin sont supprimés avant la comparaison

```
<AAA>
    <BBB id = "b1"/>
    <BBB name = " bbb "/>
    <BBB name = "bbb"/>
</AAA>
```

Xpath : le prédicat est une fonction

La fonction count() compte le nombre d'éléments sélectionnés

//*[count(livre)=3]

- //*[count(BBB)=2]
  - Sélectionne les éléments ayant deux enfants BBB

```
<AAA>
   <CCC>
      <BBB/>
      <BBB/>
      <BBB/>
   </CCC>
   <DDD>
      <BBB/>
      <BBB/>
   </DDD>
   <EEE>
      <CCC/>
      <DDD/>
   </EEE>
</AAA>
```

- //*[count(*)=2]
  - Sélectionne les éléments ayant deux enfants

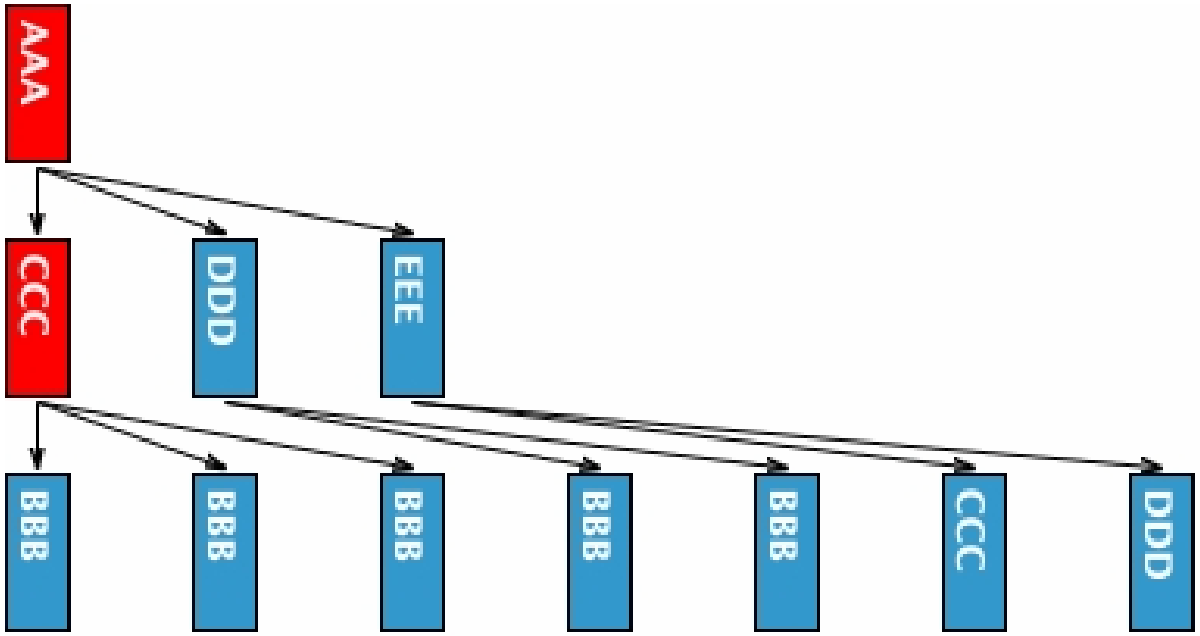
```
<AAA>
   <CCC>
      <BBB/>
      <BBB/>
      <BBB/>
   </CCC>
   <DDD>
      <BBB/>
      <BBB/>
   </DDD>
   <EEE>
      <CCC/>
      <DDD/>
   </EEE>
</AAA>
```

Xpath : le prédicat est une fonction

//*[count(*)=3]

La fonction count() compte le nombre d'éléments sélectionnés

- //*[count(*)=3]
  - Sélectionne les éléments ayant trois enfants

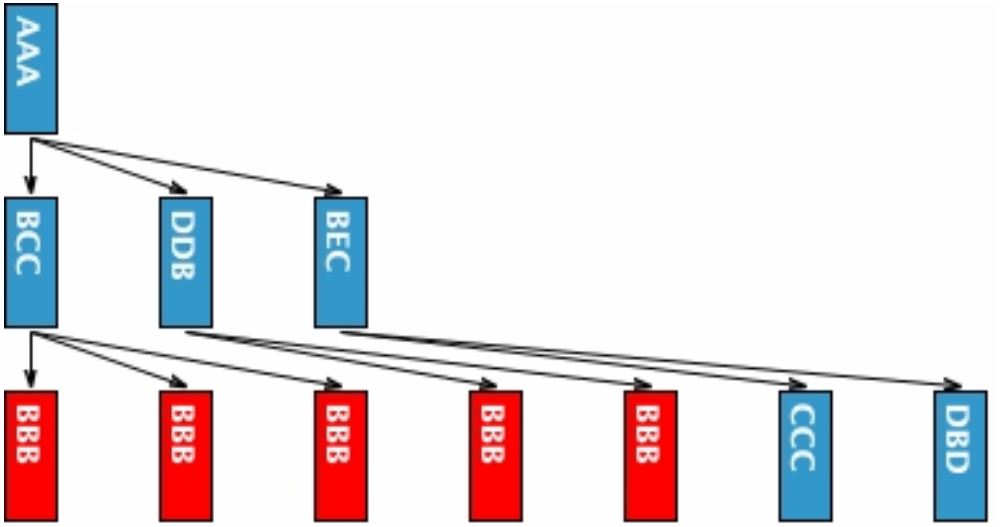


Xpath : le prédicat est une fonction

//*[name()='livre']

- //*[name()='BBB']

La fonction name() retourne le nom de l'élément, ici tous les BBB



Xpath : le prédicat est une fonction

//*[starts-with(name(),'t')]

la fonction start-with retourne vrai si la chaîne du premier argument commence par celle du deuxième

- //*[starts-with(name(),'B')]

```
<AAA>
  <BCC>
    <BBB/>
    <BBB/>
    <BBB/>
  </BCC>
  <DDB>
    <BBB/>
    <BBB/>
  </DDB>
  <BEC>
    <CCC/>
    <DBD/>
  </BEC>
</AAA>
```

Xpath : le prédicat est une fonction

//*[contains(name(),'i')]

La fonction contains retourne vrai si la chaîne du premier argument contient celle du deuxième

- //*[contains(name(),'C')]

```
<AAA>
  <BCC>
    <BBB/>
    <BBB/>
    <BBB/>
  </BCC>
  <DDB>
    <BBB/>
    <BBB/>
  </DDB>
  <BEC>
    <CCC/>
    <DBD/>
  </BEC>
</AAA>
```

Xpath : le prédicat est une fonction

//*[string-length(name()) = 5]

La fonction string-length retourne le nombre de caractères dans une chaîne.
Vous devez utiliser < comme substitutif de < et &gt comme substitutif de >

- //*[string-length(name()) = 3]

```
<AAA>
    <Q/>
    <SSSS/>
    <BB/>
    <CCC/>
    <DDDDDDDD/>
    <EEEE/>
</AAA>
```

- //*[string-length(name()) < 3]

```
<AAA>
    <Q/>
    <SSSS/>
    <BB/>
    <CCC/>
    <DDDDDDDD/>
    <EEEE/>
</AAA>
```

Xpath : le prédicat est une fonction

- //*[string-length(name()) > 3]

```
<AAA>
    <Q/>
    <SSSS/>
    <BB/>
    <CCC/>
    <DDDDDDDD/>
    <EEEE/>
</AAA>
```

Xpath : le prédicat est une fonction

- L'opérateur div réalise une division à virgule flottante, l'opérateur mod retourne le reste d'une division. La fonction floor() retourne le plus grand nombre (le plus près de l'infini positif) qui n'est pas plus grand que l'argument et qui est un entier. La fonction ceiling() retourne le plus petit nombre (le plus près de l'infini négatif) qui n'est pas plus petit que l'argument et qui est un entier
- //BBB[position() mod 2 = 0]

```
<AAA>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <BBB/>
  <CCC/>
  <CCC/>
  <CCC/>
</AAA>
```

Xpath : le prédicat est une fonction

- //BBB[position() = floor(last() div 2 + 0.5) or position() = ceiling(last() div 2 + 0.5)]

```
<AAA>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
    <BBB/>
    <CCC/>
    <CCC/>
    <CCC/>
</AAA>
```

Xpath : le prédicat est une fonction

- //CCC[position() = floor(last() div 2 + 0.5) or position() = ceiling(last() div 2 + 0.5)]

```
<AAA>
   <BBB/>
   <BBB/>
   <BBB/>
   <BBB/>
   <BBB/>
   <BBB/>
   <BBB/>
   <BBB/>
   <CCC/>
   <CCC/>
   <CCC/>
</AAA>
```

Xpath : le prédicat est une fonction

//livre | //titre

Plusieurs chemins peuvent être combinés avec le séparateur |

- //CCC | //BBB
  - Sélectionne tous les éléments CCC et BBB

```
<AAA>
    <BBB/>
    <CCC/>
    <DDD>
        <CCC/>
    </DDD>
    <EEE/>
</AAA>
```

- /AAA/EEE | //BBB
  - Sélectionne tous les éléments BBB et EEE qui sont enfants de l'élément racine AAA

```
<AAA>
    <BBB/>
    <CCC/>
    <DDD>
        <CCC/>
    </DDD>
    <EEE/>
</AAA>
```

Xpath : recherche par les fonctions

- /AAA/EEE | //DDD/CCC | /AAA | //BBB
  - Le nombre de combinaisons n'est pas restreint

```
<AAA>
    <BBB/>
    <CCC/>
    <DDD>
        <CCC/>
    </DDD>
    <EEE/>
</AAA>
```

Le langage Xpath

Recherche par les orientations ou les axes

- Plusieurs axes de désignation
  - descendant — tous les descendants du nœud courant (sauf lui-même)
  - ancestor — tous les nœuds qui contiennent le nœud courant
  - ancestor-or-self — tous les ascendants du nœud courant ET le nœud courant
  - following-sibling — tous les nœuds qui suivent le nœud courant et sont fils du même père
  - preceding-sibling — tous les nœuds qui précèdent le nœud courant et sont fils du même père
  - following — tous les nœuds qui suivent le nœud courant, de type autre que attribut ou namespace
  - preceding — tous les nœuds qui précèdent le nœud courant, de type autre que attribut ou namespace
  - namespace — tous les nœuds de type namespace déclarés au niveau du nœud courant ou de l'un de ses ancêtres

Xpath : recherche avec les axes

/child::bibliotheque

- L'axe enfant contient les enfants du nœud contextuel
- L'axe enfant est celui par défaut et il peut être omis

- /AAA
  - Équivalent à /child::AAA

```
<AAA>
    <BBB/>
    <CCC/>
</AAA>
```

- /child::AAA
  - Équivalent à /AAA

```
<AAA>
    <BBB/>
    <CCC/>
</AAA>
```

Xpath : recherche avec les axes

/child::bibliotheque/child::livre

- /AAA/BBB
  - Equivalent à /child::AAA/child::BBB

```
<AAA>
    <BBB/>
    <CCC/>
</AAA>
```

- /child::AAA/child::BBB
  - Equivalent à /AAA/BBB

```
<AAA>
    <BBB/>
    <CCC/>
</AAA>
```

Xpath : recherche avec les axes

- /child::AAA/BBB
  - Les deux possibilités peuvent être combinées

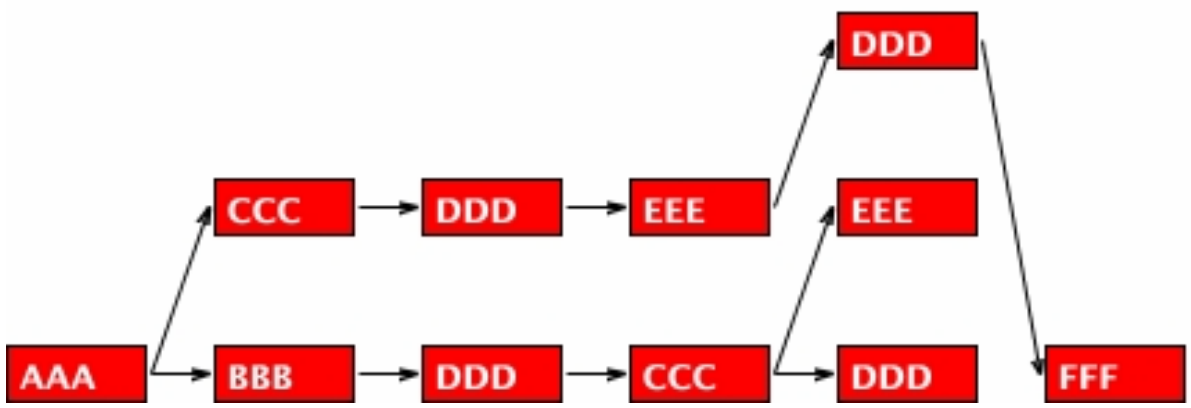
```
<AAA>
    <BBB/>
    <CCC/>
</AAA>
```

Xpath : recherche avec les axes

/descendant::*

L'axe descendant contient les descendants du nœud contextuel; un descendant est un enfant ou un petit enfant, etc. Aussi, l'axe descendant ne contient jamais de nœud de type attribut ou des noms d'espace

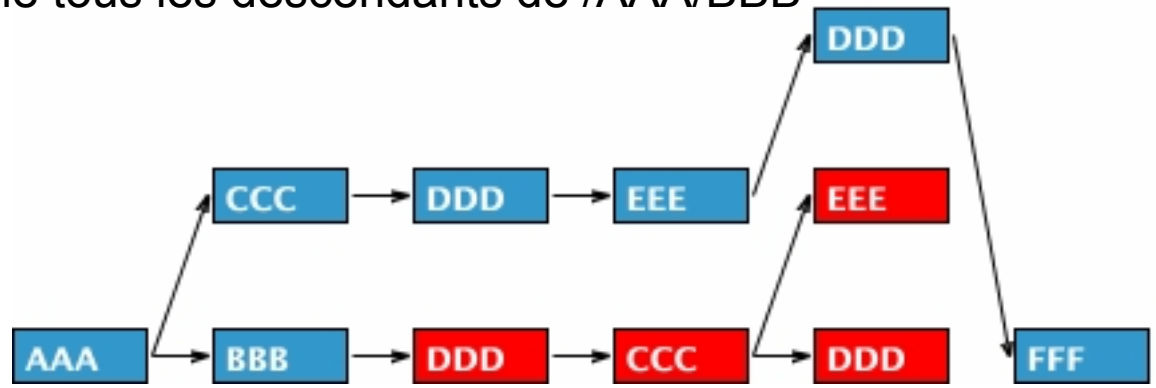
- /descendant::*
  - Sélectionne tous les descendants de l'élément racine et donc tous les éléments



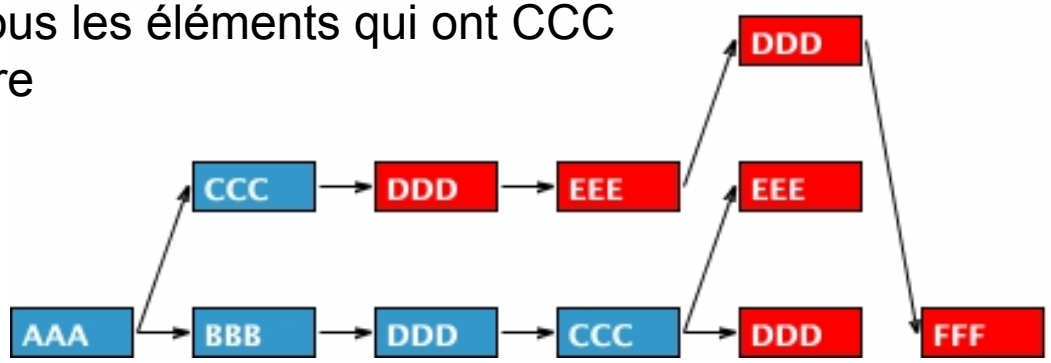
Xpath : recherche avec les axes

/bibliotheque/livre/descendant::* equiv à //livre/descendant::*

- /AAA/BBB/descendant::*
  - Sélectionne tous les descendants de /AAA/BBB



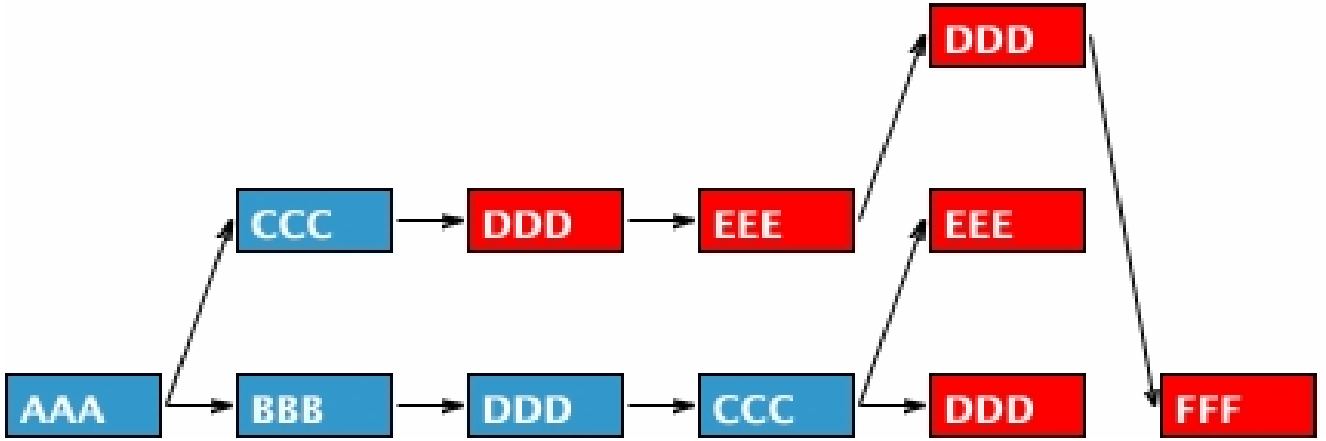
- //CCC/descendant::*
  - Sélectionne tous les éléments qui ont CCC comme ancêtre



Xpath : recherche avec les axes

//bibliotheque/descendant::titre

- //CCC/descendant::DDD
  - Sélectionne les descendants de CCC dont au moins un est un DDD

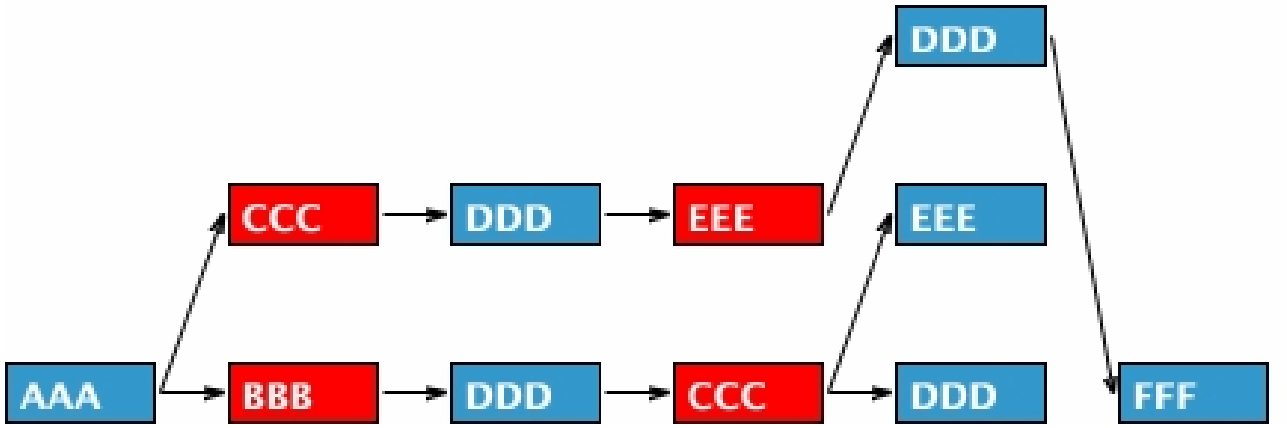


Xpath : recherche avec les axes

//livre/parent::*

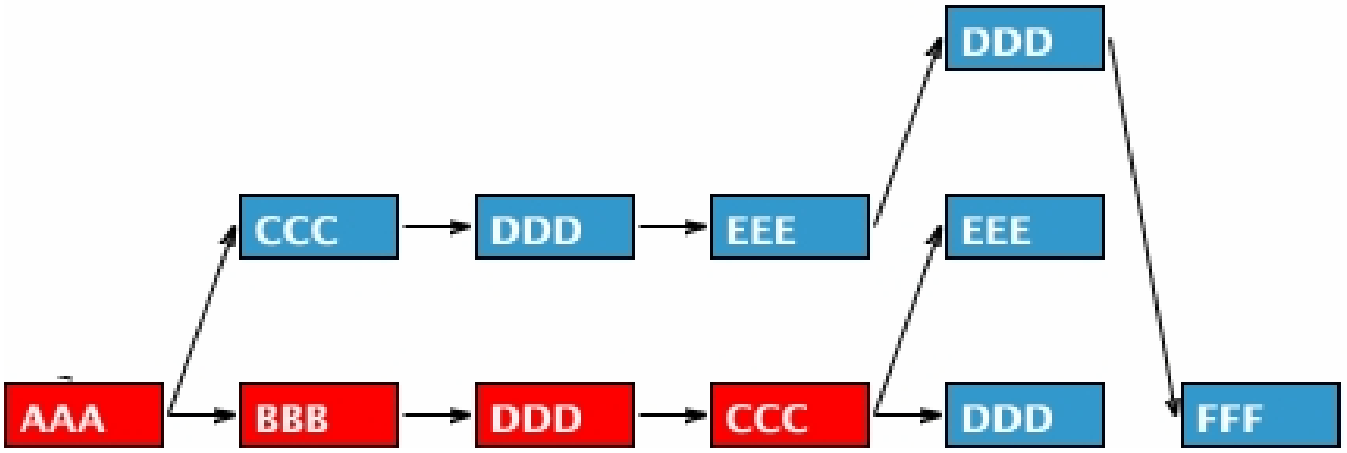
L'axe "parent" contient le parent du nœud contextuel s'il en a un

- //DDD/parent::*



Xpath : recherche avec les axes

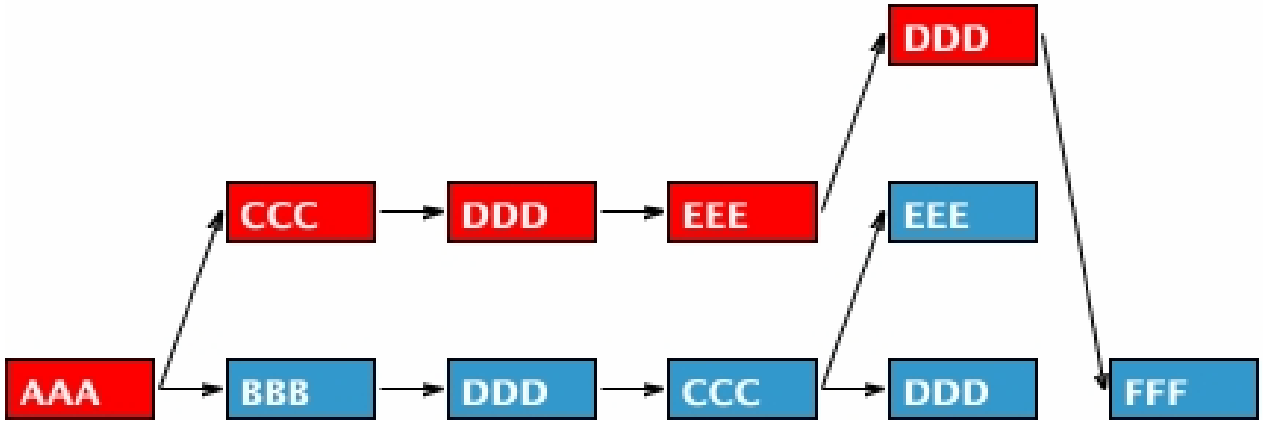
- L'axe ancêtre (ancestor) contient les ancêtres du nœud contextuel; cela comprend son parent et les parents des parents, etc. Aussi, cet axe contient toujours le nœud racine, sauf si le nœud contextuel est lui-même la racine
- /AAA/BBB/DDD/CCC/EEE/ancestor::*
  - Sélectionne tous les éléments donnés dans ce chemin absolu



Xpath : recherche avec les axes

//auteur/ancestor::*

- //FFF/ancestor::*
  - Sélectionne tous les ancêtres de l'élément FFF

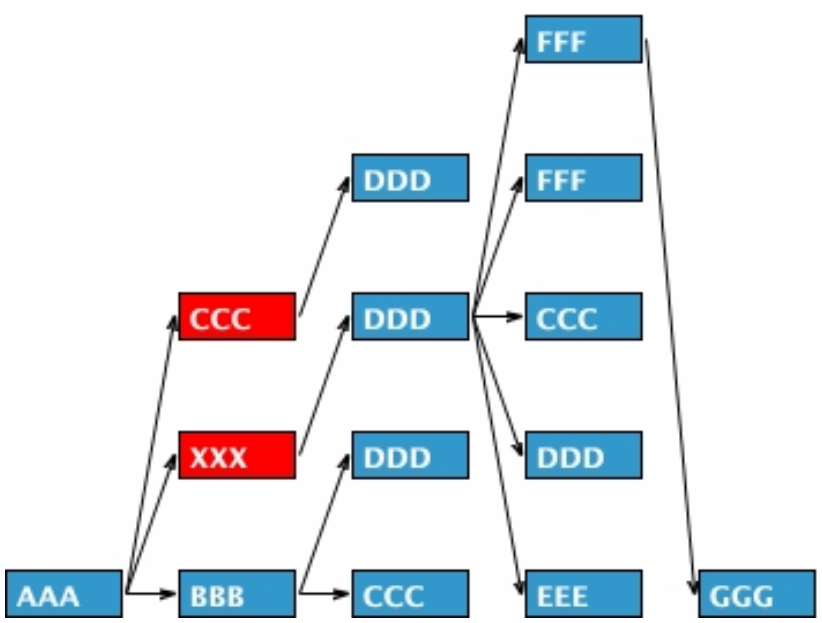


Xpath : recherche avec les axes

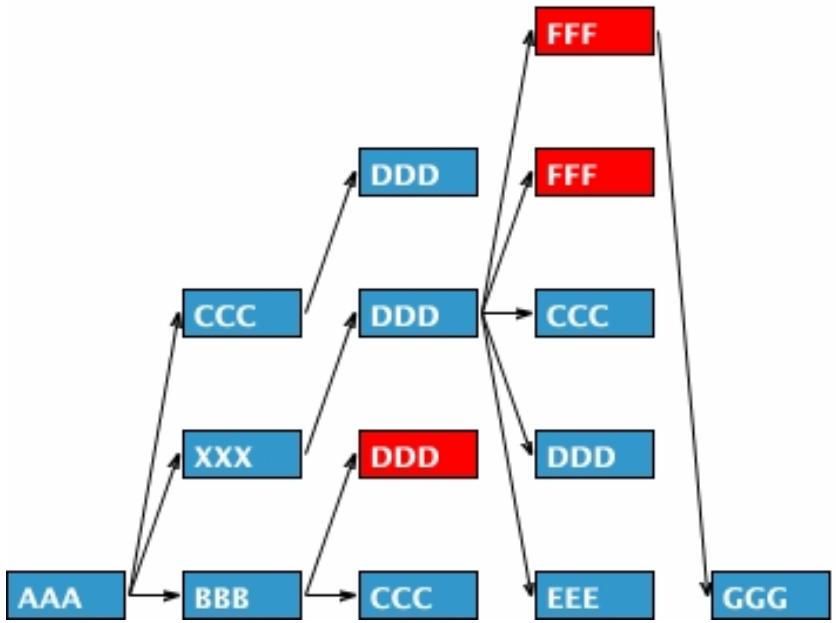
/bibliotheque/livre/titre/following-sibling::*

L'axe 'following-sibling' contient tous les nœuds frères qui suivent le nœud contextuel

/AAA/BBB/following-sibling::*



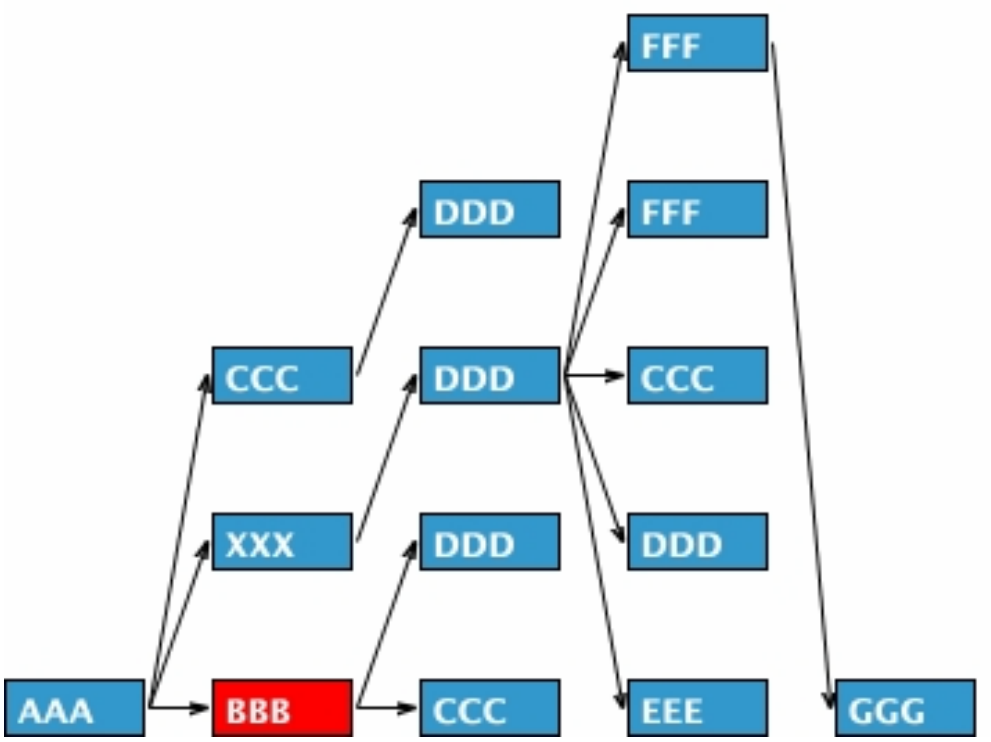
//CCC/following-sibling::*



Xpath : recherche avec les axes

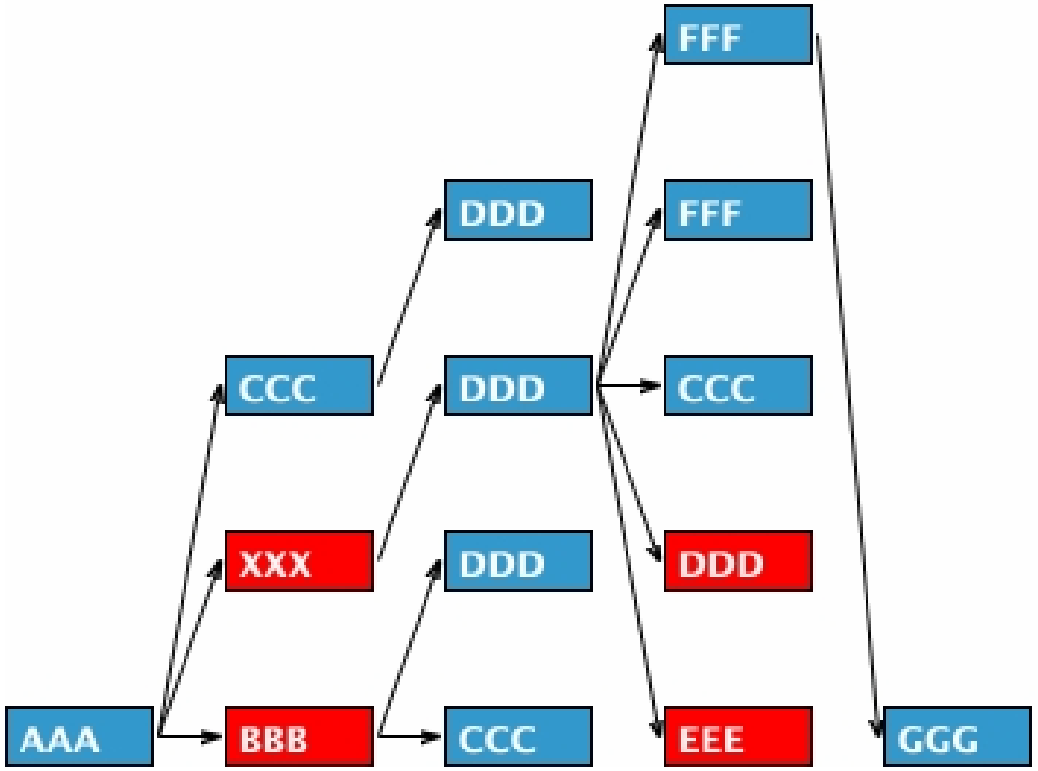
L'axe 'preceding-sibling' contient tous les frères prédécesseurs du nœud contextuel; si le nœud contextuel est un attribut ou un espace de noms, la cible précédente est vide

- /AAA/XXX/preceding-sibling::*



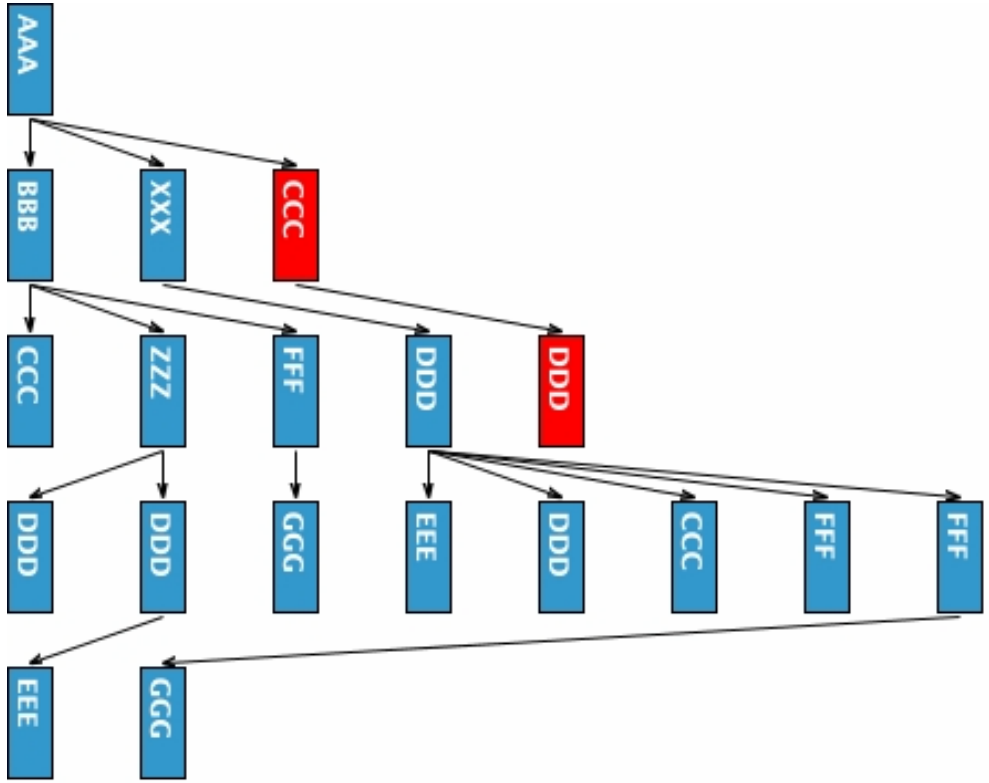
Xpath : recherche avec les axes

- //CCC/preceding-sibling::*



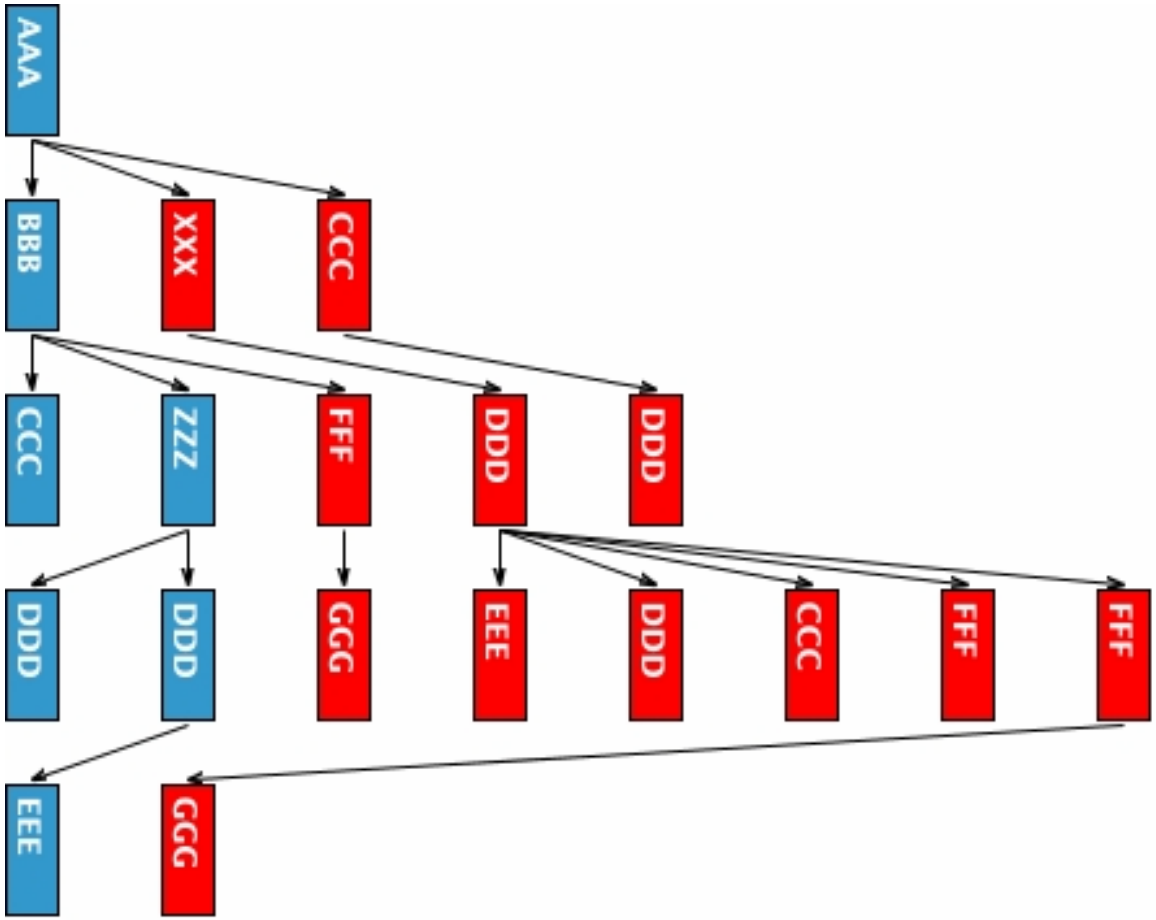
Xpath : recherche avec les axes

- L'axe suivant (following) tous les nœuds du même document que le nœud contextuel qui sont après le nœud contextuel dans l'ordre du document, à l'exclusion de tout descendant, des attributs et des espaces de noms
- /AAA/XXX/following::*



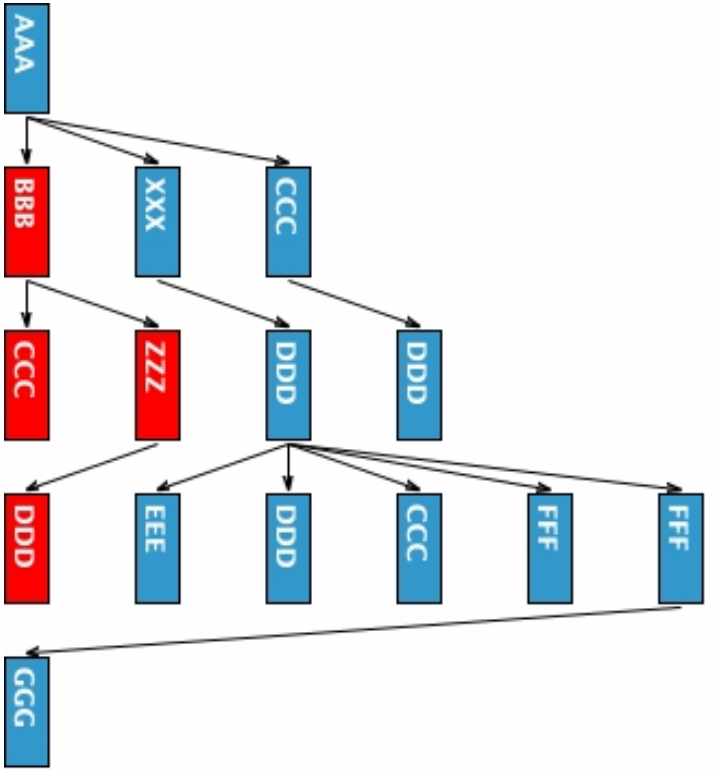
Xpath : recherche avec les axes

- //ZZZ/following::*



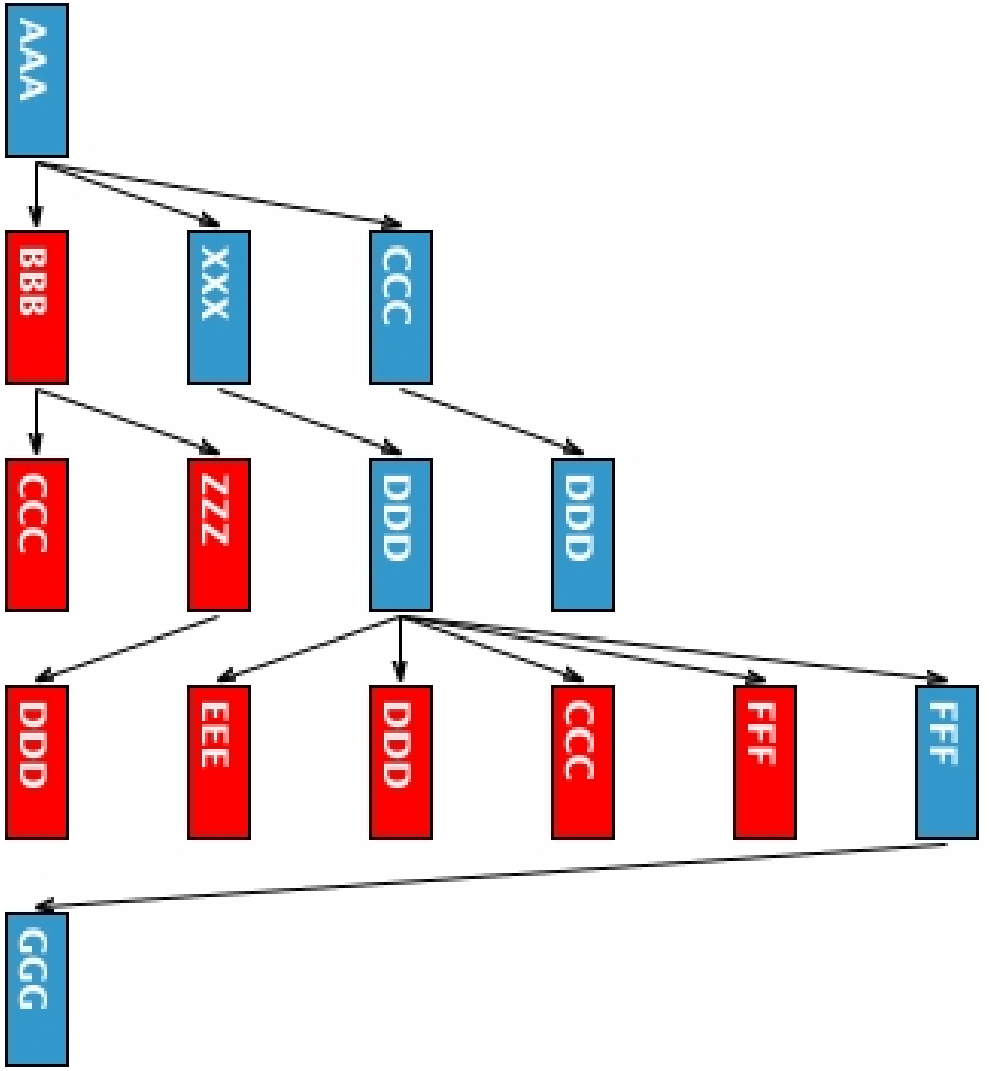
Xpath : recherche avec les axes

- L'axe cible précédente (preceding-sibling) contient tous les prédécesseurs du nœud contextuel; si le nœud contextuel est un attribut ou un espace de noms, la cible précédente est vide
- /AAA/XXX/preceding::*



Xpath : recherche avec les axes

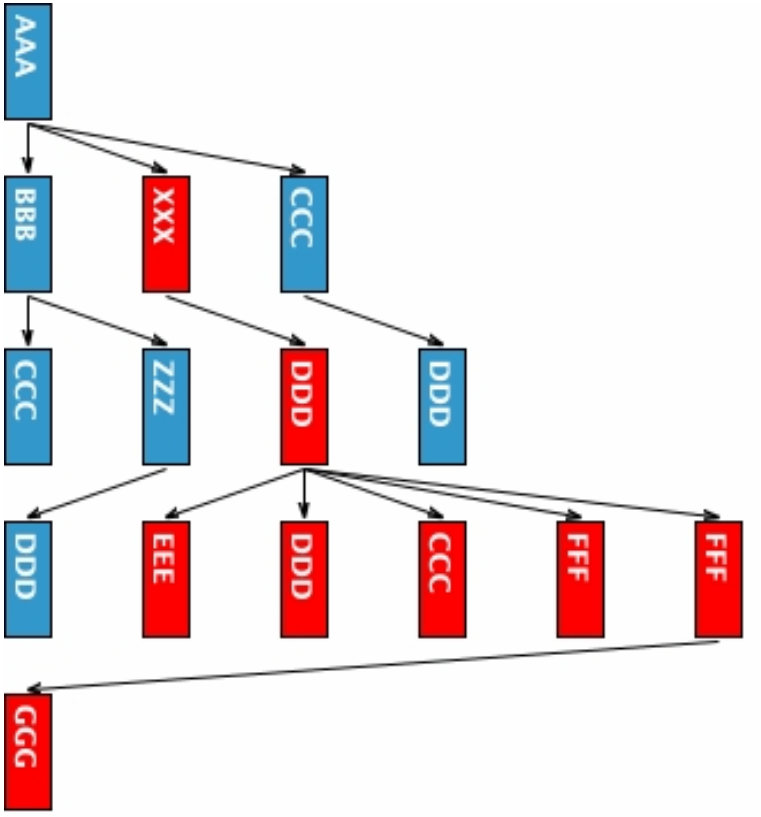
- //GGG/preceding::*



Xpath : recherche avec les axes

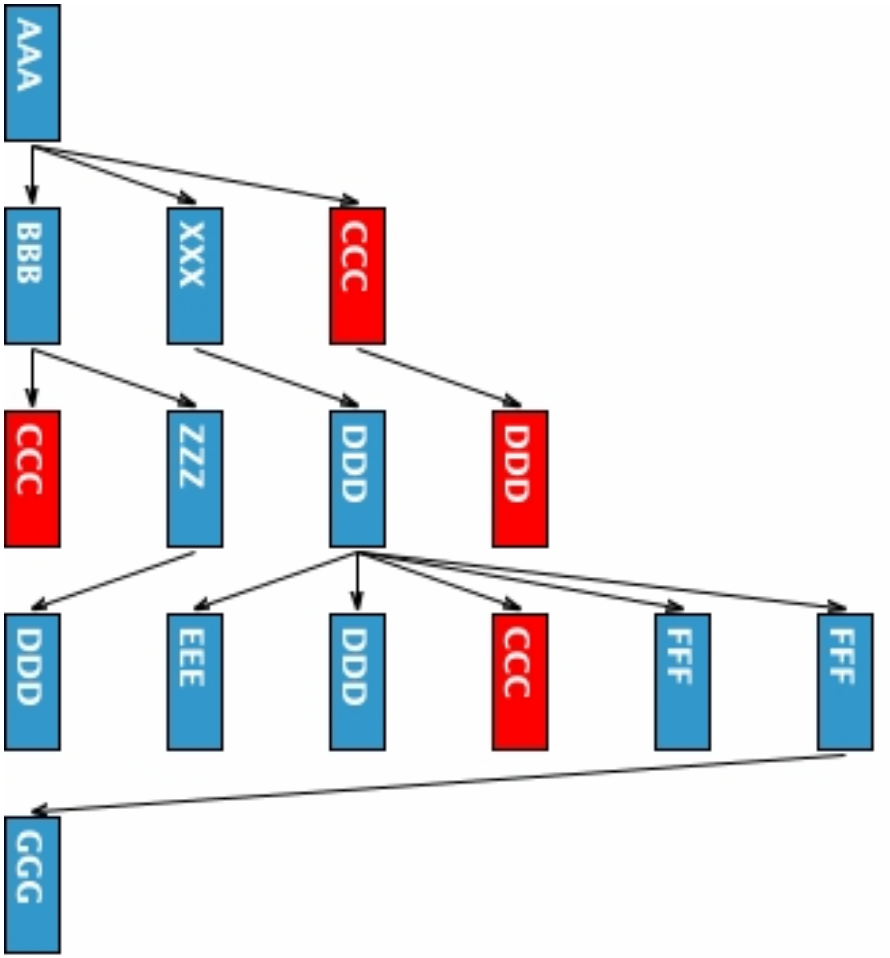
/bibliotheque/livre/descendant-or-self::*

- L'axe 'descendant-or-self' contient le nœud contextuel et ses descendants
- /AAA/XXX/descendant-or-self::*



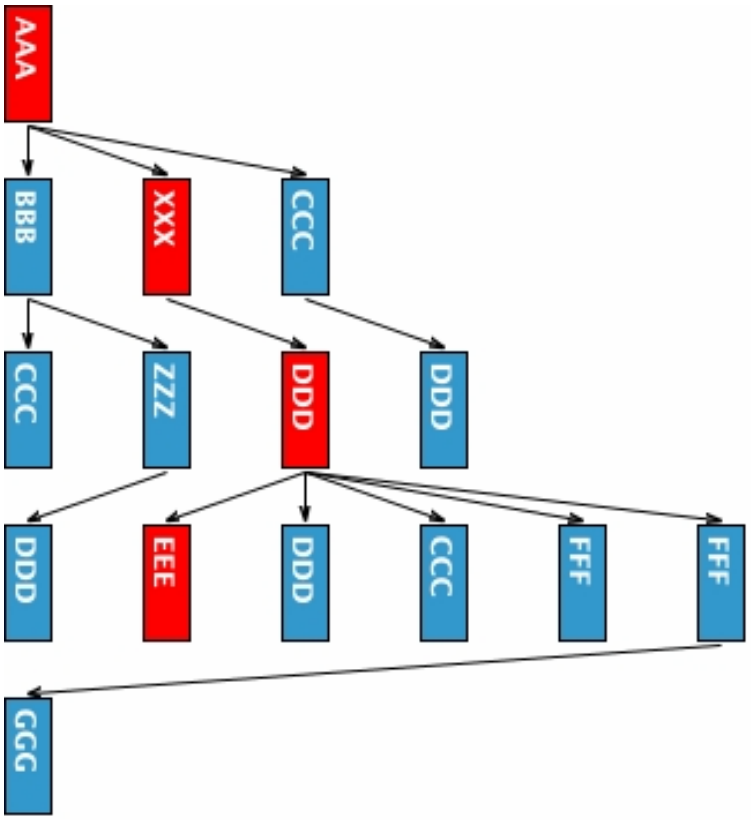
Xpath : recherche avec les axes

- //CCC/descendant-or-self::*



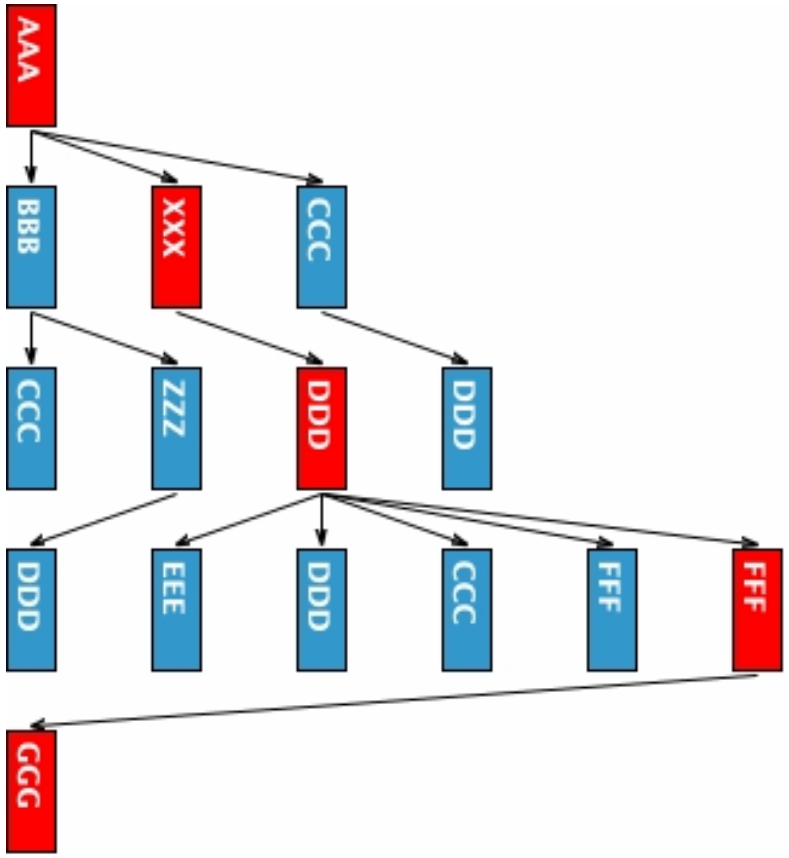
Xpath : recherche avec les axes

- L'axe ancestor-or-self contient le nœud contextuel et ses ancêtres; ainsi l'axe ancestor-or-self contient toujours le nœud racine
- /AAA/XXX/DDD/EEE/ancestor-or-self::*



Xpath : recherche avec les axes

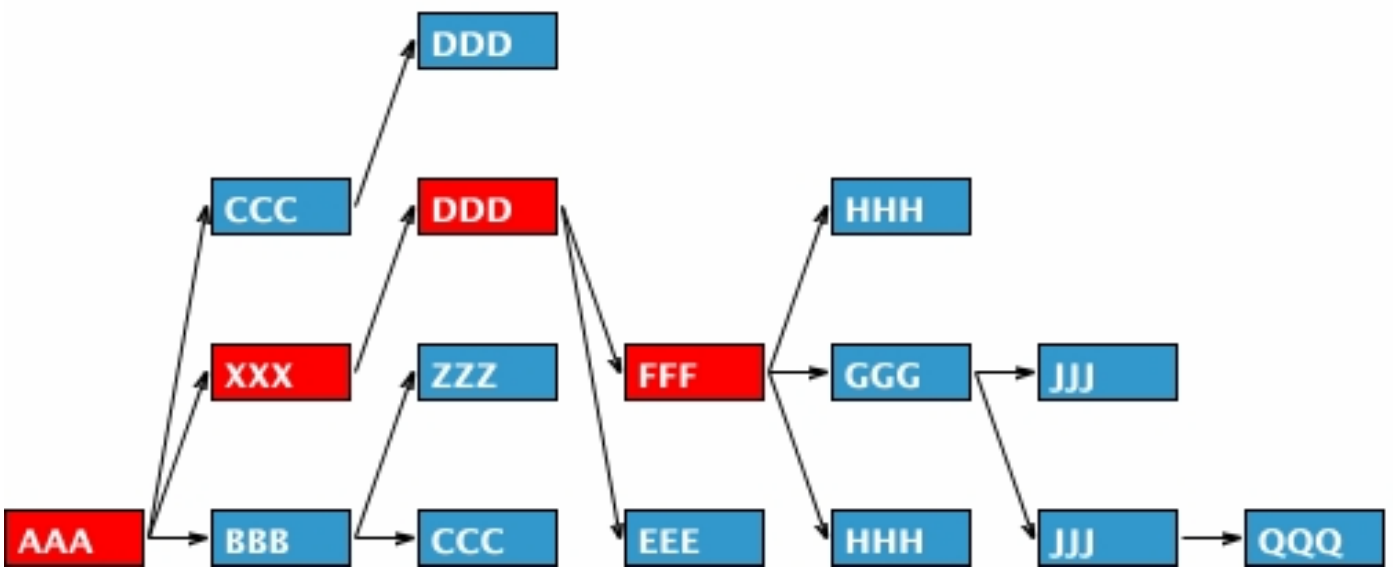
- //GGG/ancestor-or-self::*



Xpath : recherche avec les axes

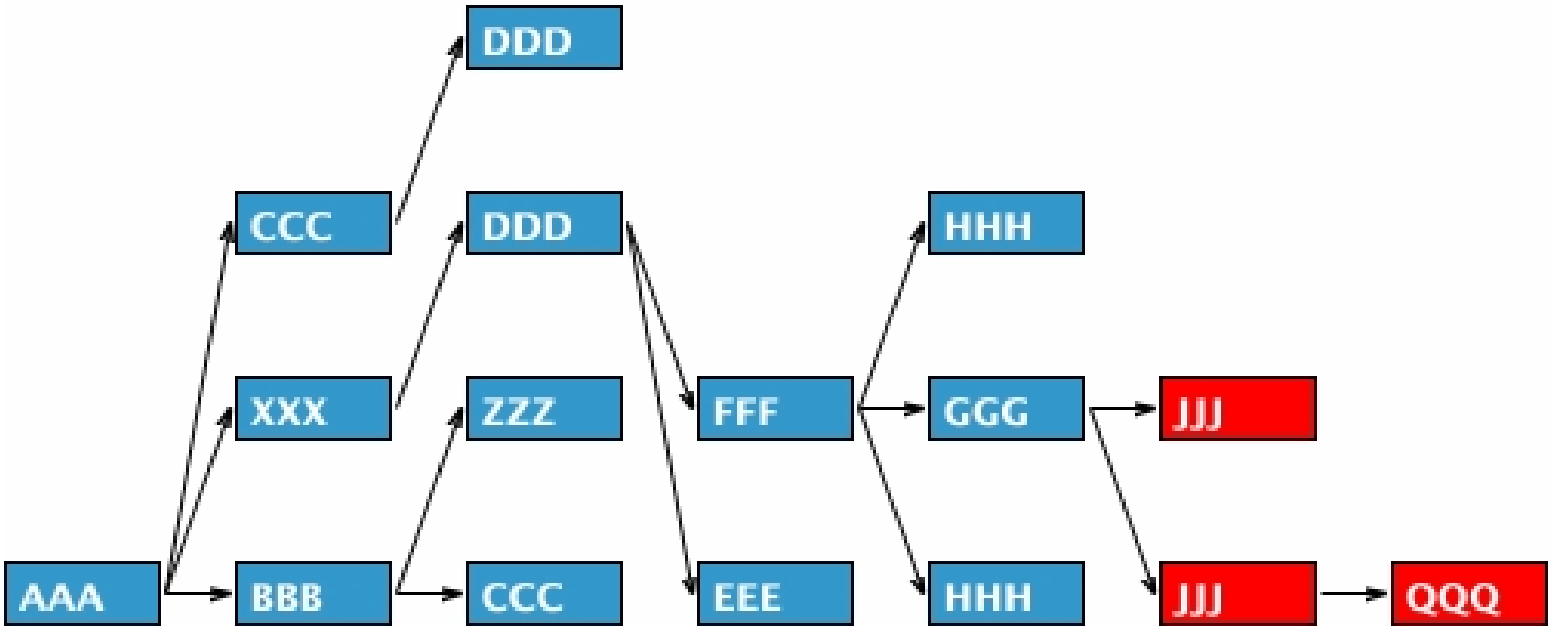
//livre/ancestor::*

- Les axes ancestor, descendant, following, preceding-or-self partitionnent un document (ignorant les attributs et les nœuds d'espace de nom) : ils ne se chevauchent pas et ensemble ils contiennent tous les nœuds d'un document
- //GGG/ancestor::*



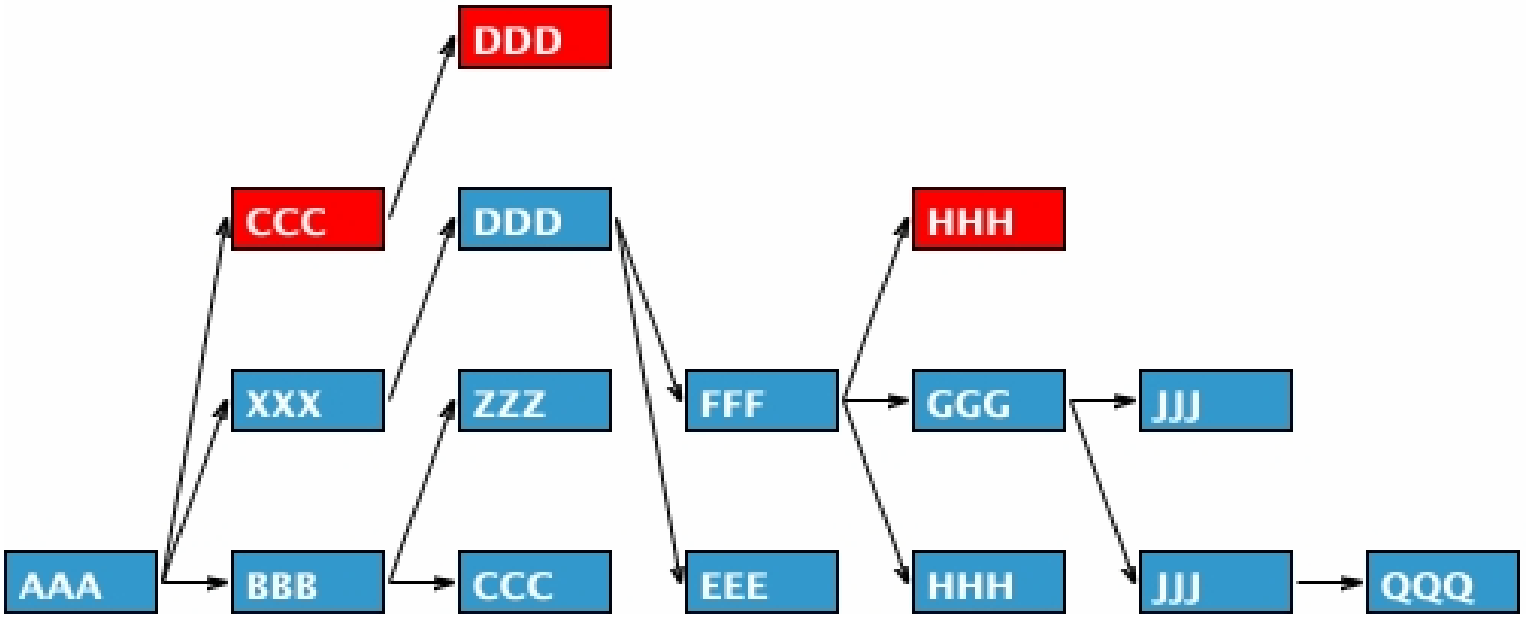
Xpath : recherche avec les axes

- //GGG/descendant::*



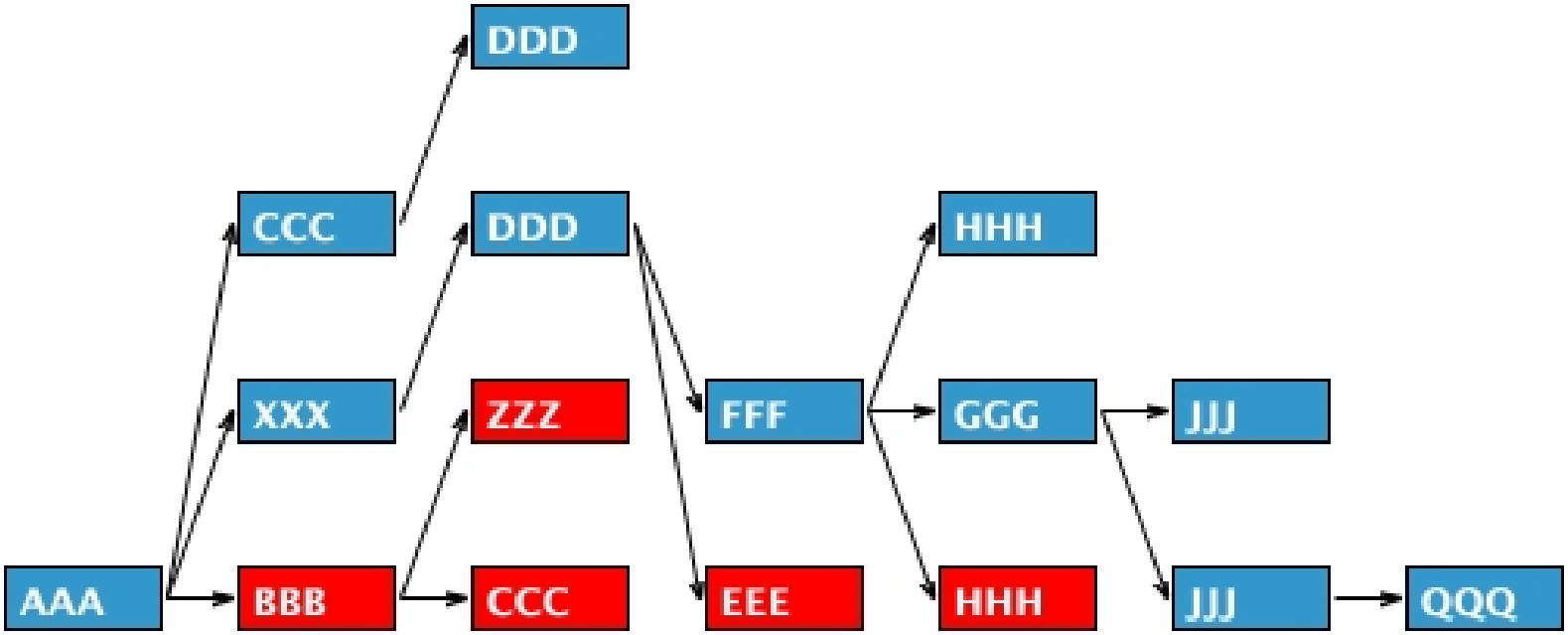
Xpath : recherche avec les axes

- //GGG/following::*



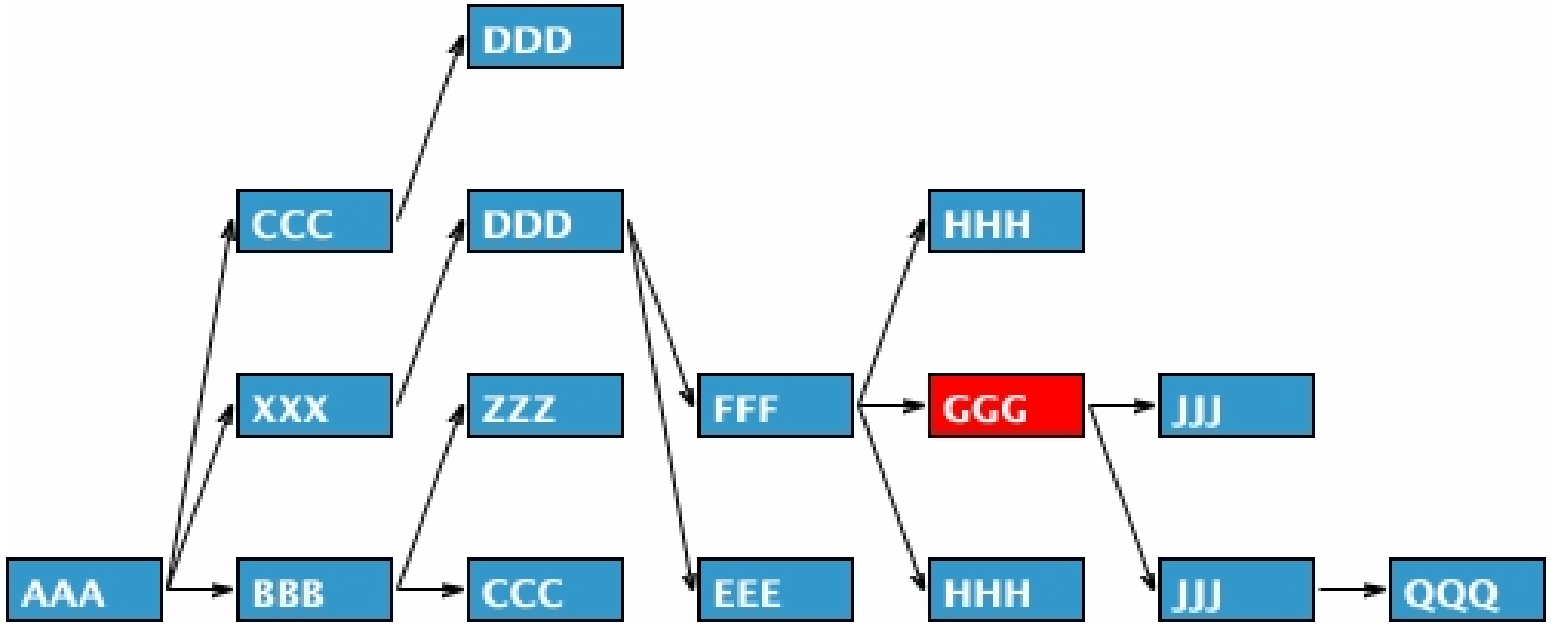
Xpath : recherche avec les axes

- //GGG/preceding::*



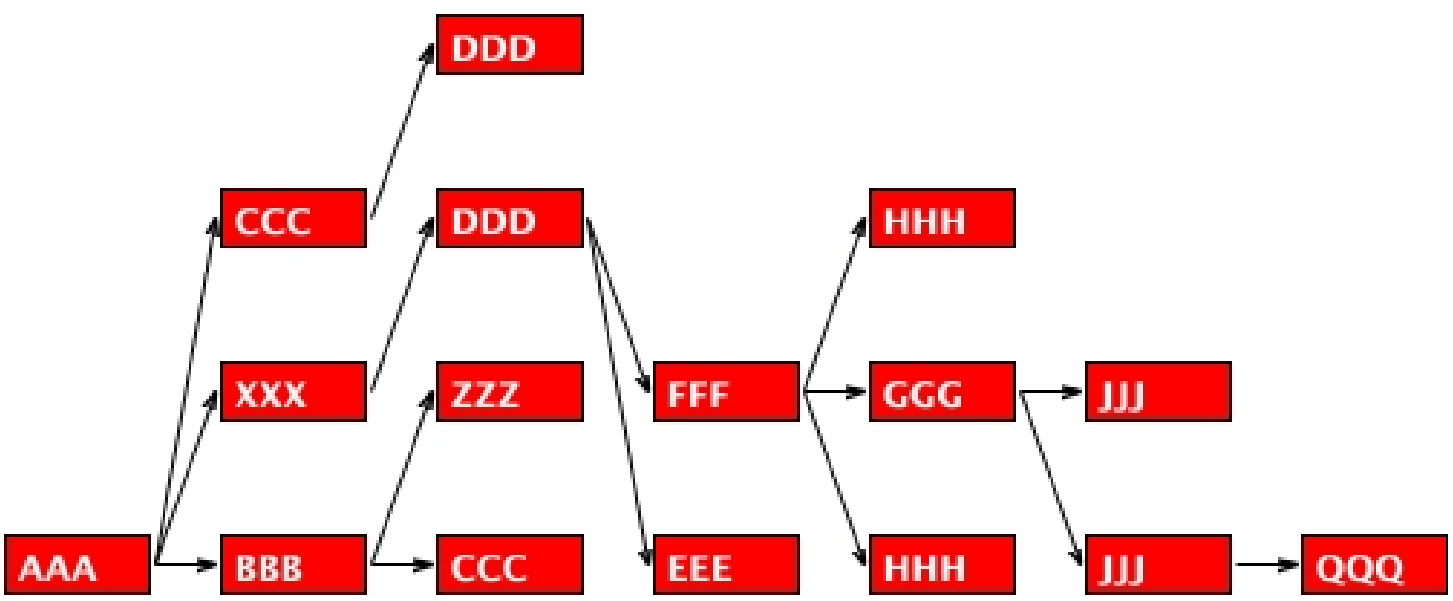
Xpath : recherche avec les axes

- //GGG/self::*



Xpath : recherche avec les axes

- //GGG/ancestor::* | //GGG/descendant::* | //GGG/following::* | //GGG/preceding::* | //GGG/self::*



Xpath : recherche avec les axes

- TD
  - enonce-TD2-XSL-Xpath