

Langage Xpath

Concepts fondateurs

Xpath : Sélection des nœuds

◆ Définition

- Xpath est le langage de parcours de l'arbre XML et de désignation des nœuds
- Xpath permet donc :
 - d'adresser le document en spécifiant un chemin de localisation
 - de savoir où on en est dans le document
 - Xpath connaît à tout moment le nœud contextuel :
 - i.e. la section du document d'où l'on part (référence)
 - Le raccourci du nœud contextuel de départ est le "."
 - d'indiquer une direction de recherche dans un document XML à partir d'un point du document
 - Vers un fils, un frère, un père, à gauche, à droite...

Sélection des nœuds

Vision de Xpath de l'arbre XML

◆ Notion de nœud

- Un document XML bien formé est associé, par la spécification Xpath, de manière non ambiguë à une structure arborescente
- Les nœuds de cette structure peuvent être de 7 types :
 - Le nœud racine
 - Les nœuds éléments (ou balises)
 - Les nœuds texte
 - Les nœuds attributs
 - Les nœuds commentaires
 - Les nœuds Processing Instructions
 - Les nœuds namespace

Sélection des nœuds

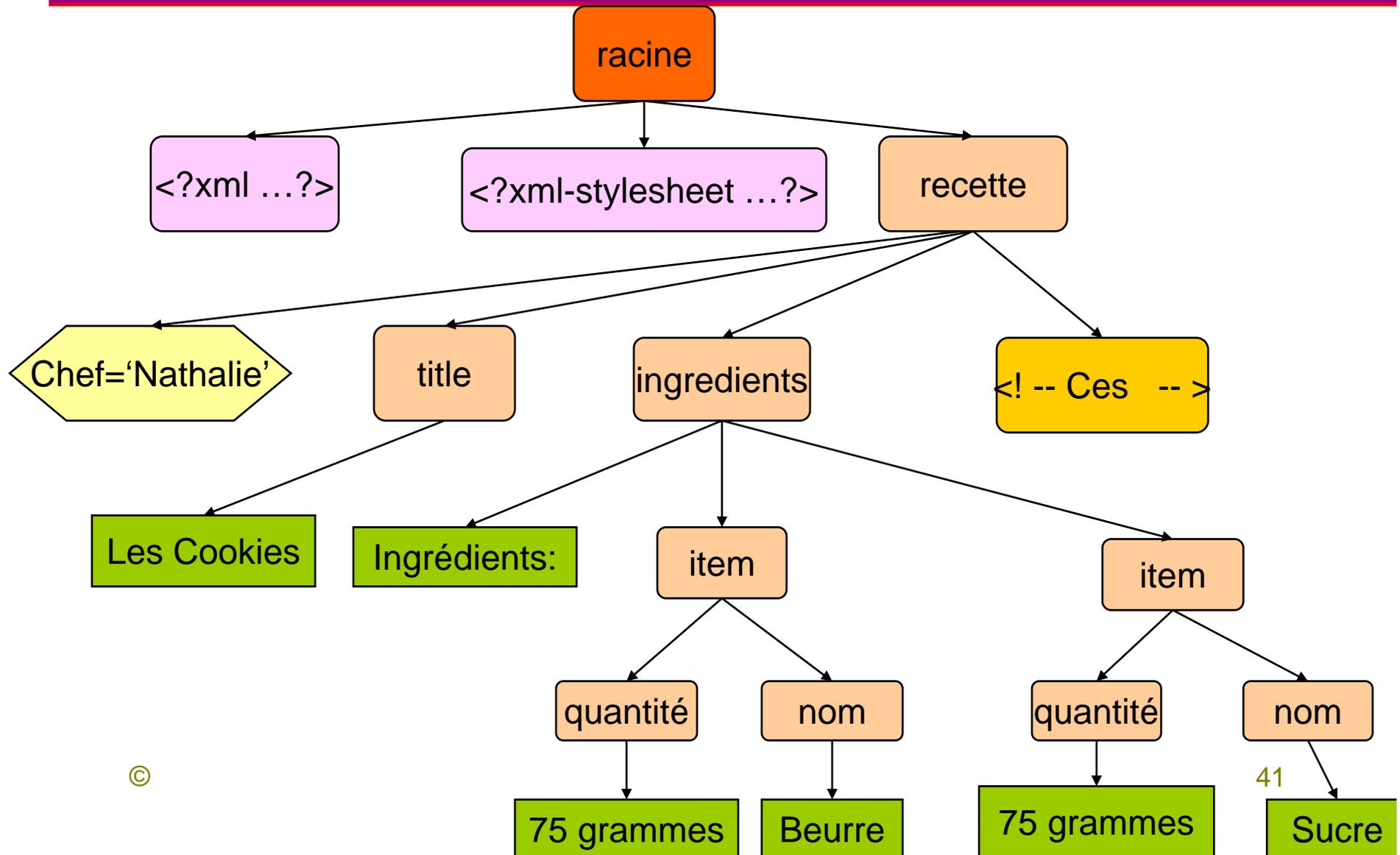
Vision de Xpath de l'arbre XML

◆ Exemple

```
<?xml version='version1.0' encoding='ISO-8859-1' ?>
<?xml-stylesheet type="text/css" href="recette.css" ?>
<recette chef = 'Nathalie'>
  <title>Les Cookies</title>
  <ingredients>Ingredients:
    <item>
      <quantite>75 grammes </quantite>
      <nom>Beurre</nom>
    </item>
    <item>
      <quantite>75 grammes </quantite>
      <nom>sucre</nom>
    </item>
  </ingredients>
  <!-- Ces cookies sont délicieux! -->
</recette>
```

Sélection des nœuds

Vision de Xpath de l'arbre XML



5 Sélection des nœuds

Les expressions Xpath

- ◆ Elles peuvent être de 4 types :
 - Groupe de nœuds
 - Les expressions Xpath identifiant un groupe de nœuds sont des 'Chemins' (Location Path)
 - Les Chemins sont intensivement utilisés dans XSLT
 - Number
 - String
 - Boolean
- ◆ Les 3 dernières (expressions Xpath *autres que les Chemins*) sont utilisées :
 - Pour exprimer des conditions
 - Pour introduire de nouvelles valeurs
 - avec `<xsl:value-of>` par exemple :
`<xsl:value-of select='string(123)' />`

Le langage Xpath

Identification de chemin

◆ Les niveaux

- Un chemin est constitué de différents niveaux séparés les uns des autres par le caractère /

niveau1/niveau2

◆ Évaluation d'un chemin

- Chaque niveau permet la sélection d'un groupe de nœuds qui est évalué relativement à un nœud courant
- Lorsque le chemin commence par /, le nœud courant est le nœud racine (*chemin absolu*)
- Le contexte d'évaluation d'un niveau est constitué de :
 - Un nœud courant
 - Un ensemble de nœuds appelé contexte
 - La position du nœud courant dans le contexte
 - La taille du contexte, les variables globales, les fonctions disponibles, ...

Le langage Xpath

Constitution des niveaux

- ◆ Un niveau permet, *à partir du nœud courant*, la sélection d'un groupe de nœuds de trois manières différentes, en spécifiant :
 - un **axe** de parcours du document (*à partir du nœud courant*)
 - le(s) **type(s) de nœud** du document à sélectionner
 - un **prédicat** devant être satisfait par les nœuds sélectionnés
- ◆ Constitution d'un niveau

axe :: type_de_nœud [**prédicat**]

ex: child :: date [year='2000']

Le langage Xpath

Les axes

- ◆ Spécifient des zones de l'arborescence du document, dans lesquelles les nœuds sélectionnés doivent se trouver
- ◆ Axes abrégés
 - child
 - les fils du nœud courant, de type autre que attribut ou namespace
 - C'est l'axe par défaut
 - site : sélectionne les fils de type « site » du nœud courant
équivalent à : `child::site`
 - self
 - le nœud courant lui-même
 - .
 - parent
 - le père du nœud courant
 - ..

Le langage Xpath

Les axes

◆ Axes abrégés (suite)

- descendant-or self
 - descendants du nœud courant y-compris lui-même
 - `..`
- attribute
 - Les attributs du nœud courant
 - `@`

Le langage Xpath

Les axes

◆ Axes sans abréviation

- **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

Le langage Xpath

Sélection des différents types de nœuds

◆ Nœud de type élément :

- par le nom
 - sélectionne tous les nœuds de type élément ayant ce nom

titre

./titre

following-sibling::titre

◆ Nœud de type attribut :

- par le nom de l'attribut

img/@href

◆ Nœuds de type commentaires, texte et processing instruction

- **comment()** **text()** **processing-instruction()**

recette//comment()

titre/text()

Le langage Xpath

Sélection des différents types de noeuds

◆ Les jokers

* remplace toute chaîne de caractères

*	les éléments d'un nom quelconque
@*	les attributs d'un nom quelconque
xsl:*	les éléments de l'espace de nom xsl

◆ Les choix multiples : **opérateur** |

prénom | nom

@id | @idref

* | @* | text()

Le langage Xpath

Expression de prédicats

◆ Prédicat = [expression]

- Avec : expression =

exp1 or exp2

exp1 and exp2

operande1 = != < <= > >= operande2

operande

◆ Les opérandes peuvent être de type :

- Expression primaire
 - Une chaîne de caractères entre quotes
 - Un nombre
 - Une référence à une variable
 - Un appel de fonction
- Expression de chemin
- Expression arithmétique (+ - x div mod)

Le langage Xpath

Evaluation de l'expression E dans [E]

◆ E de type booléen :

- retourne la valeur de E

`//name[prenom='julien' or prenom='robert']`

`Personne[@naissance <= 1976]`

`date[meeting/period/from < 800]`

◆ E de type numérique :

- retourne vrai si val(E)=position de l'élément courant

`Personne[2]`

◆ E de type ensemble de nœuds :

- retourne vrai si E est non vide

`//name[surnom]`

◆ E de type chaîne :

- retourne faux si val(E)="" et vrai sinon

Le langage Xpath

◆ Fonctions sur les nœuds

- `position()` : position du nœud courant dans le contexte
- `last()` : taille du contexte
- `count(EnsNoeuds)` : nombre de nœuds de EnsNoeuds
- `id(str)` : le nœud dont l'attribut ID est égal à str

◆ Fonctions de chaînes de caractères

- `concat(str1,str2)`
- `starts-with(str1,startstr)`
- `contains(str1,str2)`
- `substring-before(str1,str2)`
- `substring-after(str1,str2)`
- `string-length(str)`

Le langage Xpath

◆ Fonctions numériques

- floor(nb) : arrondi vers l'entier supérieur
- ceiling(nb) : arrondi vers l'entier inférieur
- round(nb) : arrondi

◆ Fonctions de conversion

- string(obj) : conversion d'obj en chaîne
- number(obj) : conversion d'obj en entier
- boolean(obj)

Xpath : recherche par le nom

Utiliser l'interface XSLT sur bibliotheque1.xml : /bibliotheque

- ◆ La base de la syntaxe Xpath est semblable à celle de l'adressage du système de fichiers
 - Si le chemin commence par '/', alors il représente un chemin absolu vers l'élément requis
 - **AAA** : Sélectionne l'élément racine AAA

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

```
<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/>
    <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

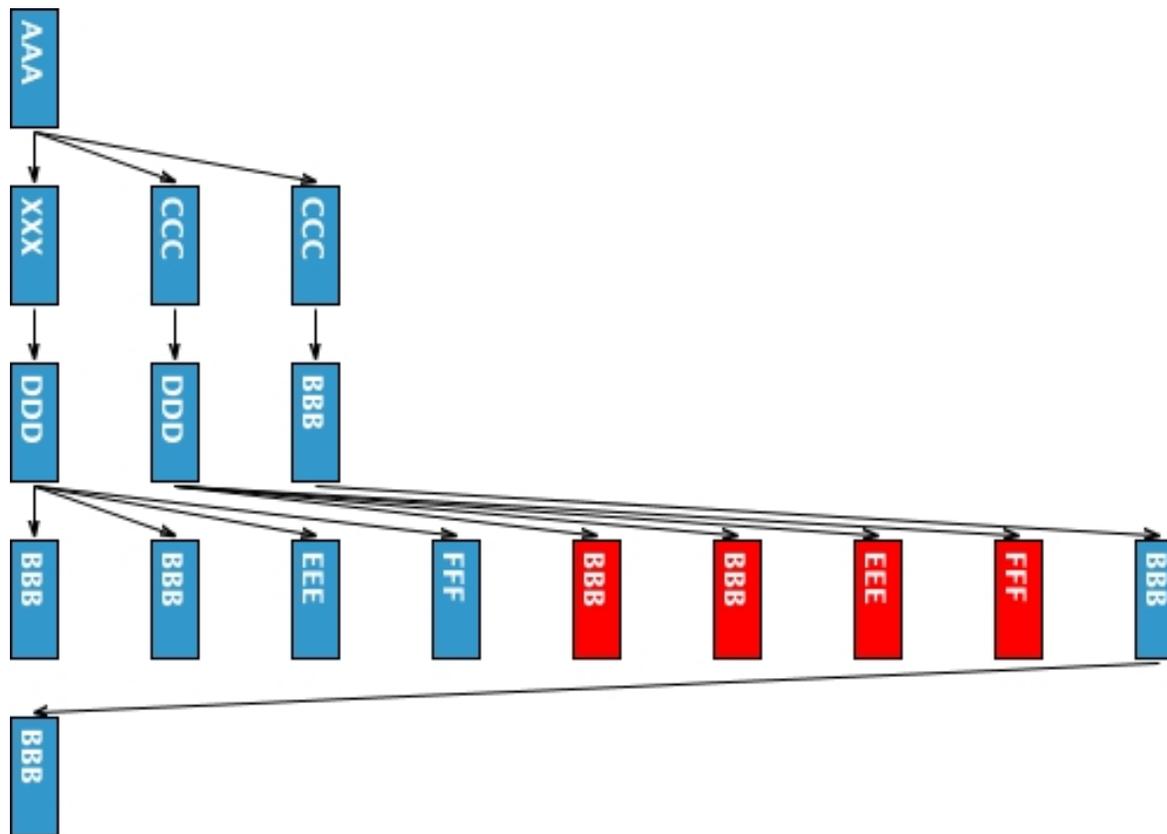
```
<AAA>
  <BBB/>
  <CCC/>
  <BBB/>
  <DDD>
    <BBB/>
  </DDD>
  <CCC>
    <DDD>
      <BBB/>
      <BBB/>
    </DDD>
  </CCC>
</AAA>
```

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`

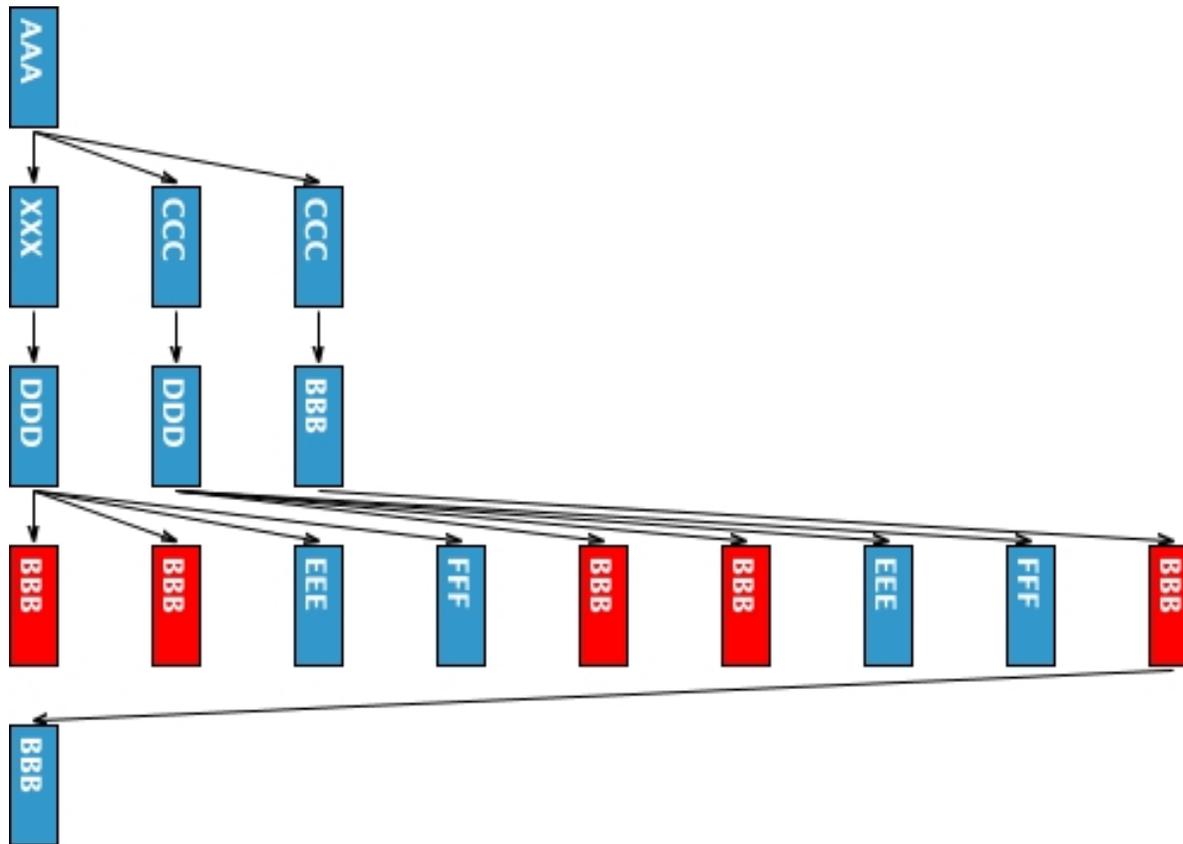


Xpath : recherche par le nom

`/*/*/titre`

◆ `/*/*/*/BBB`

- Sélectionne tous les éléments BBB qui ont trois ancêtres



Xpath : recherche par nom[prédicat]

`/bibliotheque/livre[2]`

Une expression **entre crochets** peut spécifier plus précisément un élément. 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 : recherche par l'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 : recherche par l'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 : recherche par l'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 : recherche par l'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 : recherche par l'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 : recherche par les fonctions

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>
```

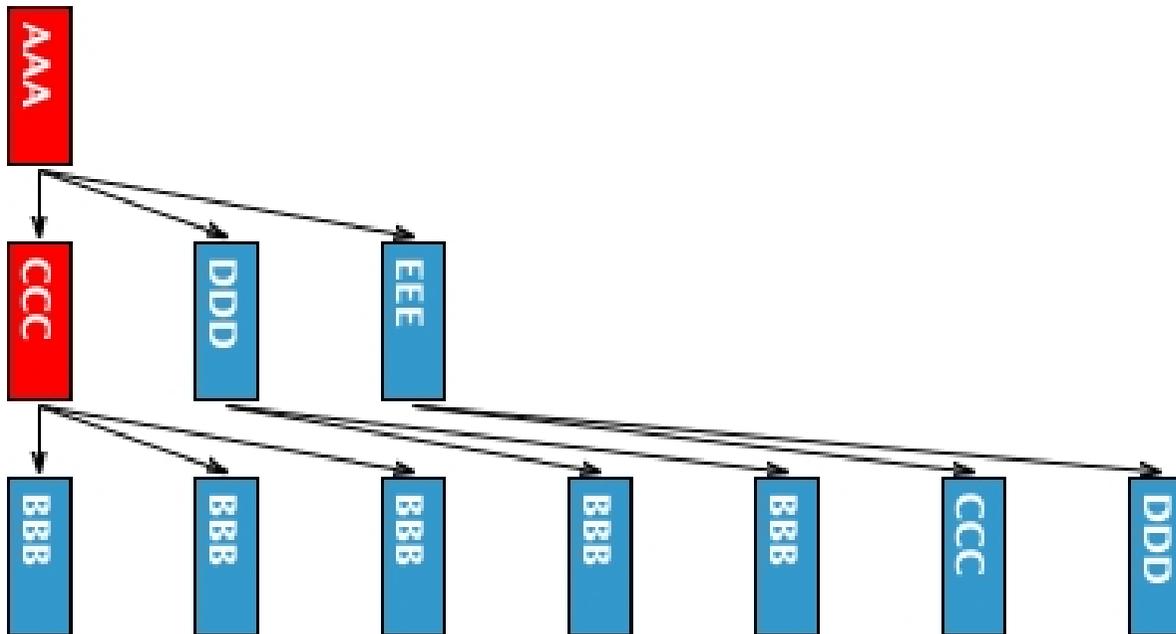
Xpath : recherche par les fonctions

`//*[count(*)=3]`

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

◆ `//*[count(*)=3]`

- Sélectionne les éléments ayant trois enfants

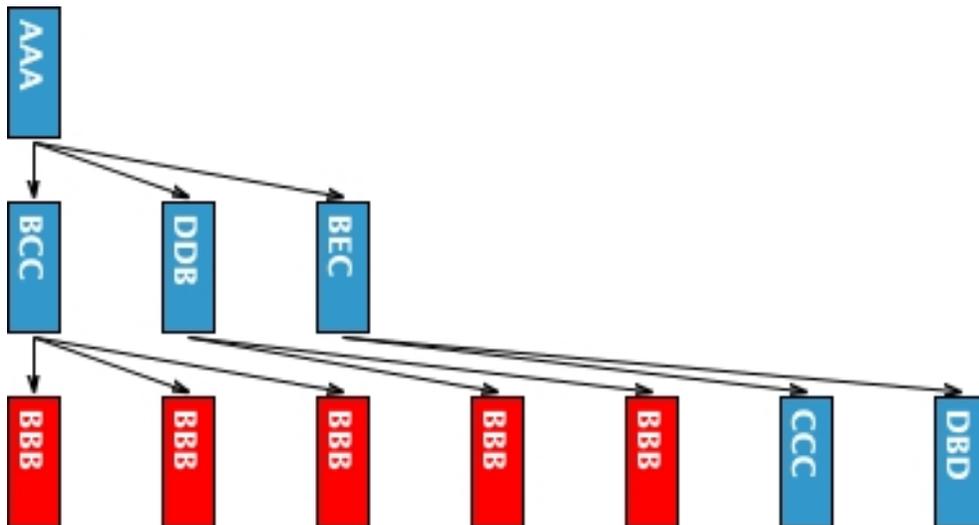


Xpath : recherche par les fonctions

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

◆ `//*[name()='BBB']`

La fonction `name()` retourne le nom de l'élément



Xpath : recherche par les fonctions

`//*[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 : recherche par les fonctions

`//*[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 : recherche par les fonctions

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

La fonction `string-length` retourne le nombre de caractères dans une chaîne. Vous devez utiliser `<` comme substitutif de `<` et `>` 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 : recherche par les fonctions

- ◆ `//*[string-length(name()) > 3]`

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

Xpath : recherche par les fonctions

//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>
```

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>
```

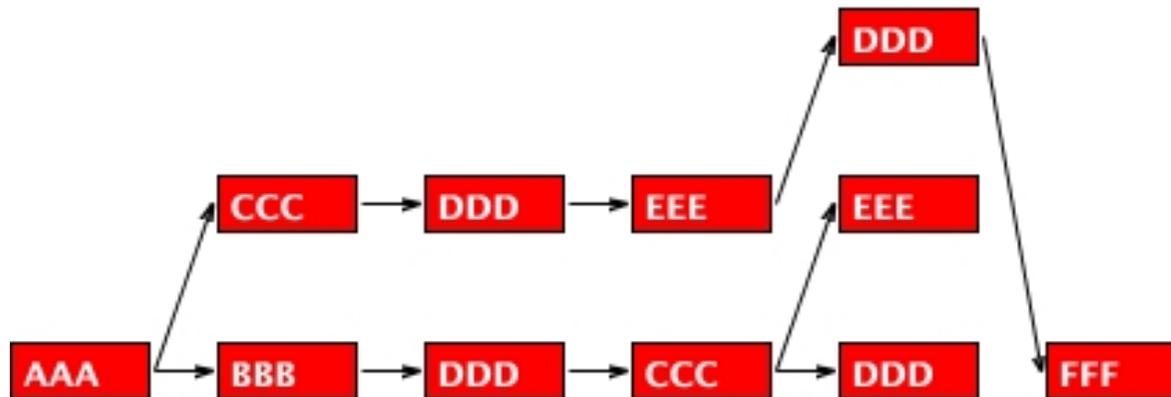
Xpath : recherche avec les axes

`/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::*`

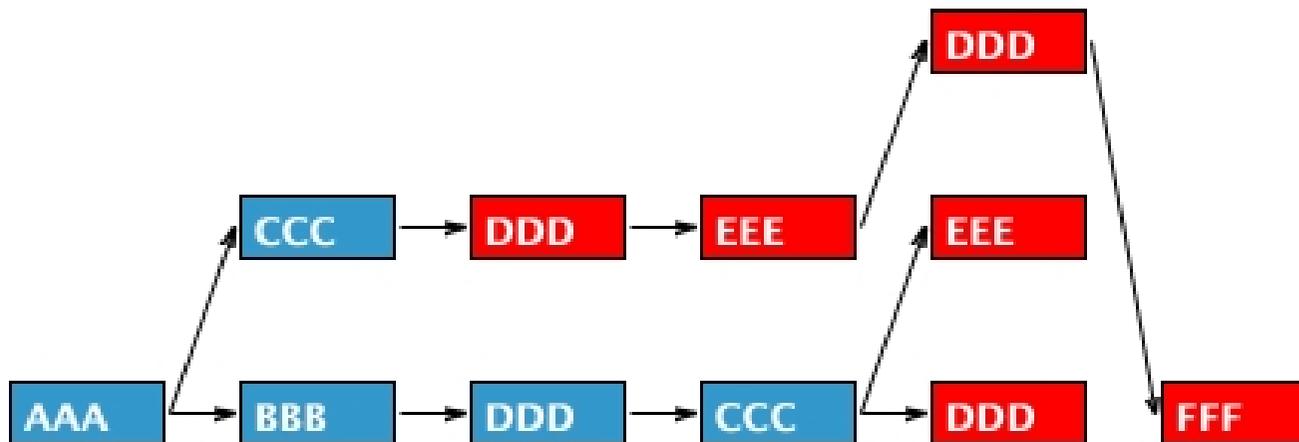
- Sélectionne tous les descendants de l'élément racine et donc tous les éléments



Xpath : recherche avec les axes

//bibliotheque/descendant::titre

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

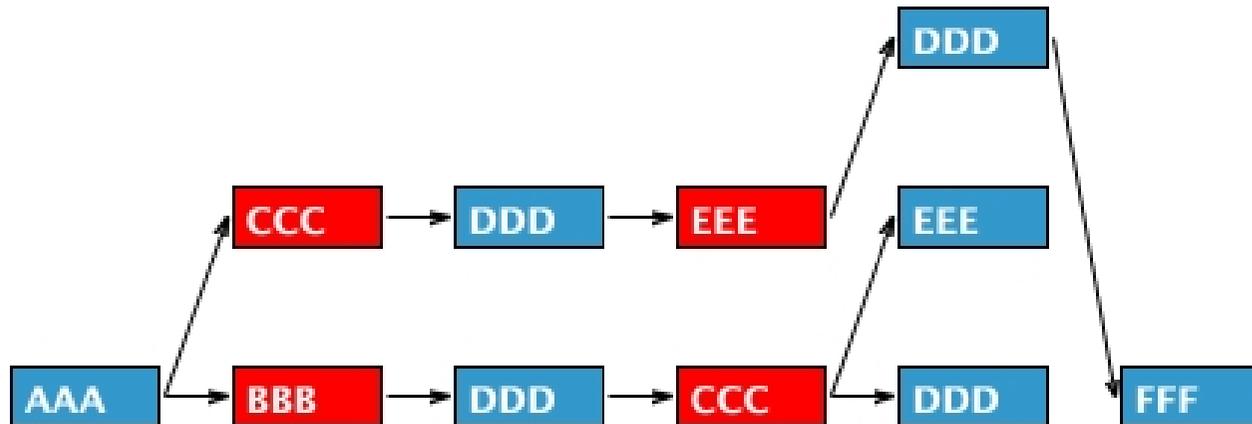


Xpath : recherche avec les axes

`//livre/parent::*`

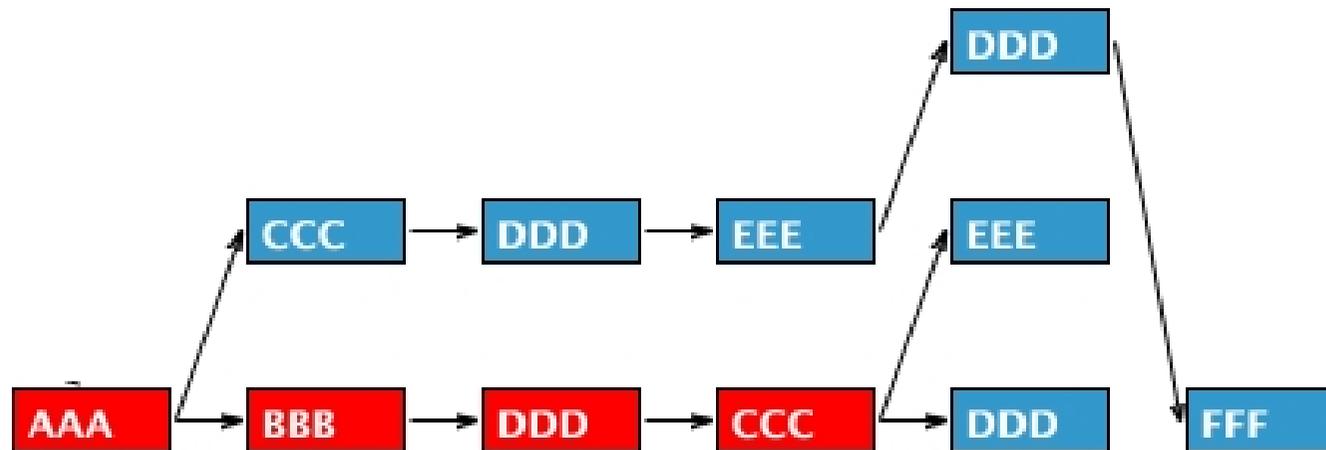
L'axe "parent" contient le parent du nœud contextuel s'il en a un

- `//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

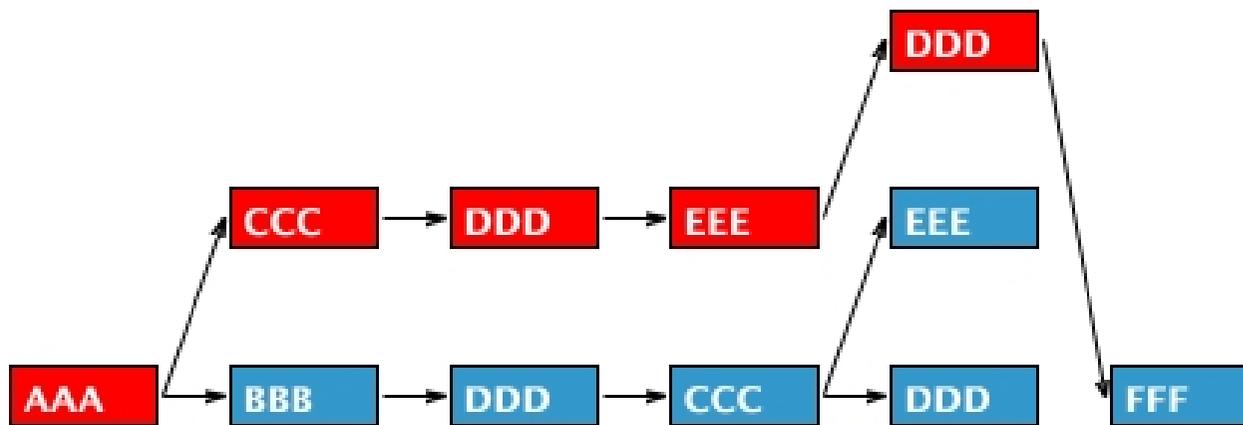


Xpath : recherche avec les axes

`//auteur/ancestor::*`

◆ `//FFF/ancestor::*`

- Sélectionne tous les ancêtres de l'élément FFF

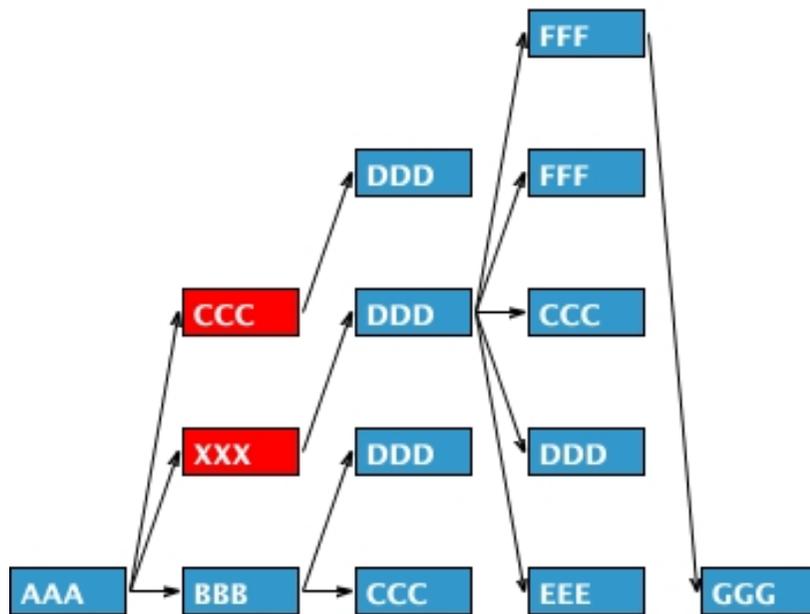


Xpath : recherche avec les axes

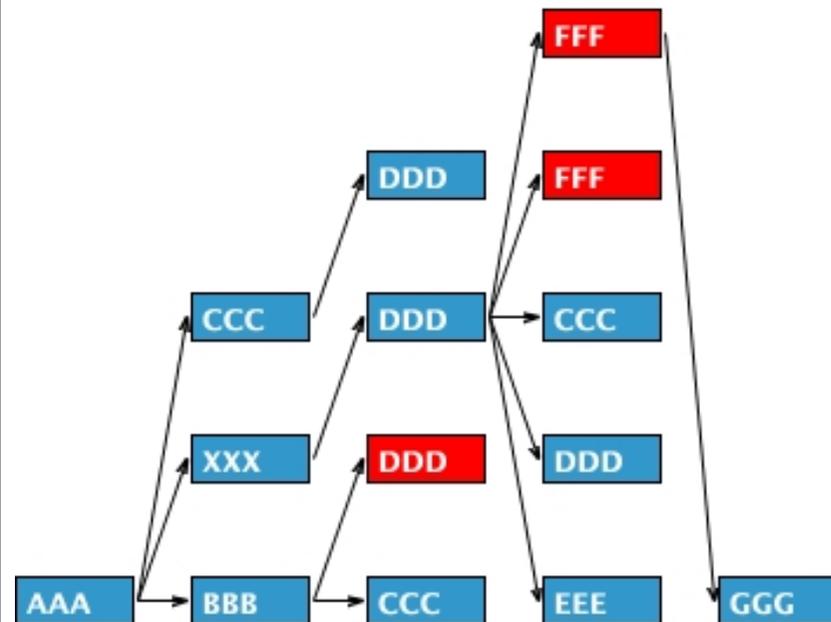
`/bibliotheque/livre/titre/following-sibling::*`

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

`/AAA/BBB/following-sibling::*`



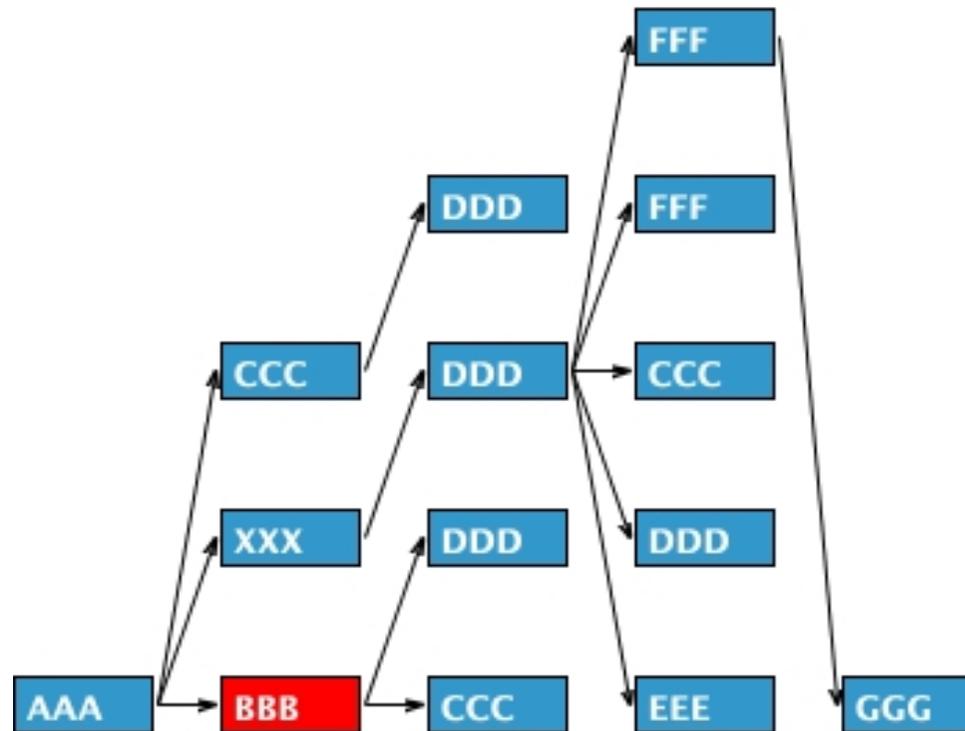
`//CCC/following-sibling::*`



Xpath : recherche avec les axes

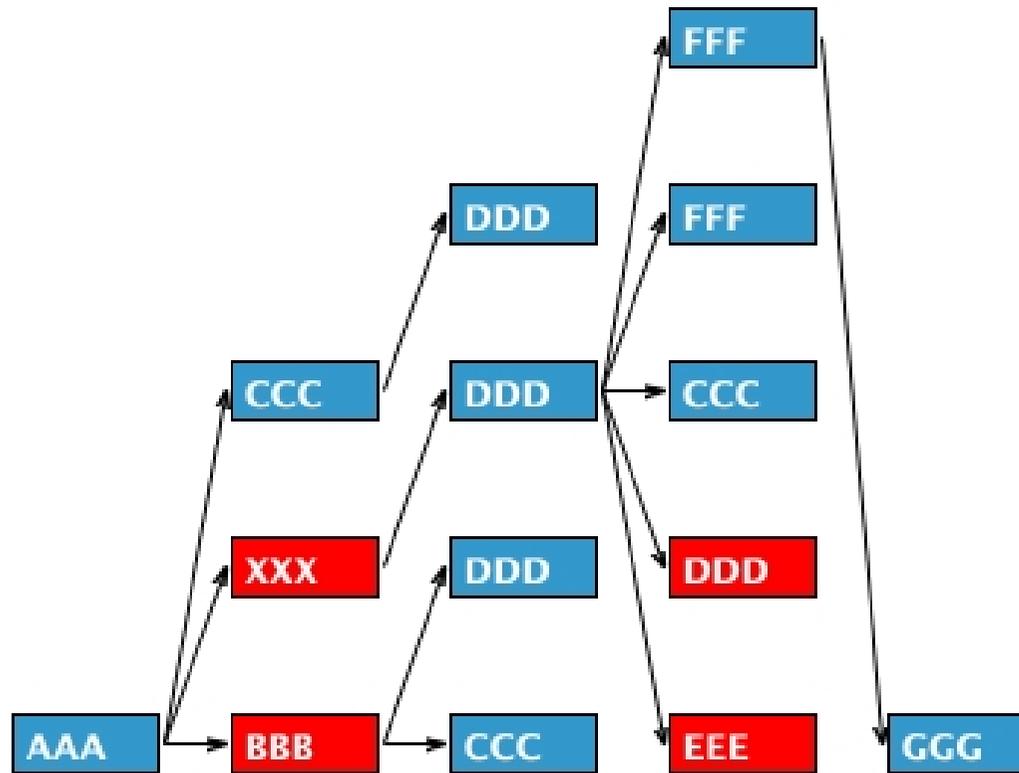
L'axe '**preceding-sibling**' contient tous les frères prédécesseurs du nœud contextuel; si le nœud contextuel est un attribut ou un espace de noms, la cible précédente est vide

◆ `/AAA/XXX/preceding-sibling::*`



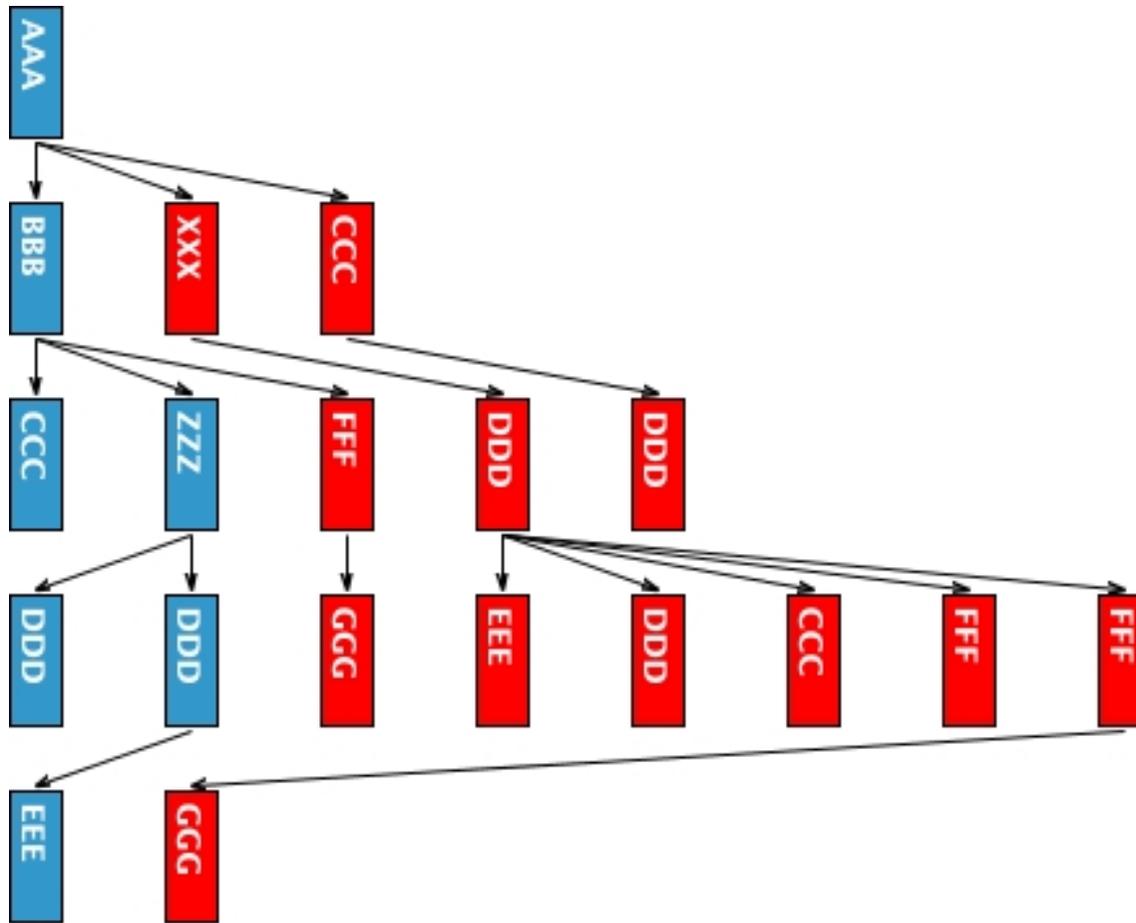
Xpath : recherche avec les axes

◆ //CCC/preceding-sibling::*



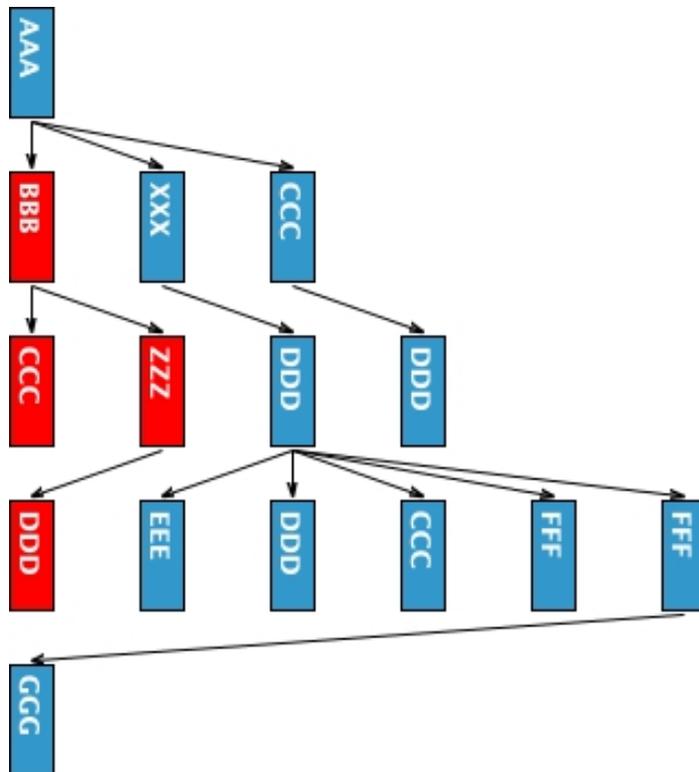
Xpath : recherche avec les axes

◆ //ZZZ/following::*



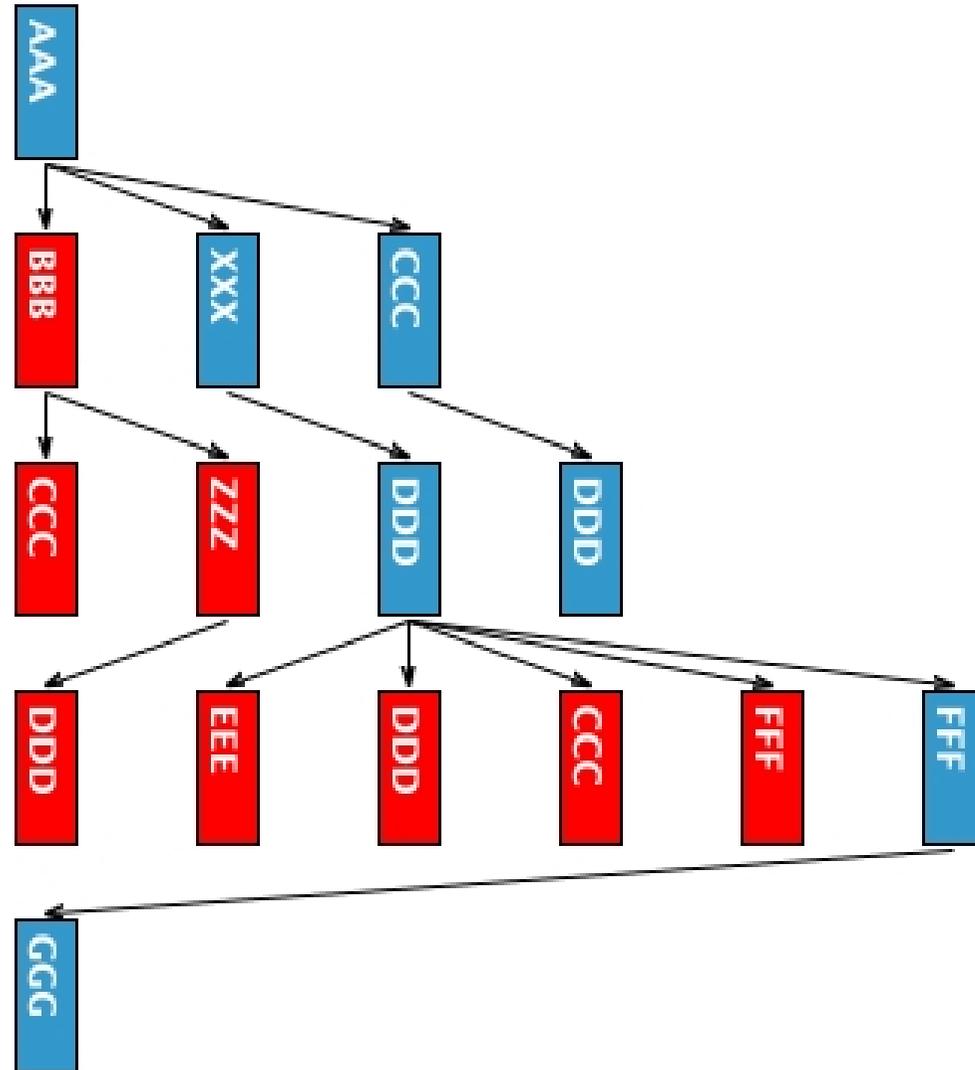
Xpath : recherche avec les axes

- ◆ 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

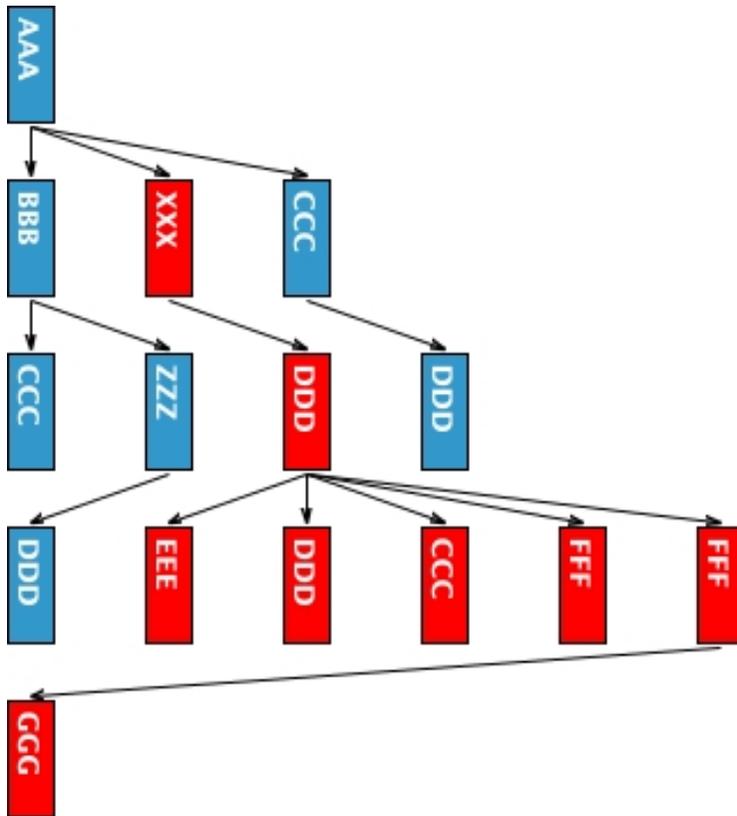
◆ //GGG/preceding::*



Xpath : recherche avec les axes

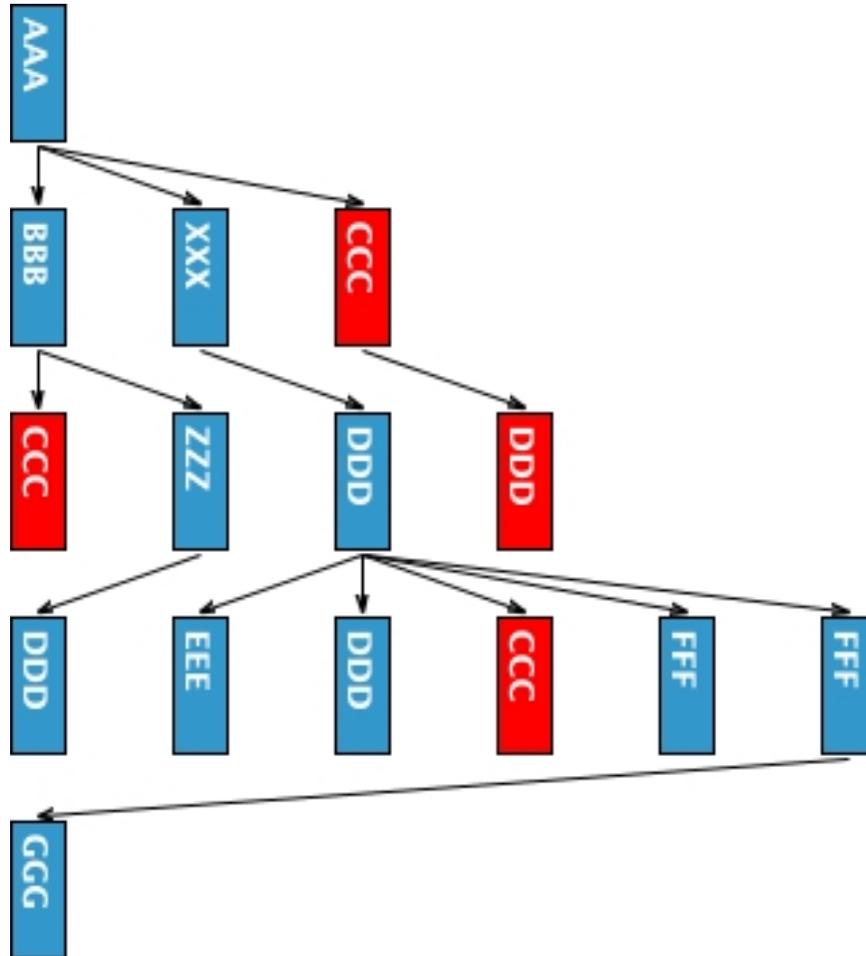
`/bibliotheque/livre/descendant-or-self::*`

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



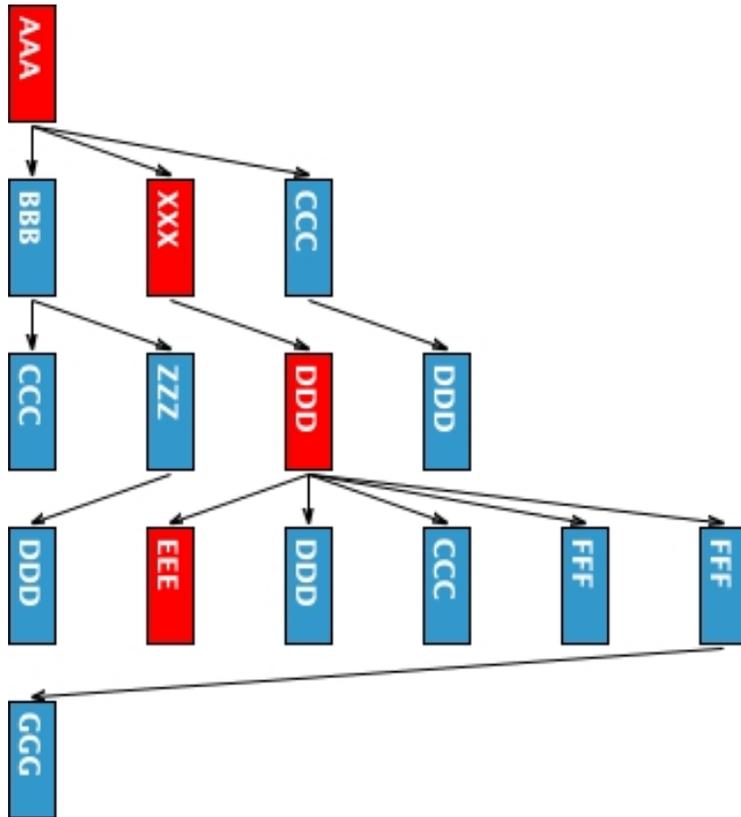
Xpath : recherche avec les axes

◆ `//CCC/descendant-or-self::*`



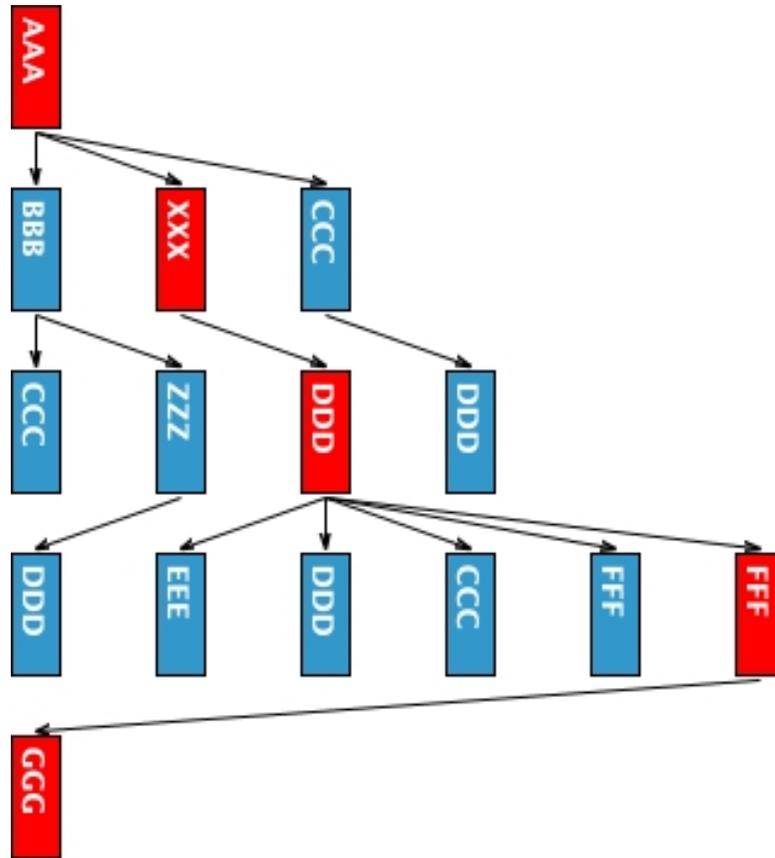
Xpath : recherche avec les axes

- ◆ 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

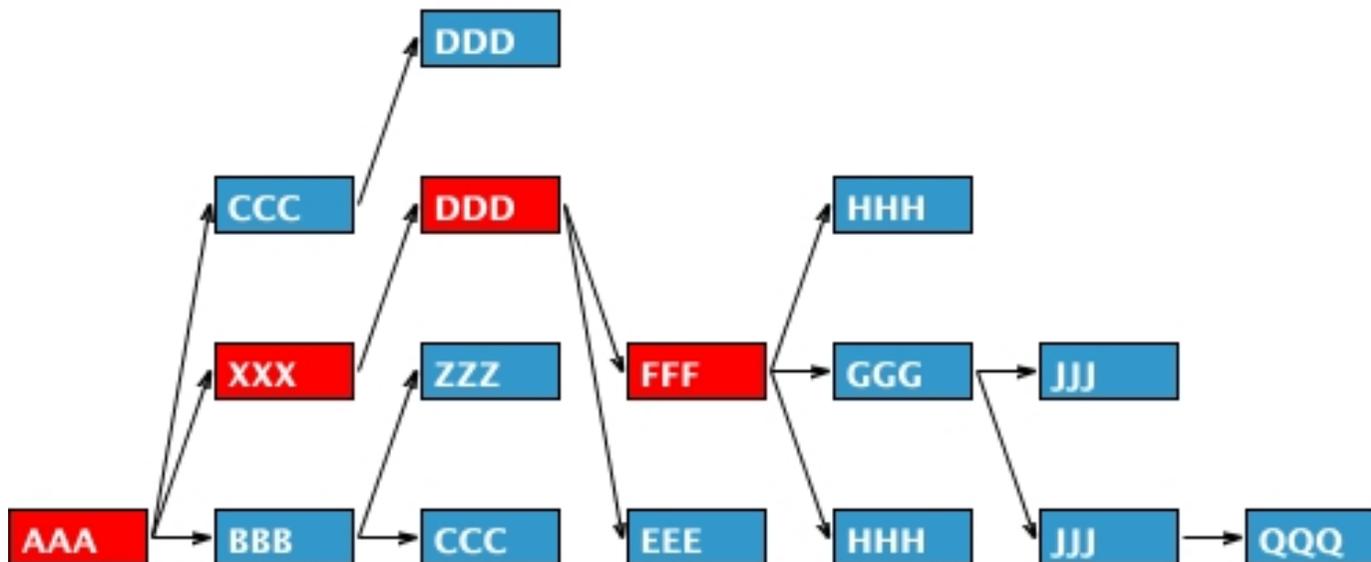
◆ //GGG/ancestor-or-self::*



Xpath : recherche avec les axes

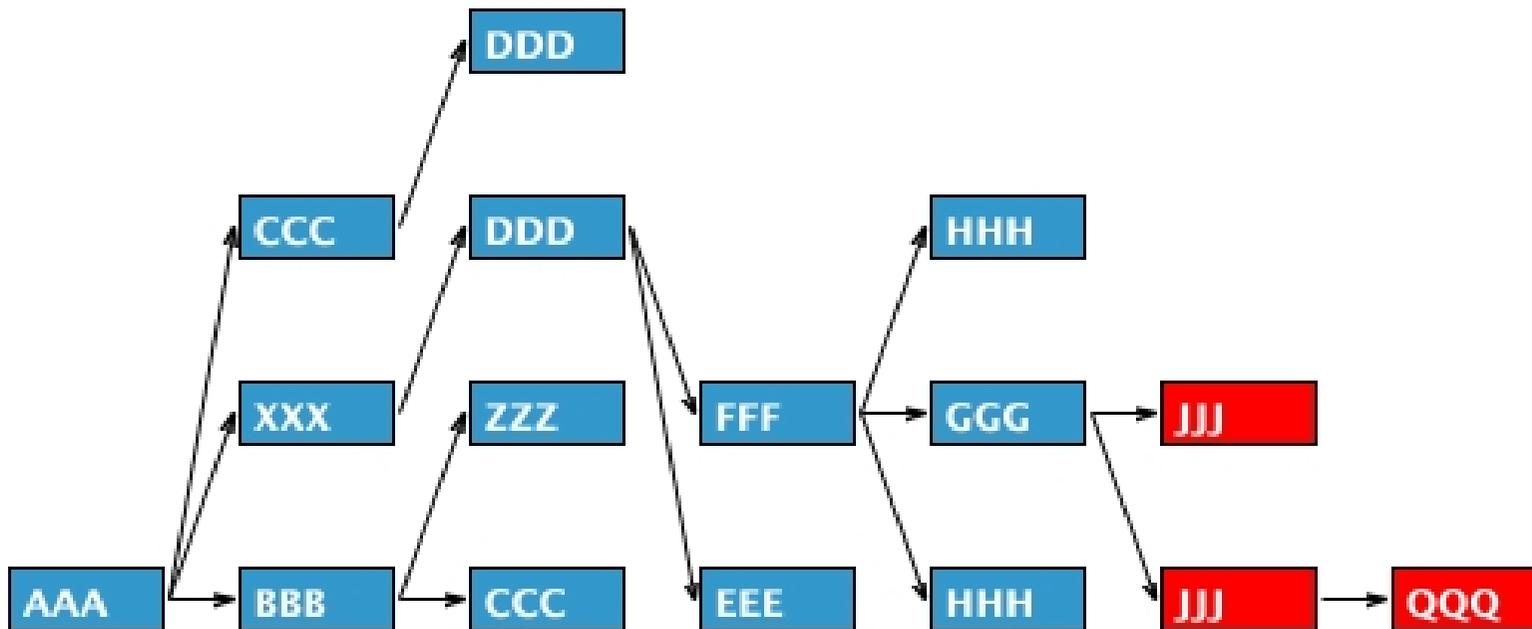
`//livre/ancestor::*`

- ◆ 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::*`



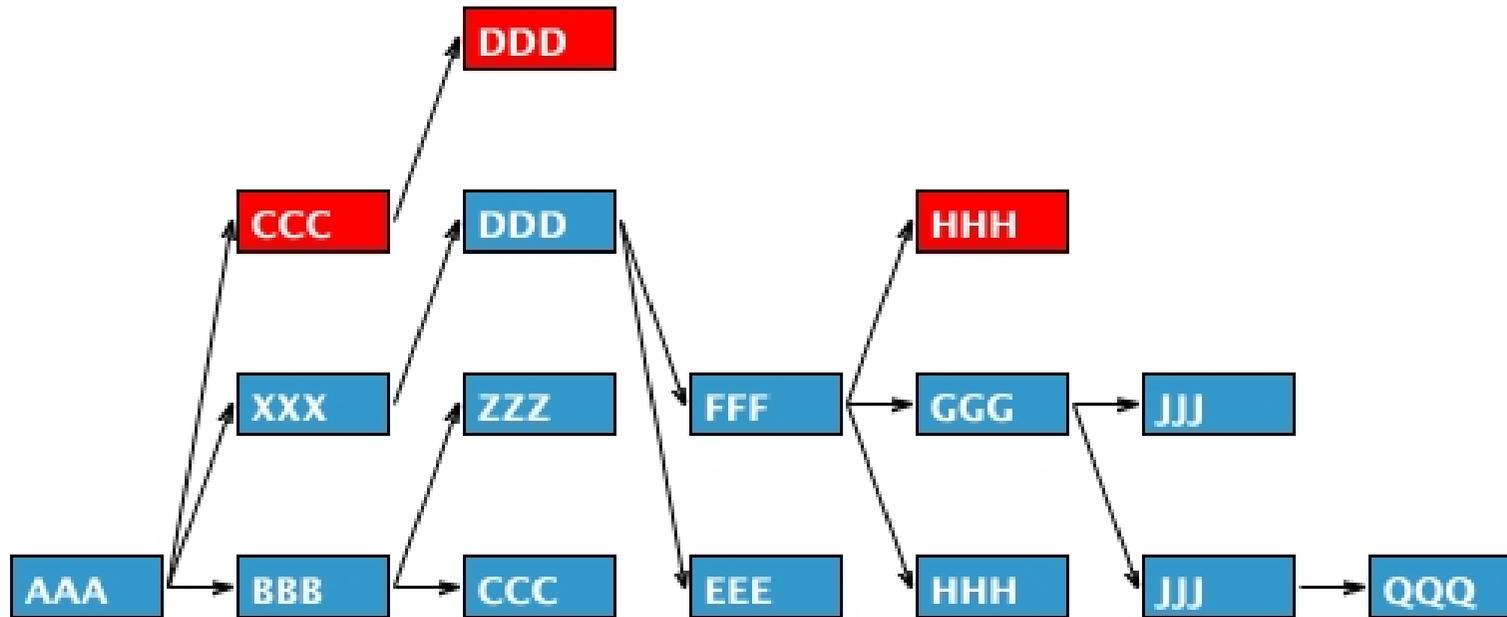
Xpath : recherche avec les axes

◆ //GGG/descendant::*



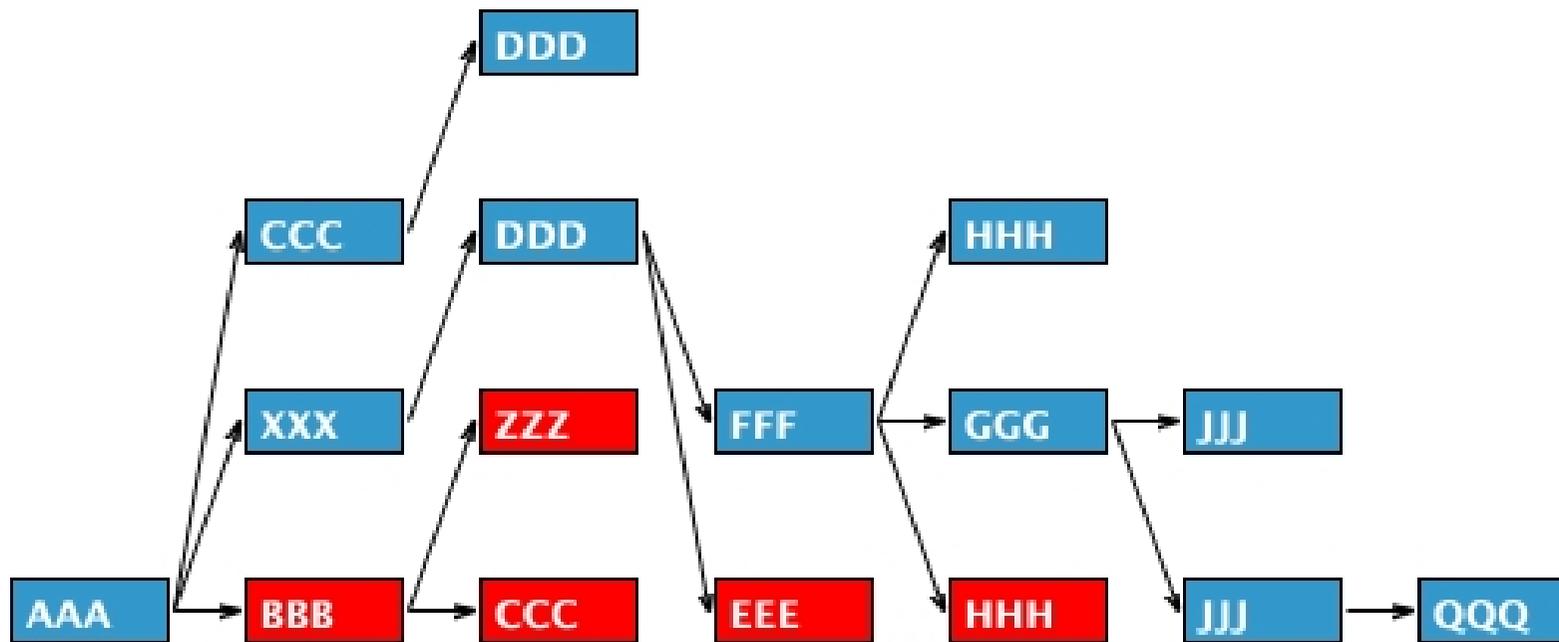
Xpath : recherche avec les axes

◆ //GGG/following::*



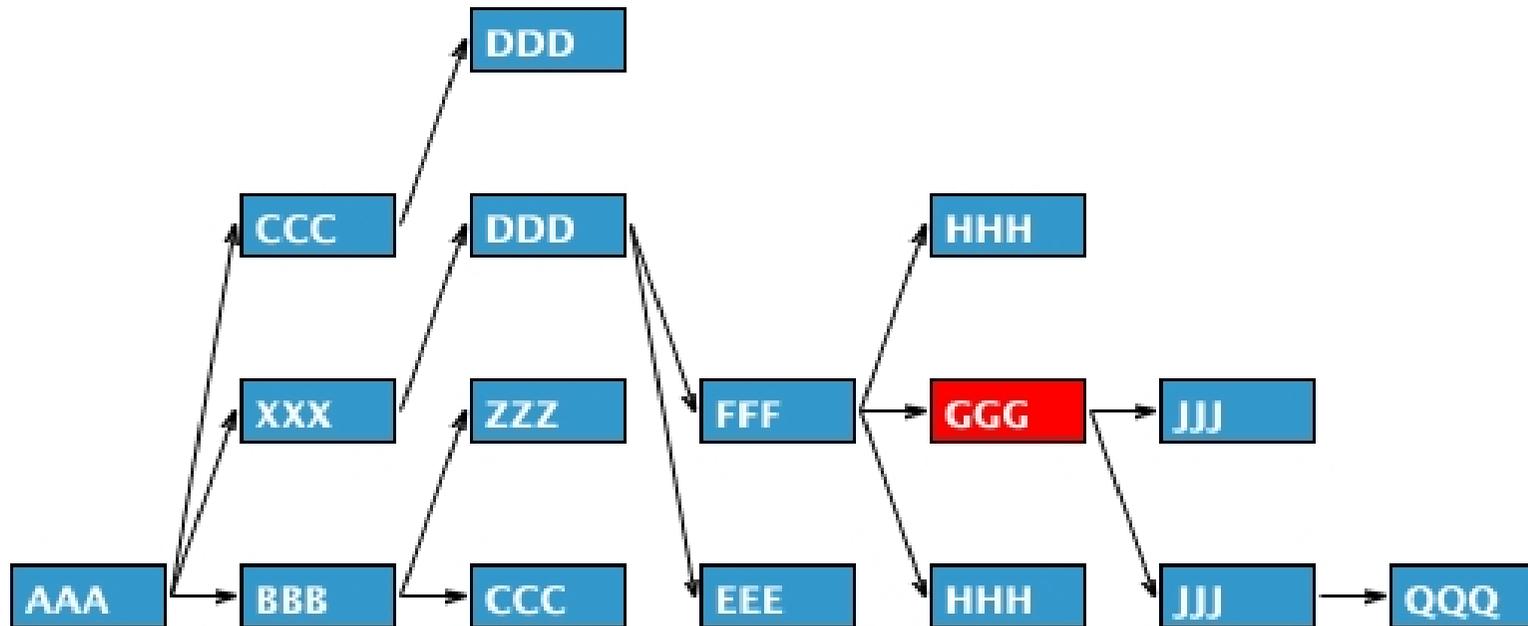
Xpath : recherche avec les axes

◆ //GGG/preceding::*



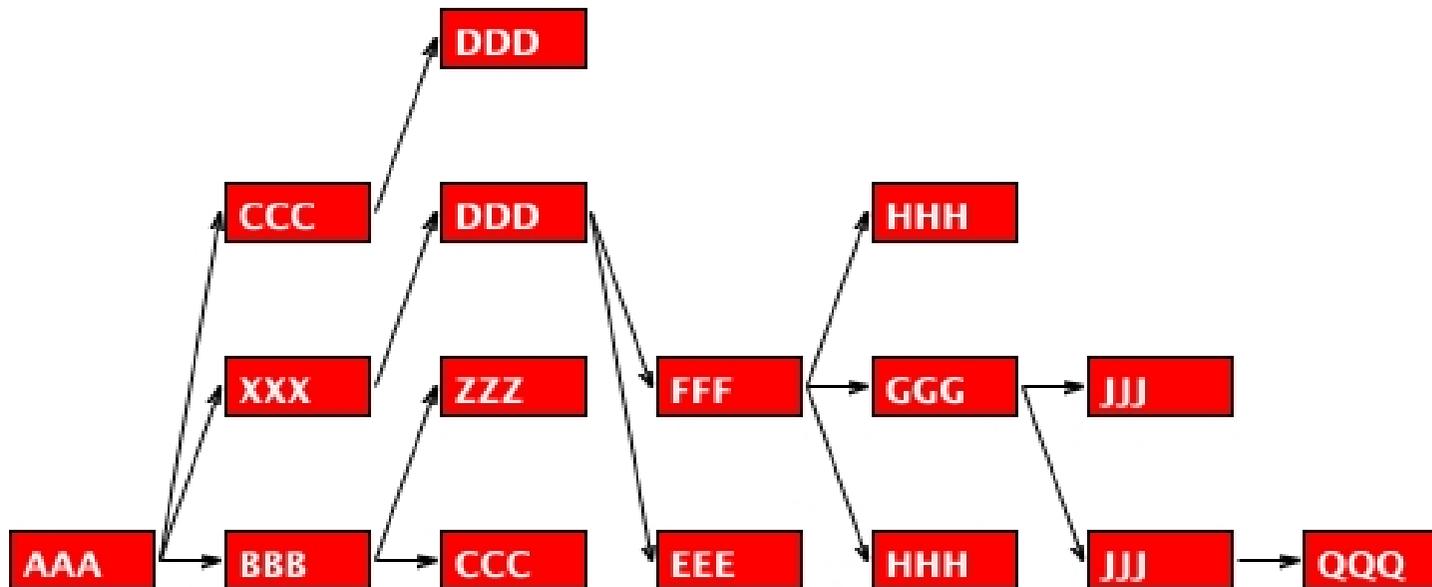
Xpath : recherche avec les axes

◆ //GGG/self::*



Xpath : recherche avec les axes

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



Xpath

- ◆ `//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/>
  <BBB/>
  <CCC/>
  <CCC/>
  <CCC/>
</AAA>
```

Xpath : résumé des fonctions

- ◆ Fonctions de sélection nodale : voir : **predicat**
 - **element[n]**
 - sélectionne le $n^{\text{ième}}$ élément *element* dans le nœud courant
 - **element[elt]**
 - sélectionne dans le nœud 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 : résumé des fonctions

◆ Fonctions nodales (suite) : voir `count/planete`

- Comptage : `count()`
 - `count(ensemble_de_noeuds)` retourne le nombre de noeuds dans `ensemble-de-noeuds`
 - Exemple :
 - `count(planete)` : renverra la valeur 9
- Nœud courant : `current ()`
 - retourne le noeud courant
- Position d'un nœud : `position()` et `last()`
 - Ces fonctions permettent de positionner un nœud par rapport à ses frères
 - Exemple :
 - `planete[position()=3]/nom` : nom de la 3ème planète (terre)
 - `planete[position()=last()]/nom` : nom de la dernière planète (pluton)

Xpath : résumé des fonctions

◆ Fonctions sur chaînes de caractères

- Conact(chaine1,chaine2,...)
 - concatène toutes les chaînes
 - Exemple :
 - `<xsl:value-of select concat('Le livre dont le titre est', livre/titre, ' a été écrit par ', livre/auteur)/>`
 - affiche la chaîne "Le livre dont le titre est Les Misérables a été écrit par Victor Hugo".
- Test de présence d'une sous-chaîne
 - starts-with(chaine1, chaine2)
 - renvoie la valeur true si chaine1 commence par la chaine2
 - contains(chaine1, chaine2)
 - renvoie true si chaine1 contient chaine2

Xpath : résumé des fonctions

◆ Fonctions sur chaînes de caractères (suite)

- Extraction de sous-chaîne
 - **substring(chaine1, decalage, longueur)**
 - retourne une sous-chaîne de chaine1 contenant longueur caractères et commençant à decalage
 - **substring-after(chaine1, chaine2)**
 - retourne la sous-chaîne de chaine1 qui suit la première occurrence de chaine2
 - **substring-before(chaine1, chaine2)**
 - retourne la sous-chaîne de chaine1 qui précède la première occurrence de chaine2

Xpath : résumé des fonctions

◆ XPath sur chaînes de caractères (suite)

- Transformation des caractères d'une chaîne
 - **translate(chaine1, chaine2, chaine3)**
 - retourne chaine1 après y avoir remplacé chaque occurrence des caractères de chaine2 par le caractère correspondant (c'est-à-dire occupant la même position) de chaine3
 - Par exemple, pour transformer une lettre quelconque et la mettre en majuscule, on utilise l'appel suivant de la fonction :
 - `translate(chaine, "abcdefghijklmnopqrstuvwxyz", "ABCDEFGHIJKLMNOPQRSTUVWXYZ")`.
- Suppression des espaces en surnombre
 - **normalize-space(chaine)**
 - retourne la chaîne de caractères en enlevant les espaces répétitifs
- **Longueur d'une chaîne:**
 - **string-length(chaine) :**
 - retourne la longueur de chaine

Xpath : résumé des fonctions

◆ Fonctions numériques

- Opérateurs disponibles
 - + pour l'addition ;
 - - pour la soustraction ;
 - * pour la multiplication ;
 - div pour la division ;
 - nombre1 mod nombre2 : reste de la division
- Fonctions de manipulation
 - ceiling(nombre) retourne le plus petit entier \geq nombre
 - floor(nombre) retourne le plus grand entier \leq nombre
 - round(nombre) arrondit nombre à l'entier le plus proche
 - sum(nombre1, nombre2, ...) retourne la somme
 - Exemple : sum(//prix) calcule la somme des contenus de tous les éléments prix du document XML

Xpath : résumé des fonctions

◆ Fonctions booléennes : voir : boolean

- element-available(chaîne)
 - retourne *true* ssi le nom étendu est le nom d'une instruction
- false()
 - retourne *false*
- function-available()
 - *true* si la fonction est disponible dans la librairie de fonctions
- lang(chaîne)
 - *true* si l'attribut *xml:lang* du noeud contextuel est le même que l'argument
- not(valeur)
 - retourne le contraire de la valeur booléenne passée en argument
- true()
 - retourne *true*

Xpath : résumé des fonctions

◆ Fonctions booléennes (suite)

- boolean(objet) : convertit l'argument en valeur booléenne
 - Le comportement dépend du type XPath de objet
 - Si objet = nombre non nul
 - Retourne true
 - si objet = 0
 - Retourne false ;
 - Si objet = chaîne non vide
 - true ; sinon, le résultat est false
 - Si objet = type booléen
 - Retourne true
 - Si l'objet = ensemble de nœuds
 - true sauf si l'ensemble de nœuds est vide (dans ce cas, la valeur est false)