

XML Path Language

## INTRODUCTION À XPATH

## Chemin de localisation XPath

- Chaîne de caractères qui identifie une partie d'un document XML (un sous-arbre)  
Identifie *tous* les nœuds correspondants
- Filtrage d'éléments sur la base de **patrons**

## Chemin de localisation XPath

### Concaténation

- d'un terme de localisation absolue:
  - sommet de l'arbre du document, père de l'élément racine : /
  - pseudo-fonction `id()`
- d'une suite de termes de localisation relative

```
/personnel/service[2]/emploi/nom
id('marketing')/emploi/nom
/personnel/service/emploi[titre="Président"]/nom
```

## Exemple

```
<personnel>
  <service id="ventes">
    ...
  </service>
  <service id="marketing">
    <emploi>
      <nom>Sarah Bellum</nom>
      <titre>Président</titre>
    </emploi>
    <groupe>
      ...
    </groupe>
  </service>
  ...
</personnel>
```

```
/personnel/service[2]/emploi/nom
id('marketing')/emploi/nom
/personnel/service/emploi[titre="Président"]/nom
```

## Cheminement et sélection des nœuds

- Définition d'axes de parcours  
Exploitation de la structure d'arbre d'un document XML
- Tests sur la valeur des éléments et des attributs
- Prédicats

```
axe_de_recherche::nom_élément_recherché[prédicat]
```

## Cheminement et sélection des nœuds

```
/personnel/service[2]/emploi/nom
/child::personnel/child::service[2]/child::emploi/child::nom
/child::personnel/child::service[position()=2]/child::emploi/child::nom

/child::personnel/descendant::emploi[child::groupe][2]

/child::personnel/child::service[attribute::id='ventes']
/child::personnel/child::service[@id='ventes']
```

## Axes de cheminement

- **child** : tous les enfants du nœud contextuel
- **parent** : l'unique nœud parent du nœud contextuel
- **self** : le nœud contextuel
- **attribute** : tous les nœuds attributs du nœud contextuel, s'il est élément
- **following-sibling**, **preceding-sibling** : tous les nœuds frères du nœud contextuel qui apparaissent après ou avant lui
- **ancestor**, **ancestor-or-self** : tous les ancêtres du nœud contextuel, son père ou lui-même en première position
- **descendant**, **descendant-or-self** : tous les descendants du nœud racine
- **following** : tous les nœuds suivant le nœud contextuel (cela exclut les descendants du nœud contextuel)
- **preceding** : tous les nœuds qui précèdent le nœud contextuel (cela exclut les descendants de ses ancêtres)
- **namespace** : tous les nœuds namespace du nœud contextuel

## Tests sur les nœuds

- Tests sur le nom de nœud
  - **child::personnel**
  - **attribute::\*** (tous les nœuds attributs du nœud contextuel)
- Tests sur le type de nœud
  - **descendant::comment()**
  - **child::text()**
  - **parent::processing-instruction()**
  - **parent::processing-instruction('xml-stylet')**
  - **ancestor::node()** (toujours vrai)

## Syntaxe abrégée

- **child::** peut être omis (child est l'axe par défaut)  
para ⇔ child::para
- **attribute::** peut être abrégé en **@**  
para[@type="warning"] ⇔ child::para[attribute::type="warning"]
- **//** ⇔ **/descendant-or-self::node()/**  
div//para ⇔ div/descendant-or-self::node()/child::para
- **.** ⇔ **self::node()** ⇔ **self::\***  
./para ⇔ self::node()/descendant-or-self::node()/child::para
- **..** ⇔ **parent::node()**  
../title ⇔ parent::node()/child::title

## Prédicats

```
/child::personnel/child::service[position()=2]/child::emploi/child::nom
/child::personnel/descendant::emploi[child::groupe][2]
/child::personnel/child::service[attribute::id='ventes']
/personnel/service/emploi[titre="Président"]/nom
```

- Filtrage d'un ensemble de nœuds selon un axe et production d'un nouvel ensemble de nœuds
- Si pour un nœud de l'ensemble, le prédicat est évalué à vrai il est conservé dans le nouvel ensemble, sinon il en est exclu

## Prédicats

### Fonctions prédéfinies

- Fonctions sur/renvoyant des ensembles de nœuds
  - node-set **id**(object)
  - number **position()**
  - number **last()**
  - number **count**(node-set)
  - string **name**(node-set)
  - string **local-name**(node-set)
  - string **namespace-uri**(node-set)
- Fonctions sur/renvoyant des chaînes de caractères
  - boolean **contains**(string, string)
  - string **concat**(string, string)
  - boolean **starts-with**(string, string)
  - string **substring**(string, number)
- Fonctions booléennes
  - boolean **boolean**(object)
  - boolean **not**(boolean)

## Prédicats

### Expressions booléennes

- La valeur booléenne d'une valeur est le résultat de l'application de la fonction boolean
- Opérateurs booléens : and, or
- Opérateurs de comparaison : = != > < <= >=

## Exemples

- `section/node()`  
selects all node children of section
- `section/*` ↔ `section/child::*`  
selects all element children of section
- `section/text()`  
selects all text children of section
- `section/@*` ↔ `section/attribute::*`  
selects all the attributes of section
- `section/*/para`  
selects all para grandchildren of section
- `/doc/chapter[5]/section[2]`  
selects the second section of the fifth chapter of doc

## Exemples

- `para[@type="warning"][5]`  
selects the fifth para child of the context node that has a type attribute with value warning
- `para[5][@type="warning"]`  
selects the fifth para child of the context node if that child has a type attribute with value warning
- `chapter[title]`  
selects the chapter children of the context node that have one or more title children
- `chapter[title="Introduction"]`  
selects the chapter children of the context node that have one or more title children with string-value equal to Introduction
- `employee[@secretary and @assistant]`  
selects all the employee children of the context node that have both a secretary attribute and an assistant attribute

## Exemples

- `/personnel/descendant::*[self::emploi]`  
Tous les éléments emploi descendants de personnel
- `/personnel/descendant::emploi[contains(titre, 'Président')]`  
Tous les éléments emploi descendants de personnel qui ont un fils élément titre contenant la chaîne 'Président'
- `/personnel/service/groupe[count(*)>5]`  
Tous les éléments groupe fils d'un élément service (fils d'un élément personnel) qui ont plus de cinq éléments fils