

# INFO-H-415

Bases de données XML  
XQuery

# XPath 2.0

- Expression de chemin dans un fichier XML
- Renvoie l'ensemble des noeuds correspondants
  - `//etudiant[prenom = "Avit"]/adresse`
  - `//client[@id = "42"]/nom`
  - `/clients/client/nom[.="Sidis"]`
- `/` : fils
- `//` : descendant
- `/u/* /v` : éléments v, petits fils de u
- `/u/* /name()` : noms des éléments fils de u

# XQuery 1.0

- XPath 2.0 + Expressions FLWOR
- “SQL pour XML”
- Turing complet
- Fonctions, récursivité, ...
- Construction de résultats

# Expressions FLOWR

- `(for|let)+ where? orderby? return!`
- `for` : itère sur un ensemble de noeuds
- `let` : assigne des valeurs à une variable
- `return` : construit le résultat
- **Exemple :**
  - ```
for $client in //clients
  let $adresse := $client/adresse
  where $client/nom = "Sidis"
  return <number>{$adresse/numero/text()}</number>
```
- Chaque clause peut être vue comme une boîte qui prend un flux de noeud en entrée et qui renvoie un flux de noeuds.

# Quantificateurs

- **Universel :**

where all  $\$var$  in  $expr$  satisfies  $bool\_expr$

- **Existentiel :**

where some  $\$var$  in  $expr$  satisfies  $bool\_expr$

# Construction de résultats

- Soit en toutes lettres

```
return <machin>{expr}</machin>
```

- Soit en utilisant des constructeurs :

```
return element machin {expr}
```

```
return element a {attribute b {12},  
                  expr}
```

# Fonctions utiles

- `distinct-values (expr)` : renvoie les *valeurs* distinctes de la séquence en entrée
- `count()`, `max()`, `abs()`, `avg()`
- `empty()`, `exists()`, `not()`
- `string-join($names, ", ")`
- `concat("(" , $name, ")")`
- `let $a :=if bool_expr then $b/u  
else $b/v`

# eXist

- Moteur de DB XML Native libre
- Lancer le client : `/opt/eXist/bin/client.sh`
- Choisir le mode “embedded”
- Aller dans la collection “dblp”
- Lancer le Query Dialog