PXML
M1-FA-MIAGE

TP3
XPATH 1.0

Auteur :
Tarik Djebien
Date :
***20 Janvier* 2012**

Exercice 1 :

Requête XPATH 1.0	***Résultat***
//eau/@id	{e1,e2,e3}

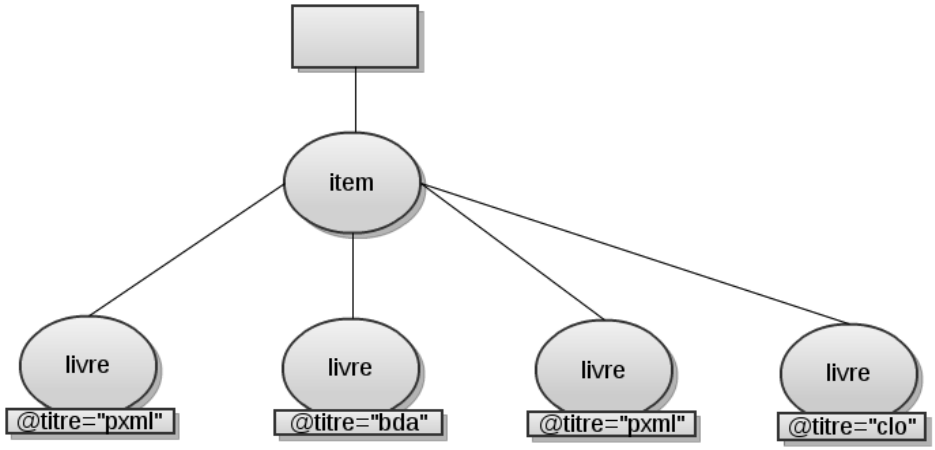
Exercice 2 : Subtilités XPATH

Question 1 :

Soit le fichier XML suivant :

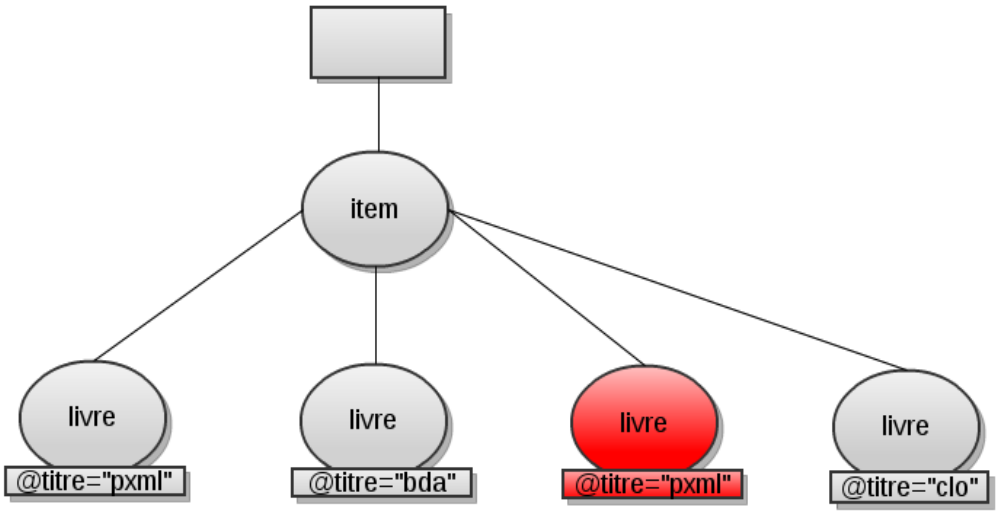
```
<item>
        <livre titre="pxml">livre1</livre>
        <livre titre="bda">livre2</livre>
        <livre titre="pxml">livre3</livre>
        <livre titre= "clo">livre4</livre>
</item>
```

1. **/item/livre[@titre=''pxml'' and position()=last()]**
   Ici, on sélectionne parmi tout les livres, fils d'item, le dernier livre et on vérifie s'il a un attribut titre dont la valeur est «pxml».



   Result = {}

2. **/item/livre[@titre=''pxml''][position()=last()]**
   Ici, on recherche tous les livres, fils d'item, dont l'attribut titre vaut « pxml ». Puis, parmi ceux ci, on choisit le dernier livre dans l'ensemble obtenu.



   Result = {livre3}

3. **/item/livre[position()=last()][@titre=''pxml'']**
   Ici, on obtient le même résultat que (1.) car on va récupérer le dernier livre et
   on va vérifier s'il respecte le prédicat de présence d'un attribut titre avec la valeur «pxml».
   Donc on a aussi Result = {}
   On remarque que un prédicat **[a and b]** equivaut à **[b][a]**

Question 2 :

- La requête **q1** XPATH **//livre[titre=''edition'']** recherche les livre ayant un nœud fils titre avec une valeur égale à "edition".
- La requête **q2** XPATH **//livre[titre=edition]** recherche les livre ayant deux nœuds fils titre et edition et dont la valeur du nœud titre est égale à la valeur du nœud edition.

Voici un exemple d'instance XML où **q1** et **q2** fournisse un résultat identique :

```
<item>
        <livre>
                <titre>edition</titre>
                <edition>edition</edition>
        </livre>
</item>
```

Question 3 :

- La requête **q3** XPATH **//livre[1]** recherche l'ensemble des premiers livre relatif au sein d'un même niveau de hauteur dans l'arbre XML.
- La requête **q4** XPATH **/descendant ::livre[1]** recherche à partir de la racine de l'arbre XML le premier élément livre qui sera rencontré.

Voici un exemple d'instance XML où **q3** et **q4** fournisse un résultat différent :

```
<item>
        <livre>Java et JDBC</livre>
          <mesLivres>
                <livre>Python programming language</livre>
                <livre>Java et XSLT</livre>
          </mesLivres>
</item>
```

Result (q3) = {livre 'Java et JDBC', livre 'Python programming language'}
Result (q4) = {livre 'Java et JDBC'}

Exercice 3 : Recettes

Recette1.dtd :

```
<!ELEMENT cuisine (recette)+>
<!ELEMENT recette (titre, categorie, ingredients,tps_preparation?, tps_cuisson?, texte, conseil?)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT tps_preparation (#PCDATA)>
<!ELEMENT tps_cuisson (#PCDATA)>
<!ELEMENT conseil (#PCDATA)>
<!ELEMENT categorie (#PCDATA)>

<!ELEMENT ingredients (ingredient)+>
<!ELEMENT ingredient (nom_ing,quantite)>
<!ELEMENT nom_ing (#PCDATA)>
<!ELEMENT quantite (#PCDATA)>

<!ELEMENT texte (etape)+>
<!ELEMENT etape (#PCDATA)>
```

Requêtes XPATH 1.0 Recette1:

1. Les éléments titres des recettes : **//recette/titre**
2. Les noms des ingrédients : **//nom_ing**
3. L'élément titre de la deuxième recette : **//recette[2]/titre**
4. La dernière étape de chaque recette : **//etape[position()=last()]**
5. Le nombre de recettes : **count(//recette)**
6. Les éléments recette qui ont strictement moins de 7 ingrédients : **//recette[count(./ingredients/ingredient)<7]**
7. Les titres (chaîne de caractères) des recettes qui ont strictement moins de 7 ingrédients : **//recette[count(./ingredients/ingredient)<7]/titre**
8. Les recettes qui utilisent de la farine : **//recette[contains(./ingredients/ingredient/nom_ing/child::text(),"farine")]**
9. Les recettes de la catégorie entrée : **//recette/categorie[text()="entrée"]**

Recette2.dtd :

```
<!ELEMENT cuisine (categorie+,ingredient+,recette+)>

<!-- les categories -->
<!ELEMENT categorie (#PCDATA|categorie)*>
<!ATTLIST categorie id ID #REQUIRED>

<!-- les ingredients -->
<!ELEMENT ingredient (#PCDATA)>
<!ATTLIST ingredient nom ID #REQUIRED>

<!-- les recettes -->
<!ELEMENT recette (titre, ingredients, texte, conseil?)>
<!ELEMENT titre (#PCDATA)>
<!ATTLIST recette categ IDREFS #REQUIRED
        temps-cuisson CDATA #IMPLIED
        temps-preparation CDATA #IMPLIED
        >
<!ELEMENT ingredients (ing-recette)+>
<!ELEMENT ing-recette (#PCDATA)>
<!ATTLIST ing-recette ingredient IDREF #REQUIRED>
<!ELEMENT conseil (#PCDATA)>
<!ELEMENT texte (etape)+>
<!ELEMENT etape (#PCDATA)>
```

Requêtes XPATH 1.0 Recette2:

10. Les éléments titres des recettes : **//recette/titre**
11. Les noms des ingrédients : **//ingredient/@nom**
12. L'élément titre de la deuxième recette : **//recette[2]/titre**
13. La dernière étape de chaque recette : **//etape[position()=last()]**
14. Le nombre de recettes : **count(//recette)**
15. Les éléments recette qui ont strictement moins de 7 ingrédients : **//recette[count(./ingredients/ing-recette)<7]**
16. Les titres (chaîne de caractères) des recettes qui ont strictement moins de 7 ingrédients : **//recette[count(./ingredients/ing-recette)<7]/titre**
17. Les recettes qui utilisent de la farine : **//recette[contains(./ingredients/ing-recette/@ingredient,"farine")]**
18. Les recettes de la catégorie entrée : **//recette[contains(@categ,"entree")]**

Exercice4 : Itunes

Requêtes XPATH 1.0 iTunes Music Library.xml :

1. Le nombre de morceaux (tracks hors PlayLists) de la bibliothèque.
   count(//key[text()="Track ID"])
2. Tous les noms d'albums.
   //key[text()="Album"]/following-sibling::string[position()=1]/text()
3. Tous les genres de musique (Jazz, Rock, . . .).
   //key[text()="Genre"]/following-sibling::string[position()=1]/text()
4. Le nombre de morceaux de Jazz.
   count(//string[text()="Jazz"])
5. Tous les genres de musique mais en faisant en sorte de n'avoir dans le résultat qu'une seule occurrence de chaque genre.
   //key[text()="Genre"]/following-sibling::string[position()=1][not(self::node() = following::string)]
6. Le titre (Name) des morceaux qui ont été écoutés au moins 1 fois.
   //key[text()="Disc Count"][number(following-sibling::integer[position()=1])>=1]/preceding-sibling::key[text()="Name"]/following-sibling::string[position()=1]/text()
7. Le titre des morceaux qui n'ont jamais été écoutés
   //key[text()="Name"][not(following-sibling::key[text()="Track Count"])]
   [not(preceding::key[text()="Playlists"])]/following-sibling::string[position()=1]
8. Le titre du (ou des) morceaux les plus anciens de la bibliothèque.

```
//key[text()="Date Added"]
[number(
  translate(
    substring(
        self::node()/following-sibling::date[position()=1],
        1,
        10
    ),"-","")
  )
 <=
number(
 translate(
   substring(
     self::node()/following::key[text="Date Added"]
     /following-sibling::date[position()=1],
   1,
   10
   ),"-","")
 )
]
/preceding::key[text()="Name"]/following-sibling::string[position()=1]
```