

Exercice 1

Question 1

//livre[titre="edition"] : selectionne le livre dont le titre est edition

//livre[titre=edition] : selectionne le livre dont il existe un element en commun entre le titre et edition

exemple:

...

<livre>

<titre>edition</titre>

<edition>edition </edition>

</livre>

...

Question 2

1. /item/livre[@titre="labd" and position()=last()]

Commentaire :

La requête donne le dernier livre dont le titre est labd

2. /item/livre[@titre="labd"] [position()=last()]

Commentaire :

/item/livre[@titre="labd"], on a un ensemble de livres labd

/item/livre[@titre="labd"] [position()=last()], dans cet ensemble precedent,

On envoie le dernier livre labd

1. /item/livre[position()=last()] [@titre="labd"]

Commentaire :

Exemple à faire

/item/livre[position()=last()] , on envoie le dernier livre

/item/livre[position()=last()] [@titre="labd"] on regarde si son attribut titre est labd

L'exemple serait de créer un ensemble de livres dont le dernier n'aurait pas le titre labd

...

<livres>

<livre titre='labd'>

<livre>

<livre titre='B'>

`<livre titre='C'>`

`<livre>`

`<livre>`

`</livres>`

Question 3

`/descendant::livre[1]` : sélectionne le premier livre pour descendant

`//livre[1]` : Le premier sélectionne tous les livres pour descendant qui sont les premiers enfants pour leurs parents.

Exemple à faire

exercice 2 :

recette1

1. les producteurs de fruits : `/ produits/ fruit / producteur`

2. les légumes produits en Espagne :

`//legume[contains(origine, 'Espagne')]`

3. les origines des clémentines de calibre 1 issues de l'agriculture biologique :

`//fruit[@type=' clementine' and @calibre='1']/
origine/bio/parent ::*/origine`

4. les producteurs bretons : `/origine|@region='Bretagne']/parent ::*/producteur`

Commentaire :

`parent ::*` permet de monter d'un nœud parent donc de origine au producteur

le résultat de la requete est la dernière partie de la requête

exercice 3 :

Question 2 :

Recette 1 :

1 Les éléments titres des recettes : `//recette/titre`

2. Les noms des ingrédients : `//ingredient/nom_ing`

3. L'élément titre de la deuxième recette : `//recette[2]/ titre`

Commentaire :

`recette[2]` permet d'accéder à la deuxième recette

2. La dernière étape de chaque recette :

`//recette/texte/etape[position()=last()]`

5. Le nombre de recettes : `count(//recette)`

6. Les éléments recette qui ont strictement moins de 7 ingrédients :

`// recette/ingredients[count(//ingredient) < 7)]/parent :: *`

7. Les titres des recettes qui ont strictement moins de 7 ingrédients :

```
// recette/ingredients[count(//ingredient) < 7)]  
  
/parent :: */ titre
```

8. Les recettes qui utilisent de la farine :

```
//ingredients(contains(nom_ing,' farine'))/  
/parent :: *
```

Commentaire :

```
ingredients(contains(nom_ing,' farine'))
```

ingredients qui contient la balise `nom_ing` qui a
pour valeur farine

8. Les recettes de la catégorie entrée

```
//recette(contains(categorie,' entrée'))
```

Recette2

1 Les éléments titres des recettes : `//recette/titre`

2. Les noms des ingrédients : `//ingredient/@NOM`

Commentaire :

`@NOM` : j'accède aux attributs nom de la balise
`ingredient`

3. L'élément titre de la deuxième recette : `//recette[2]/titre`

4. La dernière étape de chaque recette : `//texte /etape[last()]`

5. Le nombre de recettes : `count(//recette)`

Commentaire :

On a pas besoin de / ou // devant car count n'est pas un element de l'arbre

6. Les éléments recette qui ont strictement moins de 7 ingrédients :

```
//ingredients [COUNT(ing-recette] <7]/parent ::*
```

Commentaire :

Parent ::* ou parent ::recette ou /..

7. Les titres des recettes qui ont strictement moins de 7 ingrédients :

```
//ingredients [COUNT(ing-recette]  
<7]/parent ::*/titre
```

Commentaire :

C'est une suite de la question precedente

parent ::*/titre , on monte au parent recette et on accède à son fils titre

8. Les recettes qui utilisent de la farine :

```
// ingredients/ ing-  
recette[@ingredient='farine']/../..
```

Commentaire :

Arriver à l'attribut `ing-recette` pour monter aux recettes,*

On monte deux parents ingredients via .. et on arrive ensuite à la balise recette via le deuxieme ..

9. Les recettes de la catégorie entrée : `// recette[contains(@categ, 'entree')]`

Commentaire :

Contains : parceque l'attribut `categ` contient le mot entrée et bien plus

Exercice 4 :

`1]count(//key[text()='Artist'])`

Commentaire :

`text()` : permet d'accéder au contenu de la balise `key`

`2] //key[text()='Album']/ following-sibling ::*[1]`

Commentaire :

`following-sibling` : voisin de droite d'après l'organisation du fichier xml

`following-sibling ::*[1]` signifie premier voisin de droite

`3] //key[text()='Genre'] / following-sibling ::*[1]`

`4] count[//string[text()='jazz']`

```
5] distinct-values(//key [text()='Genre'] / following-sibling ::*[1]
```

Commentaire :

distinct-values, permet d'éliminer les doublons

```
6] //dict/key[text()='Play  
Count']/key[text()='Name']/following-sibling ::*[1]
```

```
7] //dict[not(key='Play Count')]/key[text()='  
Name']/ following-sibling ::*[1]
```

Commentaire :

Entre les crochets, on a bien l'opposée du
predicat : key='Play Count'

Les predicats sont bien entre les crochets

```
8] //dict[string=min(//key[text()='Year']/  
following-sibling ::*[1]))/ key[text()='Name']/  
following-sibling ::*[1]
```

Commentaire :

La fonction min me permet de récupérer le minimum
de chaque année

