

Sujet 2 - XPath

Exercice 1

1. /Contacts

Il retourne la noeud "Contacts"

2. /Contacts/Person

Il retourne la noeud "Person"

3. //Person[Firstname="John"]

Il retourne la noeud "Person" qui a le "Firstname" qui est "John"

4. //Person[Email]

Il retourne les noeuds "Person" qui a le "Email" dedans

5. /Contacts/Person[1]/Firstname/child::text()

John

Il retourne le text de "Firstname" dans le "Person[1]" . C'est le "John"

6. /Contacts/Person[1]/Firstname/text()

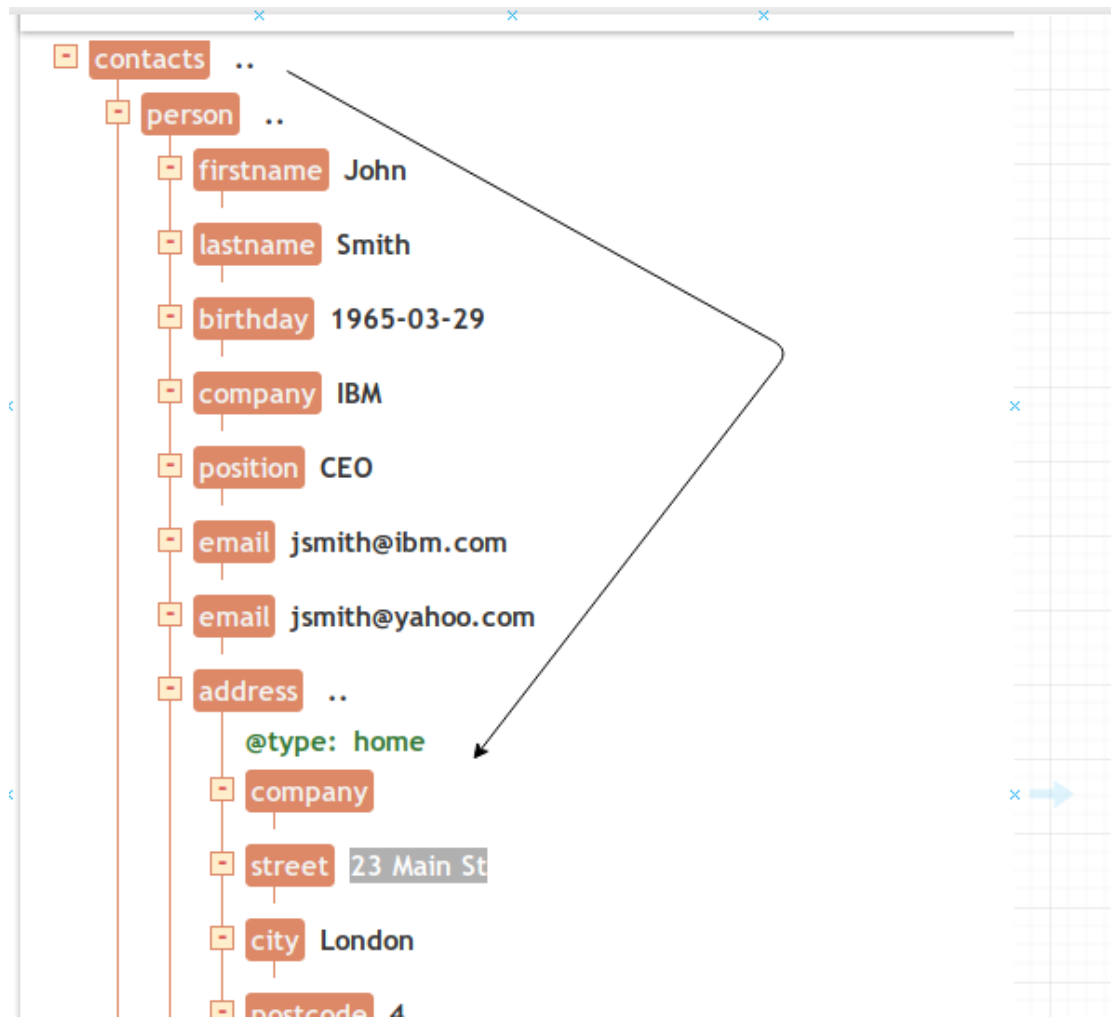
Même résultat avec la requête précédente

Comparez 5 et 6

child::text() est en même avec text()

7. /Contacts//Address[@type="home"]//Street/child::text()

23 Main St



Il retourne le text de "Street" des Person qui a l'address type qui est "home"

8. /Contacts//Address[@type="home" and City="London"]

```
<Address type="home">
  <Company/>
  <Street>23 Main St</Street>
  <City>London</City>
  <Postcode>4</Postcode>
  <Country/>
</Address>
```

Il trouve l'Address qui a le type "home" et le City "London" ensemble

9. /Contacts//Address[@type="work" and City="Dublin"]/parent::node()/Lastname/text()

Smith

d'abord, la requête "/Contacts//Address[@type="work" and City="Dublin"]" retourne:

```
<Address type="work">
  <Street>1234 High St</Street>
```

```
<City>Dublin</City>
<ZIP>1234</ZIP>
<Country>Ireland</Country>
</Address>
```

Après, il retourne la noeude parent pour donner sa Lastname avec `"/parent::node()/Lastname/text()`

10. `/Contacts//Address[@type="work" and City="Dublin"]/../Lastname/text()`

Smith

Comparez la requête 9 et 10

`"parent::node()"` est en même avec `".."`

11. `/Contacts[./Address[@type="work" and City="Dublin"]]/Lastname/text()`

SmithDunne

C'est pas la même résultat avec la requête 9 et 10

12. `/Contacts//Address[@type="work"]/ancestor::node()`

```
<Contacts>
  <Person>
    <Firstname>John</Firstname>
    <Lastname>Smith</Lastname>
    <Birthday>1965-03-29</Birthday>
    <Company>IBM</Company>
    <Position>CEO</Position>
    <Email>jsmith@ibm.com</Email>
    <Email>jsmith@yahoo.com</Email>
    <Address type="home">
      <Company/>
      <Street>23 Main St</Street>
      <City>London</City>
      <Postcode>4</Postcode>
      <Country/>
    </Address>
    <Address type="work">
      <Street>1234 High St</Street>
      <City>Dublin</City>
      <ZIP>1234</ZIP>
      <Country>Ireland</Country>
    </Address>
  </Person>
  <Person>
    <Firstname>Tom</Firstname>
    <Lastname>Dunne</Lastname>
    <Company>Today FM</Company>
    <Position/>
```

```
<Email>tom.dunn@todayfm.com</Email>
</Person>
</Contacts>
<Contacts>
  <Person>
    <Firstname>John</Firstname>
    <Lastname>Smith</Lastname>
    <Birthday>1965-03-29</Birthday>
    <Company>IBM</Company>
    <Position>CEO</Position>
    <Email>jsmith@ibm.com</Email>
    <Email>jsmith@yahoo.com</Email>
    <Address type="home">
      <Company/>
      <Street>23 Main St</Street>
      <City>London</City>
      <Postcode>4</Postcode>
      <Country/>
    </Address>
    <Address type="work">
      <Street>1234 High St</Street>
      <City>Dublin</City>
      <ZIP>1234</ZIP>
      <Country>Ireland</Country>
    </Address>
  </Person>
  <Person>
    <Firstname>Tom</Firstname>
    <Lastname>Dunne</Lastname>
    <Company>Today FM</Company>
    <Position/>
    <Email>tom.dunn@todayfm.com</Email>
  </Person>
</Contacts>
<Person>
  <Firstname>John</Firstname>
  <Lastname>Smith</Lastname>
  <Birthday>1965-03-29</Birthday>
  <Company>IBM</Company>
  <Position>CEO</Position>
  <Email>jsmith@ibm.com</Email>
  <Email>jsmith@yahoo.com</Email>
  <Address type="home">
    <Company/>
    <Street>23 Main St</Street>
    <City>London</City>
    <Postcode>4</Postcode>
    <Country/>
  </Address>
  <Address type="work">
    <Street>1234 High St</Street>
    <City>Dublin</City>
    <ZIP>1234</ZIP>
    <Country>Ireland</Country>
  </Address>
</Person>
```

"ancestor::node()" Sélectionner tous les nœuds du nœud actuel

13. /Contacts/Person[Lastname="Smith"]/following-sibling::node()/Lastname/text()

Dunne

"following-sibling::node()" sélectionne tous les noeuds du noeud qui a "Person[Lastname="Smith"]" . Ici, c'est le:

```
<Person>
  <Firstname>Tom</Firstname>
  <Lastname>Dunne</Lastname>
  <Company>Today FM</Company>
  <Position/>
  <Email>tom.dunn@todayfm.com</Email>
</Person>
```

Après, "Lastname/text()" retourne le "Dunne"

14. /Contacts/Person[following-sibling::node()/Lastname="Dunne"]/Lastname/text()

Smith

La condition "following-sibling::node()/Lastname="Dunne"" sélectionne des noeuds qui n'ont pas le "Lastname = Dunne" dedans.

Exercice 2

1. Toutes les compositions

```
/CDlist/CD/performance//composition/text()
```

2. Toutes les compositions ayant un soloist

```
/CDlist/CD//performance[soloist]/composition/text()
```

ou

```
/CDlist/CD//performance/soloist/../../composition/text()
```

3. Toutes les performances avec un seul orchestra mais pas de soloist

```
/CDlist/CD//performance[orchestra and not(soloist)]
```

Resultat:

```
<performance>
  <composition>Piano Concerto No. 3</composition>
  <orchestra>London Symphony Orchestra</orchestra>
  <conductor>Eugen Jochum</conductor>
</performance>
```

4. Tous les soloists ayant joué avec le London Symphony Orchestra sur un CD publié par Deutsche Grammophon

```
/CDlist/CD[publisher="Deutsche Grammophon"]//performance[orchestra="London
Symphony Orchestra"][soloist]/soloist/text()
```

Resultat:

```
Amy Gibber
```

Sauf "/soloist/text()", on voit que:

```
<performance>
  <composition>Fantasias Op. 54</composition>
  <soloist>Amy Gibber</soloist>
  <orchestra>London Symphony Orchestra</orchestra>
  <conductor>Peter Zokarovic</conductor>
</performance>
```

5. Tous les CDs comportant des performances du London Symphony Orchestra

```
/CDlist/CD[./performance[orchestra="London Symphony Orchestra"]]
```

Exercice 3

1. Le titre du cinquième livre dans la liste

```
/booker/award[5]/title/text()
```

Resultat:

```
Hotel du Lac
```

2. L'auteur du sixième livre dans la liste.

```
/booker/award[6]/author/text()
```

Resultat:

```
A S Byatt
```

3. Le titre du livre qui a gagné en 2000

```
/booker/award[year=2000]/title/text()
```

Resultat:

```
<title>The Blind Assassin</title>
```

4. Le nom de l'auteur du livre intitulé Possession

```
/booker/award[title="Possession"]/author/text()
```

Resultat:

```
A S Byatt
```

5. Le titre des livres dont J M Coetzee est l'auteur.

```
/booker/award[author="J M Coetzee"]/title
```

Resultat:

```
<title>Disgrace</title>  
<title>Life and Times of Michael K</title>
```

6. Le nom de tous les auteurs qui ont obtenu un prix depuis 1995

```
/booker/award[year > 1995]/author/child::text()
```

Resultat:

Margaret Atwood
J M Coetzee
Ian McEwan
Arundhati Roy
Graham Swift

7. Le nombre total de prix decernes.

```
sum(/booker/award)
```

Resultat:

35

Exercice 4:

Requete	recette1.xml	recette2.xml
1	/cuisine/recette/titre	/cuisine/recette/titre
2	/cuisine/recette/ingredients/ingredient/nom_ing/text()	/cuisine/ingredient/@nom
3	/cuisine/recette[2]/titre/text()	/cuisine/recette[2]/titre/text()
4	/cuisine/recette/texte/etape[position() = last()]	/cuisine/recette/texte/etape[position() = last()]
5	count(/cuisine/recette)	count(/cuisine/recette)
6	/cuisine/recette/ingredients[count(ingredient) < 7]/..	/cuisine/recette/ingredients[count(/ing-recette) < 7]
7	/cuisine/recette/ingredients[count(ingredient) < 7]/../titre/text()	/cuisine/recette/ingredients[count(/ing-recette) < 7]/../titre/text()
8	/cuisine/recette/ingredients/ingredient[nom_ing="farine"]/parent::*/..	/cuisine/recette/ingredients/ing-recette[@ingredient = "farine"]/../../
9	/cuisine/recette[categorie = "Entrée"]	/cuisine/recette[contains(@categ, 'entree')]

Exercice 5:

1. Le nombre de morceaux (tracks hors PlayLists) de la bibliotheque.

```
count(/plist[@version="1.0"]/dict/dict/key)
```

Resultat:


```
2123
```

2. Tous les noms d'albums

```
/plist[@version="1.0"]/dict/dict/dict/key[text() =  
"Album"]/following::string[1]/text()
```

3. Tous les genres de musiques

```
/plist[@version="1.0"]/dict/dict/dict/key[text() =  
"Genre"]/following::string[1]/text()
```

4. Le nombre de morceaux de Jazz

```
count(/plist[@version="1.0"]/dict/dict/dict/key[text() = "Genre" and  
following::string[1] = "Jazz"])
```

Resultat:

```
183
```

5. Tous les genres de musique mais en faisant en sorte de n'avoir dans le resultat qu'une seule occurrence de chaque genre.

6. Le titre (Name) des morceaux qui ont ete ecoute au moins 1 fois

```
count(/plist[@version="1.0"]/dict/dict/dict/key[text() = "Play Count"]/preceding-  
sibling::node()[text() = "Name"]/following-sibling::node()[1]/text())
```

7. Le titre des morceaux qui n'ont jamais ete ecoutes

```
count(/plist[@version="1.0"]/dict/dict/dict[not(key[text() = "Play  
Count"])]/preceding-sibling::node()[text() = "Name"]/following-sibling::node()  
[1]/text())
```

8. Le titre du (ou des) morceaux les plus anciens (renseignement Year) de la bibliotheque. (J'AI FAIT CET EXERCICE AVEC UN AMI DANS LE CLASSE: ERWAN)

```
//key[text() = 'Year'][not(following-sibling::integer[1] > //integer[preceding-  
sibling::key[1]/text()='Year'])]/preceding-sibling::key[text()='Name']/following-
```

```
sibling::string[1]/text()
```