XPath

L3 Info & L3 MIASHS *TP2 – Corrigé*

Exercice 1.

Question 1. Renvoie l'ensemble du contenu de la balise *Contacts*.

Question 2. Renvoie l'ensemble du contenu des balises *Person* inclues dans la balise *Contacts*.

Question 3. Renvoie l'ensemble du contenu des balises *Person* dont le prénom est *John*.

Question 4. Renvoie l'ensemble du contenu des balises *Person* possédant une balise *Email*.

Question 5. Renvoie la valeur de la balise *Firstname* de la première personne apparaissant dans *Contacts*.

Question 6. Le résultat est identique. L'expression XPath `/child::text()` est équivalente à `/text()`.

Question 7. Retourne la valeur de la balise *Street* contenue dans la balise *Addresse* qui a l'attribut *type* égal à *home*. L'utilisation des `//` indique qu'on regarde parmi tous les descendants.

Question 8. Renvoie l'ensemble du contenu des balises *Address* de type *home* et contenant une balise *City* ayant pour valeur *London*.

Question 9. Renvoie la valeur de la balise *Lastname* de la balise qui est le parent de la balise *Address* de type *work* et localisée à Dublin.

Question 10. Le résultat est identique. L'expression XPath `/parent::node()` est équivalente à `/..`

Question 11. La requête récupère dans un premier temps la balise *Contacts* possédant parmi ses descendants une balise *Address* de type *work* et localisée à Dublin. Puis récupère la valeur de toutes les balises *Lastname* descendants de celle-ci.

Question 12. Renvoie l'ensemble du contenu des balises ancêtres de la balise *Address* de type *work*. À savoir, le *document*, la balise *Contacts*, et la balise *Person* de John Smith.

Question 13. Renvoie la valeur de la balise *Lastname* de la balise sœur de la personne ayant pour nom de famille *Smith*.

Question 14. Renvoie la valeur de la balise *Lastname* contenue dans la balise *Person* qui a une sœur après elle possédant une balise *Lastname* égale à Dunne.

Exercice 2.

Question 1. `/CDlist/CD/performance/composition/text()`

Question 2. `/CDlist/CD/performance[soloist]/composition/text()`

Question 3. `/CDlist/CD/performance[not(soloist) and count(orchestra)=1]`

Question 4. `/CDlist/CD/performance[orchestra="London Symphony Orchestra"`

```
and ../publisher="Deutsche Grammophon"]/soloist/text()
```

Question 5. `/CDlist/CD[performance/orchestra="London Symphony Orchestra"]`

Exercice 3.

Question 1. `/booker/award[5]/title/text()`

Question 2. `/booker/award[6]/author/text()`

Question 3. `/booker/award[title/@price="2000"]/title/text()`

Question 4. `/booker/award[title="Possession"]/author/text()`

Question 5. `/booker/award[author="J M Coetzee"]/title/text()`

Question 6. `/booker/award[year >= 1995]/author/text()`

Question 7. `count(/booker/award)`

Exercice 4.

Question 1.
- `recettes1.xml : /cuisine/recette/titre`
- `recettes2.xml : /cuisine/recette/titre`

Question 2.
- `recettes1.xml : /cuisine/recette/ingredients/ingredient/nom_ing/text()`
- `recettes2.xml : /cuisine/recette/ingredients/ing-recette/@ingredient`

Question 3.
- `recettes1.xml : /cuisine/recette[2]/titre`
- `recettes2.xml : /cuisine/recette[2]/titre`

Question 4.
- `recettes1.xml : /cuisine/recette/texte/etape[last()]`
- `recettes2.xml : /cuisine/recette/texte/etape[last()]`

Question 5.
- `recettes1.xml : count(/cuisine/recette)`
- `recettes2.xml : count(/cuisine/recette)`

Question 6.
- `recettes1.xml : /cuisine/recette[count(ingredients/ingredient) < 7]`
- `recettes2.xml : /cuisine/recette[count(ingredients/ing-recette) < 7]`

Question 7.
- `recettes1.xml : /cuisine/recette[count(ingredients/ingredient) < 7]/titre/text()`
- `recettes2.xml : /cuisine/recette[count(ingredients/ing-recette) < 7]/titre/text()`

Question 8.
- `recettes1.xml : /cuisine/recette[ingredients/ingredient/nom_ing="farine"]`
- `recettes2.xml : /cuisine/recette[ingredients/ing-recette/@ingredient="farine"]`

Question 9.
- `recettes1.xml : /cuisine/recette[categorie="Entrée"]`
- `recettes2.xml : /cuisine/recette[contains(@categ, "entree")]`

Exercice 5.

Question 1. `count(/plist/dict/dict/key)`

Question 2. `/plist/dict/dict/dict/key[. = "Album"]/following-sibling::string[1]`

Question 3. `/plist/dict/dict/dict/key[. = "Genre"]/following-sibling::string[1]`

Question 4. `count(/plist/dict/dict/dict/key[. = "Genre"]/following-sibling::string[1 and text() = "Jazz"])`

Question 5. `(/plist/dict/dict/dict/key[. = "Genre"]/following-sibling::string[1])[not(text() = following::text())]`

Question 6. `/plist/dict/dict/dict/key[. = "Play Count"]/following-sibling::integer[1][. > 0]/../key[. = "Name"]/following-sibling::string[1]`

Question 7. `/plist/dict/dict/dict[not(key="Play Count")]/key[. = "Name"]/following-sibling::string[1]`

Question 8. `/plist/dict/dict/dict[key="Year" and not(integer[preceding-sibling::key[1]="Year"] > ../dict/integer[preceding-sibling::key[1]="Year"])]/key[. = "Name"]/following-sibling::string[1]`