

Exercice 1 (XPath)

Le fichier booker.xml contient une liste de livres (les gagnants du fameux booker prize) avec leur auteur et l'année de l'obtention du prix. Un extrait de ce fichier est donnée ci-dessous :

```
<?xml version="1.0"?>

<booker>

  <award>

    <author gender='m'>Kingsley Amis</author>
    <title price=50>The Old Devils</title>
    <year>1986</year>

  </award>

  <award>

    <author gender='f'>Margaret Atwood</author>
    <title price=100>The Blind Assassin</title>
    <year>2000</year>

  </award>

  <award>

    <author gender='m'>John Berger</author>
    <title price=25>G</title>
    <year>1972</year>

  </award>

  ..

</booker>
```

Trouvez les expressions XPath qui retournent les informations suivantes :

1. les noms de tous les auteurs,
2. le cinquième livre dans la liste,
3. l'auteur du sixième livre dans la liste,
4. les noms des auteures (féminins),
5. le titre du livre qui a gagné en 2000,
6. L'auteur du prix situé (dans le document XML) après le prix décerné en 1986,
7. le nom de l'auteur du livre intitulé "Possession",
8. le titre des livres dont "J M Coetzee" est l'auteur,
9. le nom de tous les auteurs qui ont obtenu un prix depuis 1995,
10. le nombre total de prix décernés,
11. les auteurs d'au moins un livre dont le prix est supérieur à 30\$,
12. renvoyer les auteurs des livres dont le titre contient le mot "Blind",
13. tous les livres de Margaret Atwood ayant gagnés un prix,
14. le prix moyen des livres écrits par des auteures féminins.

Exercice 2 (XSLT)

Considérons le document booker.xml précédent. Ecrire les programmes XSLT qui effectuent les transformations ci-dessous.

1. Affichent la liste des auteurs dans un format HTML qui respecte le format d'affichage suivant :

Lauréats booker prize

Auteur : nom de l'auteur

Titre : titre

Année : année dans laquelle le prix a été gagné

2. Ecrire un template XSLT qui permet d'afficher les auteurs dans une liste HTML comme suit :

```
<ul>
  <li> Kingsley Amis </li>
  <li> Margaret Atwood </li>
  ...
</ul>
```

3. Ecrire un template XSLT, qui réutilise le template précédent (en utilisant xsl:apply-templates), pour générer le document HTML suivant :

```
<h1> Lauréats booker prize </h1>
  <h2>Authors</h2>
    <ul>
      <li> Kingsley Amis </li>
      <li> Margaret Atwood </li>
      ...
    </ul>
```

4. En utilisant le constructeur `<xsl:if>`, écrire un template XSLT qui permet d'afficher les auteurs d'un livre ayant un prix `>` à 65\$.
5. Transformer le template précédent en un template paramétré, nommé *Select-par-prix* et ayant comme paramètre d'entrée *mprix*, qui permet d'afficher les auteurs d'un livre dont le prix est supérieur à la valeur du paramètre *mprix*.
6. Ecrire un template XSLT qui permet d'afficher la liste des auteurs. Le nom des auteures féminin doit être en gras et ceux des auteurs masculin en italique (utiliser le constructeur `<xsl:choose>`).
7. Ecrire un template XSLT qui permet d'afficher la liste des livres par ordre croissant des années d'attribution des prix.
8. Ecrire un template XSLT qui permet d'afficher les auteurs et le nombre de prix pour chaque auteur

Exercice 3

On veut écrire un programme XSLT générique, c'est à dire qui peut s'appliquer à n'importe quel document XML. Ce programme prend un document XML en entrée et renvoie en sortie un *autre* document XML :

- La racine du document de sortie est un élément `<DOCUMENT>`
- Chaque élément du document d'origine :

`<tag att1="v1" . . .attn="vn"> . . .</tag>`

est transformé en :

`<ELEMENT name="tag"> <ATTRIBUTES count="n">`

`<att1 >v1 </att1 > ...`

`<attn >vn </attn > </ATTRIBUTES> <CHILDREN>...`

`</CHILDREN>`

`</ELEMENT>`— Chaque noeud texte (type `#PCDATA`) est transformé en un élément :

`<TEXT> </TEXT>`