# LO17 – TD : Introduction au Langage Perl

## 1 Introduction

### 1.1 Fichier de travail

Le fichier que nous allons utiliser pour découvrir les différentes fonctionnalités de *Perl* est un fichier bibliographique balisé dont le nom est `sa_b.tei` [1]. Sa structure est donnée figure 1. À une ligne (une suite de caractères terminée par un caractère fin de ligne) - sauf celle contenant `<listBiBibl>` et `</listBibl>` - correspond un enregistrement bibliographique.

```
<listBibl>
...
<bibl id=1><author>Gross, M.</author><date>1990</date><title>Sur la
      notion harrissienne de transformation et son application au français
      </title><title level=j>Langage</title><biblScope>99</biblScope>
      <publisher>Larousse</publisher><pubPlace>Paris</pubPlace>
      </bibl>¶
<bibl id=2><author>Lyons, J.</author><date>1980</date><title>
      Sémantique linguistique</title><publisher>Larousse</publisher>
      <pubPlace>Paris</pubPlace></bibl>¶
<bibl id=3><author>Maingueneau, D.</author><date>1986</date><title>
      Éléments de linguistique pour le texte littéraire</title>
      <publisher>Bordas</publisher><pubPlace>Paris</pubPlace></bibl>¶
...
</listBibl>
```

FIGURE 1 – Extrait du fichier `sa_b.tei` [1] (Les caractères de contrôle apparaîssent en clair pour améliorer la lisibilité de la structure de chaque ligne)

### 1.2 Rappel des notions de base

**En-tête** Les programmes *Perl* doivent débuter par la commande :

```
#!/usr/bin/perl
```

**Commentaires** Le caractère `#` est utilisé pour commenter les programmes.

**Variables** Les variables *Perl* (les *scalaires*) sont de la forme `$<chaîne>`, où `<chaîne>` est une suite de caractères qui ne doit pas contenir d'opérateur pré-défini de *Perl* .

**Syntaxe** Un programme *Perl* , comme un programme C, est une suite d'instructions terminées par un point-virgule. Pour les structures de contrôle (`while, if, ...`) les *blocs de programme* sont *obligatoirement* délimités par des accolades ouvrantes et fermantes : `{` et `}`.

## 2 Exercices

1. Écrire un programme *Perl* qui affiche chaque ligne du fichier `sa_b.tei` précédée d'un numéro de ligne.
2. On souhaite connaître le nombre d'entrées bibliographiques présentes dans le fichier de travail.

   Pour chaque ligne, vous allez compter le nombre d'occurrences de la balise `<bibl>` (on suppose ici qu'il n'y a qu'une seule occurrence de la balise de début d'entrée bibliographique par ligne et qu'elle est placée en début de ligne). Pour cela, vous allez parcourir tout le fichier de données ligne par ligne et pour chaque ligne vous regarderez si elle commence par la balise `<bibl>`. Si c'est le cas, alors vous incrémenterez un compteur.
3. On cherche à extraire le texte placé entre deux balises. Le nom de la balise sera indiqué en argument de la commande. Vous utiliserez une expression régulière pour extraire sur chaque ligne, le texte placé entre les suites de caractères `<balise>` et `</balise>`.
4. On veut afficher la liste des balises utilisées dans le fichier de travail et les trier par ordre alphabétique.

   Vous allez parcourir le fichier ligne par ligne, et pour chaque ligne :

   (a) vous éliminerez le texte placé entre les balises ;

   (b) pour chaque nouvelle balise vous créerez un nouvel élément dans un tableau ;

   Lorsque le fichier aura entièrement été parcouru, le tableau contenant la liste des balises sera trié, dans l'ordre alphabétique, puis affiché.

   Note : pour cet exercice, vous utiliserez au maximum la variable par défaut `$_`.

## Références

[1] Benoît Habert – Cecile Fabre – Fabrice Issac. De l'écrit au numérique : constituer, normaliser, exploiter les corpus électroniques. Paris, InterEdit., 1998.