

Compte rendu du TP

Le compte rendu (à rendre à la fin du TP sur Moodle) sera constitué d'une archive contenant votre code **commenté**.

Indications

Pour chaque exercice vous devrez créer un fichier *Python* répondant à la question. Un code source *Python* porte l'extension `.py` et peut être exécuté depuis le terminal en utilisant la commande suivante :

```
$ python fichier.py
```

Voici un code source minimal qui importe la bibliothèque de fonctions *ElementTree* puis lit un fichier XML. L'arbre XML est stocké dans la variable `root`.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Importation de l'API.
import xml.etree.ElementTree as ET

# Chemin vers le fichier xml.
FILE = 'championnat.xml'

# Création du parser et récupération de l'arbre XML du document.
root = ET.parse(FILE).getroot()
```

Exercice 1.

Question 1. Exécutez le fichier source `exercice1.py` et indiquez ce qu'il produit.

Question 2. Modifiez le code contenu dans `exercice1.py` afin qu'il affiche tous les clubs et toutes les journées du fichier `championnat.xml`.

Exercice 2.

Question 1. Vérifiez que chaque journée comporte 10 rencontres. Indiquez les journées ne vérifiant pas cette propriété.

Question 2. Écrivez un code python générant un document XML contenant les journées et les rencontres du championnat, mais n'indiquant pas les scores.

Exercice 3.

Question 1. Écrivez un code python permettant d'indiquer pour quels couples de clubs il manque un match. Si pour un couple de clubs (c_1, c_2) il manque les deux matchs, il faudra indiquer que le match c_1c_2 (où c_1 est receveur) et c_2c_1 (où c_2 est receveur) sont tous les deux manquants.

Question 2. Indiquez pour chaque club combien de buts il a marqué et combien de buts il a reçu au total sur tout le championnat. Affichez également contre quel clubs il a le plus marqué. Si il y a plusieurs clubs répondant à cela, n'en retourner qu'un seul.

Exercice 4.

Écrivez un code python permettant de générer un document XML contenant uniquement la liste des clubs. Cependant, chaque balise *club* doit contenir la liste des rencontres pour lesquels il était receveur. Les rencontres doivent être regroupé sous la balise *rencontresReceveur*. **Conseil :** créez une fonction `ajouter_rencontre(rencontre, racine)` qui ajoute le sous-arbre *rencontre* au sous-arbre *racine*.

Exercice 5.

Vérifiez que les journées sont numérotés par rapport à l'ordre chronologique (.i.e la journée 1 se déroule avant la journée 2 qui se déroulé avant la journée 3, etc ...). **Conseil :** créez une fonction `is_pair_correct(journee1, journee2)` qui retourne vraie si la *journee1* et *journee2* sont correctes d'un point de vue chronologique. Servez vous de la bibliothèque *datetime* permettant de comparer des dates.