

TP 3

Exercice 1: Fonctionnement des listes

Prévoyez le résultat que vous obtiendrez si vous exécutez le code suivant :

```
s = [0 , 0]
t = s
t[0] = 5
s[1] = 9
print( s )
print( t )
```

Exécutez ce code sur la machine et comparez avec ce que vous avez prédit. Si vous vous êtes trompé, essayer d'expliquer pourquoi.

Prévoyez le résultat que vous obtiendrez si vous exécutez les codes suivants :

```
import copy
s = [0 , 0]
t = copy.copy(s)
t[0] = 5
s[1] = 9
print( s )
print( t )
```

```
import copy
s = [[0,1] , [2,3]]
t = s
t[0][0] = 5
s[0][1] = 9
print( s )
print( t )
```

```
import copy
s = [[0,1] , [2,3]]
t = copy.copy(s)
t[0][0] = 5
s[0][1] = 9
print( s )
print( t )
```

```
import copy
s = [[0,1] , [2,3]]
t = copy.deepcopy(s)
t[0][0] = 5
s[0][1] = 9
print( s )
print( t )
```

Exécutez ce code sur la machine et comparez avec ce que vous avez prédit. Que fait `copy` et `deepcopy` ?

Exercice 2: Le miroir

Écrire un programme qui renverse tous les éléments d'un tableau `T` passé en paramètre.

Écrire un programme qui prend en paramètre un tableau `T` et qui renvoie un nouveau tableau qui est le miroir de `T`.

Exercice 3: Boîte à outils

Dans cet exercice, vous allez écrire des fonctions qui seront réutilisées plus tard dans d'autres exercices et d'autres TPs.

1) Tapez le programme suivant :

```
def demander_un_entier():
    while True:
        try:
            return int(
                raw_input( "Veuillez donner un entier: " )
            )
        except ValueError:
            print "Vous n'avez pas donné un entier."

entier = demander_un_entier()
print entier
```

Testez le programme. Que fait ce programme ? Sachant que la fonction `float` convertit une chaîne de caractères en un réel, écrivez une fonction `demandeur_un_reel()`.

2) Dans la paquet `random`, il y a une fonction `randint(a,b)` qui prend en paramètres deux entiers `a` et `b` et qui renvoie un entier au hasard compris entre `a` et `b`.

Écrire une fonction qui prend trois entiers `a`, `b` et `n` en paramètres et qui retourne un tableau contenant `n` entiers au hasard compris entre `a` et `b`.

Écrire une fonction qui prend un entier `n` et deux réels `a` et `b` en paramètres et qui retourne un tableau contenant `n` réels compris entre `a` et `b`.

3) Créez un module `util` dans lequel vous enregistrerez toutes les fonctions de l'exercice.

Exercice 4: Moyenne

1) Écrire un programme python qui prend une liste de réels en paramètres et qui renvoie la moyenne de ces entiers.

2) Testez votre programme en utilisant des tableaux contenant des entiers aléatoires.

Exercice 5: Nombres premiers

- 1) Écrire une procédure `primale(n)`, prenant en entrée un nombre entier `n`, et qui renvoie la valeur `True` (vrai) ou `False` (faux) selon que `n` est premier ou non.
- 2) Écrire un programme qui prend en paramètre un entier `n` et qui renvoie sous forme de liste, les `n` premiers entiers premiers.

Exercice 6: Le crible d'Ératosthène

Le crible d'Ératosthène est un procédé qui permet de trouver tous les nombres premiers inférieurs à un certain entier naturel donné `n`. L'algorithme procède par élimination : il s'agit de supprimer d'une table des entiers de 2 à `n` tous les multiples d'un entier. En supprimant tous les multiples, à la fin il ne restera que les entiers qui ne sont multiples d'aucun entier, et qui sont donc les nombres premiers.

On commence par rayer les multiples de 2, puis à chaque fois on raje les multiples du plus petit entier restant.

On peut s'arrêter lorsque le carré du plus petit entier restant est supérieur au plus grand entier restant, car dans ce cas, tous les non-premiers ont déjà été rayés précédemment.

À la fin du processus, tous les entiers qui n'ont pas été rayés sont les nombres premiers inférieurs à `n`.

- 1) Sur papier, appliquez le crible d'Ératosthène pour déterminer les nombres premiers inférieurs à 30.
- 2) Implémenter une fonction qui prend en paramètre un entier `n` et qui renvoie la liste des entiers inférieurs à `n` en utilisant le crible d'Ératosthène. Pour cela, vous utiliserez une liste `L` de taille `n+1`. Dans cette liste, `L[i]` vaudra `True` si `i` est un nombre premier, `False` si `i` n'est pas un nombre premier et `None` si l'algorithme n'a pas encore déterminé la primalité de `i`.

Exercice 7: Les nombres de Catalan

Les nombres de Catalan sont définis par la formule :

$$C_n = \begin{cases} 1 & \text{si } n = 0 \\ \sum_{i=0}^{j=n-1} u_i \cdot u_{n-1-i} & \text{si } n > 0 \end{cases}$$

Écrire une fonction qui prend en paramètre un entier `N` et qui renvoie les `N` premiers nombres de Catalan.