UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

# Outils Info. - TD : Feuille 4

## Exercice 1 – Accumulateur dans liste

Toutes les questions de cet exercice vont suivre le même modèle. Le programme va d'abord créer une liste vide (appelée par exemple `resultat`), puis parcourir les valeurs d'une liste existante (par exemple appelée `entree`) et ajouter des valeurs à `resultat` (en utilisant la fonction `resultat.append(...)`). À la fin le programme doit afficher la liste `resultat`.

**Q1)** Écrire un programme `lescarres.py` qui crée une liste avec les nombres donnés suivants `10, 1.414, 9, 10.10` et calcule une nouvelle liste contenant les carrés de ces nombres. À la fin, le programme doit afficher la liste `resultat` directement avec `print()` (vous pouvez aussi afficher sa somme avec la fonction `sum()`).

**Q2)** Écrire un programme `leslogs.py` qui crée une liste avec les nombres donnés suivants `100, 1.414, 9, 10.10` et calcule une nouvelle liste contenant les logarithmes de ces nombres. On utilisera le logarithme en base 10, qui peut être calculé grâce à la fonction `log10` du module `math`.

**Q3)** Écrire un programme `tousfloat.py` qui crée une liste avec les chaînes de caractères suivantes `"100", "1.414", "9", "10.10", "999"` et calcule une nouvelle liste contenant des réels obtenus à partir de l'interprétation de ces chaînes de caractères.

## Exercice 2 – Technique de la fonction principale

Reprendre chacune des questions de l'exercice précédent, sans utiliser de variables globales. Aide : utiliser la « technique de la fonction principale ».

## Exercice 3 – Fonctions et listes

**Q4)** Écrire une fonction `liste_carres` qui reçoit un paramètre (supposé être une liste de nombres) et renvoie une nouvelle liste contenant les carrés de ces nombres. NB : la fonction ne doit rien afficher.

**Q5)** Écrire une fonction `liste_floats` qui reçoit un paramètre (supposé être une liste de chaînes de caractères) et renvoie une nouvelle liste contenant les nombres (flottants) correspondants. NB : la fonction ne doit rien afficher.

**Q6)** Écrire un programme (sans variables globales) qui utilise ces deux fonctions et la fonction `split` (entre autres) : le programme doit demander à l'utilisateur de taper une liste de nombres séparés par des espaces, et afficher la liste de leurs carrés.

**Q7)** Écrire une fonction `input_floats` qui ne reçoit aucun paramètre et demande à l'utilisateur de taper une liste de nombres séparés par des espaces et renvoie la liste des nombres correspondants.

**Q8)** Écrire une fonction `tous_sauf_un` qui reçoit une liste et renvoie une nouvelle liste contenant exactement les mêmes éléments mais sans le premier (il y a donc un élément de moins dans la liste renvoyée).

**Q9)** Écrire une fonction `un_sur_deux` qui reçoit une liste et renvoie une nouvelle liste contenant deux fois moins d'éléments, en en prenant un sur deux (le premier, puis le troisième, …).

**Q10)** Écrire un programme (sans variables globales) en réutilisant au mieux vos fonctions. Le programme doit d'abord demander à l'utilisateur de taper une liste qui commence par un « ordre » et se fini par un nombre quelconque de réels (le tout séparé par des espaces). Si l'on considère que les nombres entrés sont $\{x_i\}_{i=1}^{N}$, le programme doit au afficher :

Si l'ordre est `somme`, la valeur de $\sum_{i=1}^{N} x_i$.

Si l'ordre est `norme`, la valeur de $\sqrt{\sum_{i=1}^{N} x_i^2}$.

Si l'ordre est `alt`, $\sum_{i=1}^{N/2} x_{2i-1} - \sum_{i=1}^{N/2} x_{2i}$ (c'est à dire $(x_1 + x_3 + x_5 + ...) - (x_2 + x_4 + x_6 + ...)$).

## Exercice 4 – Qtido et Jeux de Dames

La feuille de TD 2 liste les fonctions disponibles dans la bibliothèque qtido. La bibliothèque qtido contient aussi deux fonctions `est_souris` et `coordonnees_souris` pour manipuler les événements souris (clicks).

Voici un exemple simple qui trace un petit damier et affiche (avec print) les coordonnées de la souris quand l'utilisateur clique (ici, sans jamais réafficher la fenêtre) :

```
from qtido import *

f = creer(250, 250)

for j in range(5):
    for i in range(5):
        if (i+j)%2 == 0:
            couleur(f, 1, 0, 0)
        else:
            couleur(f, 0, 0, 1)
        rectangle(f, i*50, j*50, i*50+50, j*50+50)

while not est_fermee(f):
    attendre_evenement(f, 100) # un max de 100 milli secondes
    e = dernier_evenement(f)
    if est_souris(f, e, "PRESS"):     # PRESS pour l'appui sur le bouton de la souris
        xy = coordonnees_souris(f, e)
        print("Click en x =", xy[0], "et y =", xy[1])
```

**Q11)** Reprendre le programme donné ci-dessus mais en supprimant toutes les variables globales.

Nous allons maintenant faire un jeu de dames.

**Q12)** Écrire une fonction `damier_vide` qui reçoit une fenêtre en paramètre et affiche un damier blanc/beige et noir/marron de 10 par 10 cases de taille 50x50.

Nous allons maintenant afficher le damier. L'affichage sera fait en boucle, comme pour réaliser une simulation, bien que dans les premières questions l'affichage ne change pas au cours du temps.

Nous rappelons ici la structure typique d'un programme de simulation (où les choses entre «» sont à remplir selon la simulation) :

```
# Imports des modules nécessaires
...
# Définitions de fonctions utilitaires
...

def principale():
    # initialisation de l'état du système
    «...»

    while not «condition-de-fin»:
        # affichage de l'état du système
        «...»

        # attente d'un événement (ou du temps)
        «...»
        # mise à jour de l'état du système
        «...»

principale()
```



**Q13)** Écrire un programme qui affiche en boucle le damier vide, dans une fenêtre définie avant avec la bonne taille.

Nous voulons maintenant représenter des pions sur le damier (les pions sont sur les cases noires). Il y a deux types de pions (deux joueurs), les pions rouge et les pions bleus. Au départ, il y a deux lignes de pions bleus en haut et deux lignes de pions rouges en bas.

**Q14)** Quelle information faut-il pour pouvoir afficher les pions sur le plateau ? Autrement dit, quel est l'état du système ?

**Q15)** Comment représenteriez-vous cet état dans votre programme Python ? (opt: 2 réponses possibles)

**Q16)** Quel est l'état initial du système ? Comment et où écririez vous en Python l'initialisation du système ?

**Q17)** Écrire et utiliser une fonction `pions` qui affiche les pions données (état) dans une fenêtre donnée.

**Q18)** Faire que quand on clique sur une case non-vide, le pion disparaisse grâce à une modification de l'état du système.

**Q19)** Faire que quand on clique sur une case non-vide, le pion disparaisse grâce à une modification de l'état du système mais que l'on se rappelle aussi de la couleur du pion retiré (on saisi le pion).

**Q20)** Faire que quand on clique *alors* sur une case vide on ajoute un nouveau pion (on repose le pion).

**Q21)** (défi) Essayer de faire que les règles du jeu de dames soient appliquées et vérifiées.