

Modélisation et robotique

Exercices 5

1 Révisions du cours

Question A. Prédire ce qui sera affiché puis vérifier dans le REPL.

1. `for i in range(10):`
 `print(i)`
2. `L = [1, 2, 'Pim', 'Pam', '-2']`
 `for i in range(L):`
 `print(i)`
3. `L = [1, 2, 'Pim', 'Pam', '-2']`
 `for i in range(len(L)):`
 `print(i)`
4. `for i in range(10):`
 `print(L[i])`
5. `print ('1' + '1')`
6. `print ('1' + 1)`

Question B. Triangle. Dire ce qui sera affiché.

```
triangle = [[10, 10], [20,20], [10, 20]]
for sommet in triangle:
    print(sommet)

for [x,y] in triangle:
    print(x)

print(zip(*triangle))
print(zip(*(zip(*triangle))))
print(zip(*[range(1,4), ['Tommie Smith', 'Peter Norman', 'John Carlos']]))
```

2 Petits programmes

Question C. Sommer les entier de la liste. Définir une fonction qui prend en entrée une liste et retourne la somme des entiers de la liste.

Question D. Concaténer les chaînes de caractère. Définir une fonction qui prend en entrée une liste et retourne la chaîne de caractère obtenue en mettant bout à bout toutes les chaînes de caractères contenues dans la listes (dans l'ordre naturel).

Question E. Entiers, chaînes et listes. Définir une fonction qui prend en entrée une liste et retourne un triplet contenant la somme des entiers de la liste, la concaténation de toutes les chaînes de la liste et la liste formée en joignant toutes les listes de la liste.

Question F. Barycentre. Définir une fonction barycentre qui prend en entrée une liste de points (chaque point est une paire de coordonnées (x,y)) et retourne le point barycentre de la liste.

Question G. Centre de gravité. Même question mais cette fois ci la liste contient des triplets (x, y, poids) et on veut déterminer leur centre de gravité.

Question H. Énumérer. Soit la liste `L = [1, 2, 'Pim', 'Pam', '-2']`. Créer le code qui affichera :

```
l'element 0 de la liste est 1
l'element 1 de la liste est 2
l'element 2 de la liste est Pim
l'element 3 de la liste est Pam
l'element 4 de la liste est -2
```

De trois façon différentes. En utilisant `len`, en utilisant un compteur de boucle, puis en utilisant `zip`.

3 Problèmes

Mettre à jour votre machine, puis lancer une session python (*Spyder*) et créer un nouveau fichier qui commencera par ces lignes :

```
try:
    initialisation
except NameError: # premiere execution dans ce REPL
    initialisation = True
    from gardenworld import *
    from game0 import *
    init('tiny_complete')
    orientation = 2
    xcoord = 9
    ycoord = 0
```

Assurez vous dans les options d'exécution(menu exécution, configurer...) que la console choisie soit bien : « exécuter dans la console python ou ipython active ».

Vous allez définir vos fonctions dans le fichier mais vous n'y mettez que les définitions de fonctions et rien d'autre. En particulier, vous ferez les appels de fonctions dans la console (un REPL).

Pour tester définissez une fonction `demitour` dans le fichier, lancez l'exécution du fichier avec l'icône vert en forme de triangle (ou en tapant F5) de façon à rendre la fonction disponible dans la console et testez là en faisant appel à votre fonction dans la console.

Question I. Orientation absolue. Maintenant définissez des fonctions `tgo()`, `tdo()`, `quelle_orientation()` comme dans le TD 4 qui maintiendront la connaissance de l'orientation absolue dans la variable globale `orientation` que vous avez défini plus haut. Redéfinissez la fonction `demitour` de façon à ce que, elle aussi, préserve l'orientation. Mais au fait où est le Nord (utiliser `quelle_orientation`).

Question J. Position en coordonnées absolues. Les variables globales `xcoord` et `ycoord` doivent servir à connaître les coordonnées absolues de notre héros. Le centre du plateau a pour coordonnées (0, 0).

Définir une fonction `avxy`, qui remplacera `av` dans toutes vos nouvelles fonctions, de telle sorte que les coordonnées resteront exactes à chaque déplacement.

Question K. Aller à. Définir une fonction `aller_a` qui reçoit en entrée une paire de coordonnées `xbut`, `ybut` et déplace le héros de sa position actuelle à la position `xbut`, `ybut`.