

Langage Python A3 TD1

Cette séance, dont le contenu TD est léger, peut commencer par découvrir l'IDE spyder (via anaconda) et parcourir les menus et toolbox.

Reprendre les slides du CMO1 et expérimenter sur l'interpréteur.

Pas de notebook pour le moment, il faut absolument travailler sur un IDE offrant un interpréteur et un éditeur de script.

Pour ce TD, et pour chaque TD, vous devriez réaliser deux dépôts sur DVO.

- **Le premier dépôt à la fin de la séance du TD**
- **Le deuxième dépôt avant la prochaine séance de TD (avant et non pas pendant la séance)**

Le format de dépôt : TD1_Nom_Prenom.py.txt

Attention : il faut absolument ajouter l'extension txt à votre fichier python, pour que DVO ne le traite pas comme fichier dangereux ou indésirable et surtout pour pouvoir le lire (en preview) sans être contraint à le télécharger.

Exo 0 : Votre code doit avoir la forme suivante:

```
# %% zone de l'exo zéro elle doit contenir l'Exo 0 et les méthodes afférentes
def Exo0():
    """
    Il s'agit d'une méthode à titre d'exemple
    l'instruction help(Exo0) vous retourne cette documentation
    grâce à l'utilisation des triples guillemets
    ou bien dans la console de spyder saisissez Exo0?
    """
    print("voila mon Exo de bienvenue")

# %% zone de l'exo1
def Exo1():
    print("Exo1")

# %% zone du main

if __name__ == '__main__':
    #Exo0()
    Exo1()
```

Exo 1: Affectez les variables temps et distance par les valeurs 6.892 et 19.7.

- Calculez et affichez la valeur de la vitesse.
- Améliorez l'affichage en imposant deux chiffres après le point décimal.

Exo 2: Écrire un programme calculant le maximum et le minimum de deux nombres.

- avec un test à une alternative
- avec un test simple
- avec un test ternaire

Exo 3: Ecrire la fonction **volBoite(x1,x2,x3)** de manière à ce qu'elle puisse être appelée avec un, deux, ou trois arguments.

- 1 argument : la boîte est considérée comme cubique (l'argument étant l'arête de ce cube).

- 2 arguments : la boîte est considérée comme un prisme à base carrée. (Dans ce cas le premier argument est le côté du carré, et le second la hauteur du prisme).

- 3 arguments : la boîte est considérée comme un parallélépipède.

Par exemple :

```
print volBoite() # donne -1 ou None (-> erreur).
print volBoite(5.2) # donne 140.608
print volBoite(5.2, 3) # donne 81.12
print volBoite(5.2, 3, 7.4) # donne 115.44
```

Exo 4: Définissez une fonction **eleMax(liste,debut,fin)** qui renvoie l'élément ayant la plus grande valeur dans la liste transmise. Les deux arguments **debut** et **fin** indiqueront les indices entre lesquels doit s'exercer la recherche, et chacun d'eux pourra être omis (cf. exercice précédent).

Exemples de la fonctionnalité attendue :

```
>>> serie = [9, 3, 6, 1, 7, 5, 4, 8, 2]
>>> print(eleMax(serie))
9
>>> print(eleMax(serie, 2, 5))
7
>>> print(eleMax(serie, 2))
8
>>> print(eleMax(serie, fin=3, debut=1))
6
```

Exo 5: Soient les listes suivantes :

```
t1 = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
t2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet', 'Août', 'Septembre', 'Octobre',
'Novembre', 'Décembre']
```

Écrivez une fonction qui retourne une nouvelle liste t3.

Celle-ci contiendra tous les éléments des 2 listes en les alternant :

```
t3 = ['Janvier', 31, 'Février', 28, 'Mars', 31, 'Avril', 30, etc...]
```

Exo 6: Ecrivez une fonction qui recopie une chaîne de caractères (dans une nouvelle variable) en l'inversant. Exemple : "zorglub" devient "bulgroz"

Exo 7: Que fait cette fonction? testez-là

```
def mafonction(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

Exo 8: Télécharger le fichier tableau.dat depuis DVO, ouvrez le avec un éditeur de texte pour avoir une idée sur le contenu (puis fermer le). chargez le contenu dans une liste d'entiers (par code python).

Implémentez le tri fusion (`def TriFusion(myList):`)

tester votre code en comparaison avec la méthode de l'exo 7.