

Exercice 1

1. Définir en Python deux listes L4 et L6 vérifiant:
 - L4 est la liste formée des dix premiers entiers non nuls multiples de 4 supérieurs ou égaux à 1.
 - L6 est la liste formée des dix premiers entiers non nuls multiples de 6 supérieurs ou égaux à 1.
2. Effectuer la concaténation des deux listes : L4+L6 .
3. Que contient la liste 2*L4 ?
4. Ecrire une fonction `sum_liste` qui renvoie la liste `Lst_som` obtenue en ajoutant terme à terme et dans l'ordre les éléments des deux listes L4 et L6 .

Exercice 2

En langage Python , il existe une méthode de liste nommée `count(x)` qui compte le nombre d'occurrences de l'élément x dans la liste .Par exemple:

```
L=[1,2,4,1,5,1,6]
>>> L.count(1)
3
>>> L.count(9)
0
```

Sans utiliser la méthode `count(x)` , écrire une fonction `compter(liste,x)` qui prend pour paramètres une liste et une valeur x et qui renvoie le nombre d'occurrences de x dans liste.

Exercice 3

Ecrire une fonction `Echanger(L,i,j)` qui prend pour paramètres une liste L et deux entiers naturels i et j et qui renvoie une liste où les éléments d'indices i et j de L, s'ils existent, ont été permutés.

Exercice 4

On donne la liste `Points = [[2, 1], [5, 10], [1, 11], [4, 5], [4, -1]]`

Chaque élément de la liste `Points` est une liste représentant les coordonnées d'un point dans le plan muni d'un repère.

1. Ecrire une fonction `f1` qui renvoie la liste des points de la liste `Points` situés en dessous de la droite d'équation $y = x$.
2. Même question pour les points situés en dessus de la droite d'équation $y = 3x - 1$

Exercice 5

1. Ecrire une fonction *pluriels* qui a pour paramètres une liste de mots et qui renvoie une liste contenant les éléments (chaînes de caractères) se terminant par la lettre 's'.
2. Ecrire une fonction *voyelles* qui a pour paramètres une liste T et qui renvoie une liste contenant les éléments de T qui contiennent au moins deux voyelles.

Exercice 6

1. Ecrire une expression Python dont la valeur est la liste des entiers pairs de 2 à 22 compris.
2. Ecrire une fonction `liste_pairs(a,b)` dont les paramètres a et b sont des entiers pairs et qui renvoie la liste des entiers pairs de a à b compris.
3. Ecrire une fonction de paramètre n qui renvoie une liste de n entiers pris au "hasard" entre a et b .

Exercice 7 (d'après meef-nsi Lyon)

Écrire une fonction Python qui prend une liste d'entiers en paramètre et retourne un couple (2-uplet) de listes, la première étant la sous-liste des entiers pairs, et l'autre la sous-liste des entiers impairs.

Exercice 8

Dans cet exercice on utilise le module PIL de Python

```
from PIL import Image

# Créer une image de 600x600 pixels avec un fond blanc
img = Image.new('RGB', (600, 600), (255, 255, 255))

#pour dessiner un pixel de couleur rouge au point de coordonnées (x,y)
img.putpixel( (x, y), (255, 0, 0) )

#pour afficher l'image
img.show()
```

Pour dessiner un rectangle , on crée une bande rectangulaire dont le sommet haut-gauche a pour coordonnées le tuple (a,b) et pour dimension la largeur u et la hauteur v.



1. Ecrire une fonction **rectangle** qui prend pour paramètres (a,b,u,v) qui dessine un rectangle de dimensions u*v dont le sommet haut-gauche a pour coordonnées (a,b) . La fonction devra renvoyer une liste contenant tous les pixels (x,y) (2-uplets) du rectangle.

Pour afficher à l'écran le rectangle créé à la question précédente , on utilise la fonction **dessiner**(fig,coul) qui prend pour paramètre une liste de pixels (nommée fig) et un 3-uplet **coul** .

Par exemple , pour afficher à l'écran un rectangle **rectA** avec la couleur (255,200,10) , on appelle :

```
dessiner(rectA, (255,200,10))
```

2. Ecrire le code de la fonction **dessiner**.

Exercice 9

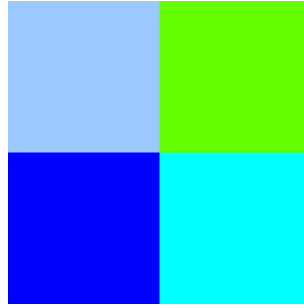
On peut créer une copie d'une figure (comme un rectangle tel que décrit dans l'exercice précédent) .

On écrit pour cela une fonction **translater**(T,a,b) T désignant la liste des pixels d'une figure et où le tuple (a,b) représente les coordonnées du vecteur de la translation.

1. Compléter le code de la fonction **translater**

```
def translater(T , a , b ):
    '''
    T : une liste de tuples (les pixels)
    a ,b : le vecteur de la translation
    '''
    T1=[ ]
    for point in T:
        .....
    return T1
```

On veut juxtaposer quatre carrés comme sur la figure ci-dessous.



2. Ecrire un script qui réalise une telle figure en utilisant les fonctions `rectangle` et `translater` définies ci-dessus.

Afficher à l'écran une telle figure à l'aide de la fonction `dessiner`.

(Il n'est pas nécessaire de respecter les mêmes couleurs pour les carrés de la figure.)

3. Ecrire une fonction `dessiner_T` qui construit la lettre T comme sur la figure ci-dessous.



4. Construire et afficher de la même manière les lettres majuscules **H** et **E** en respectant la même hauteur que pour la lettre **T** construite précédemment.
5. Rédiger un code qui affiche le mot **THE** .