



Avant toute chose, créez un jupyter notebook intitulé : *cc1\_nom\_prenom.ipynb*. Vous avez le droit d'utiliser le cours, les corrections d'exercices **Il est interdit d'effectuer des recherches google et d'utiliser une quelconque IA générative telle que ChatGPT. Tout manquement, entrainera directement une note de 0.**

### Exercice 1

Pour chacun des programmes suivants, donnez les valeurs de chaque variable à la fin de l'exécution du programme, et indiquez ce qui s'affiche.

```
x,y=3.5,5
z=x/y
p=7

if (x>3)and(z*10!=p):
    print("Blue")
elif (p>y)or(x>4):
    print("Red")
else:
    print("Black")
```

```
i,j,k=1,2,3
x,y,z=1.1,1.2,1.3
i=i+x
j=j+y
k=k+z
print(i,j,k,x,y,z)
if (i+j>k)and(j-i==x):
    print("Blue")
else:
    print("Red")
```

```
def f1(x,y):
    if (x<y):
        return 3
    else:
        return 4

def f2(n):
    return 2*n-1

x=3.4
y=5.3
print(f2(f1(y,x)))
```

### Exercice 2

Écrivez une fonction qui permet de trouver tous les diviseurs d'un nombre "n" entre 2 et 100.

- Le programme doit demander à l'utilisateur d'entrer un nombre entier naturel "n".
- Une fois le nombre saisi, le programme renvoie tous les diviseurs de "n" dans une liste.

### Exercice 3

Soit le polynôme P défini par  $P = x^3 - 10x^2 - 2610x + 84$

Ecrivez une fonction qui, étant donné un entier naturel x, retourne la valeur de ce polynôme pour x.

Ecrivez un programme qui affiche la plus petite valeur prise par ce polynôme pour les entiers appartenant à l'intervalle [-50, 50].

### Exercice 4

Créez une structure de données dans lequel :

- les clés sont les prénoms de différentes personnes. (Il n'y a aucun prénom en double)
- les valeurs sont les mois de naissance de ces personnes stockées sous forme de nombres entiers (1 pour janvier, ... 12 pour décembre)

Vous devez écrire une fonction anniversaire prenant en arguments le dictionnaire créer ci-dessus ainsi qu'un numéro d'un mois et renvoyant une liste contenant les prénoms des personnes nées durant ce mois.

### Exercice 5

Développer une version simplifiée du jeu de démineur en utilisant Python.

Étape 1: Initialisation du Champ de Mines



Créer une grille 2D pour représenter le champ de mines. Chaque cellule peut être une mine (\*) ou non (0). Utilisez une taille de grille prédéfinie (par exemple, 10x10) et un nombre fixe de mines. Placez les mines de manière aléatoire.

Étape 2: Calcul des Mines Adjacentes

Écrire une fonction qui, pour chaque cellule, calcule le nombre de mines dans les cellules adjacentes (les cellules autour d'une case) et leur position. Faites attention aux bords de la grille pour éviter les erreurs d'indexation.

### Exercice 6

Au Scrabble, chaque lettre possède une valeur et le score d'un mot est la somme des valeurs des lettres qui le compose.

**VALEURS = {**

**'A' : 1, 'B' : 3, 'C' : 3, 'D' : 2, 'E' : 1, 'F' : 4, 'G' : 2, 'H' : 4,**

**'I' : 1, 'J' : 8, 'K' : 10, 'L' : 1, 'M' : 2, 'N' : 1, 'O' : 1, 'P' : 3,**

**'Q' : 8, 'R' : 1, 'S' : 1, 'T' : 1, 'U' : 1, 'V' : 4, 'W' : 10, 'X' : 10,**

**'Y' : 10, 'Z' : 10**

**}**

Par exemple, la valeur du mot : GIRAFE est :  $2+1+1+1+4+1=10$ .

Écrire une fonction `score_scrabble` qui prend en paramètre une chaîne de caractères mot et renvoie sa valeur au Scrabble. Le mot ne doit comporter que les lettres de l'alphabet en majuscules et il peut être vide. La valeur des lettres de l'alphabet est donnée sous la forme d'un dictionnaire.

Voici des exemples à tester :

```
assert score_scrabble("KAYAK") == 32
```

```
assert score_scrabble("INFORMATIQUE") == 23
```