

# TP1: Introduction à python

CODA 2 eme année – Python – 2025

*Il existe plusieurs façon pour faire certains exercices, pratiquez les différentes syntaxes Python*

Lists :

A partir du code suivant :

```
numbers = [ 1, 3, 4, 5, 6, 7, 8, 9, 1 ,2 ,3]
```

1 – Additionner tout les chiffres du tableau et afficher la somme

2 – Créer un sous tableau avec tout les chiffres impairs

3 – A partir du tableau number, créer un nouveau tableau en multipliant tout les chiffres par 2

4 – Creer un tableau avec tout les chiffres de 1 à 100

4 – Créer un tableau avec 100 chiffres aléatoires entre 1 et 100

5 – En utilisant la fonction random.sample, créer un tableau avec 10 chiffre aléatoire entre 1 et 100 sans avoir de doublons.

Sets :

A partir du code suivant :

```
a = {1, 2, 3, 4, 5}  
b = {4, 5, 6, 7}
```

1 – Comment effectuer les actions suivantes sur un set

- Ajouter l'élément 8
- Retirer l'élément 2
- Vérifier que l'élément 3 se trouve dans un set

2 - Comment obtenir la différence entre 2 sets ?

3 – A partir du code suivant :

```
numbers = [ 1, 3, 4, 5, 6, 7, 8, 9, 1 ,2 ,3]
```

Récréer un set à partir de ce tableau en supprimant tout les doublons

## Dictionnaires :

1 – A partir du code suivant :

```
prices = {"apple": 1.5, "banana": 1.2, "orange": 2.0}
```

- Afficher la liste des nom et prix des produits dans la console
- Ajouter un nouveau produit dans le dict
- Calculer la somme des prix

2 - A partir du code suivant :

```
prices = {"apple": 1.5, "banana": 1.2, "orange": 2.0}  
cart = {"apple": 2, "banana": 3, "orange": 1}
```

Calculer le total du panier

2 – A partir de la string suivante

```
sentence = "apple banana apple orange banana apple"
```

Écrire un code qui compte les occurrences de chaque mot et les mets dans un dictionnaire

## Fonctions :

1 – Créer une fonction qui prend en paramètre un mot et qui retourne un booléen pour indiquer si le mot est un palindrome

2 – La suite de Fibonacci est une suite de nombres entiers dans laquelle chaque nombre est la somme des deux nombres qui le précédent. Elle commence par les nombres 0 et 1

Créer une fonction qui prend en paramètre un nombre n et qui génère les n premiers nombres de la suite de Fibonacci.

## OOP :

1 - Créer une classe Vector2D qui possède 2 propriétés x et y.

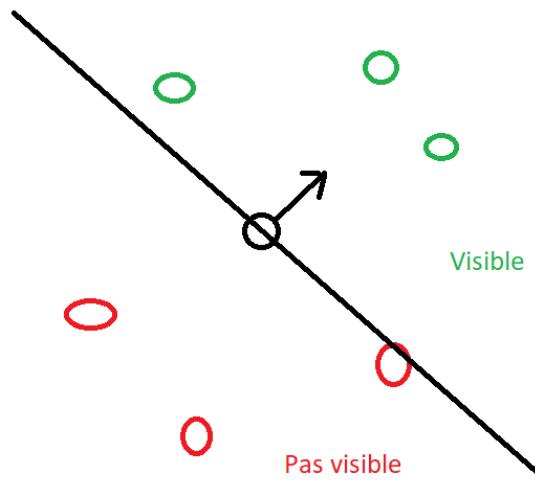
2 – Créer une classe Player qui possède les propriétés suivantes :

Un nom

Une position(Vector2D)

Une direction (Vector2D)

3 – On souhaite maintenant coder un algorithme qui permet de savoir si un joueur est visible par un autre



Un joueur est visible si celui ci est en face de lui dans un rayon de 180 degré. Pour ce faire, on peut se servir le produit scalaire.

Une des formules du produit scalaire est la suivante

$$A \cdot B = A_x \times B_x + A_y \times B_y$$

Ou A et B sont 2 vecteurs dans un espace 2D (x,y)

En fonction du produit scalaire on peut en déduire les choses suivantes :

si  $A \cdot B > 0$  : B est devant A

si  $A \cdot B < 0$  : B est derrière A

si  $A \cdot B == 0$  : B est à 180 degré de A

Créer une méthode is\_visible dans la classe Player, qui prend un autre Player en paramètre et qui retourne un booléen pour indiquer si le 2 eme player est visible ou non.

Remarque : Vous aurez besoin de calculer le Vecteur entre 2 joueurs. Ou mettre cette méthode ?

4 – Créer une méthode are\_visible qui prend en paramètre une list de Player et qui retourne la list des Player visibles.

Bonus : Trouvez un moyen de gérer un FOV (field of view) et une distance de vision maximum