

QUIZZ 1

QUIZZ 1

1. On considère la liste `L = [3, 4, 1, 2, 5]`. Que vaut `L[-1]` ?
2. En utilisant une boucle, créer une liste contenant 10 zéros.
3. Écrire une fonction `minimum()` qui prend en paramètre une liste de nombres et qui renvoie la valeur minimal de ces nombres. On n'utilisera pas la fonction `min()` de Python.
4. Écrire une fonction `position_minimum()` qui prend en paramètre une liste de nombres et qui renvoie la position de la valeur minimale de ces nombres.

QUIZZ 2

QUIZZ 2

1. En utilisant une boucle, créer la liste des premiers cubes jusqu'à 125 inclus : `[0, 1, 8, 27, 64, 125]`.

QUIZZ 2

1. En utilisant une boucle, créer la liste des premiers cubes jusqu'à 125 inclus : `[0, 1, 8, 27, 64, 125]`.
2. Créer la même liste par compréhension.

QUIZZ 2

1. En utilisant une boucle, créer la liste des premiers cubes jusqu'à 125 inclus : `[0, 1, 8, 27, 64, 125]`.
2. Créer la même liste par compréhension.
3. Écrire une fonction `moyenne_ponderree(notes, coefficients)` qui prend en paramètres :
 - une liste de notes `notes`;
 - une liste de coefficients `coefficients`,et qui renvoie la moyenne pondérée des notes pondérées par les coefficients.

QUIZZ 3

QUIZZ 3

1. On considère la liste ci-dessous :

```
liste_1 = [  
    ["Maryam", "Alexandre", "Jacques"],  
    ["Joseph", "Michael", "Jean-Pierre"],  
    ["Pierre", "Leonhard", "Henri"],  
    ["David", "Paul", "Pierre-Simon"]  
]
```

- Que vaut `liste_1[3][2]`?
- Comment peut-on obtenir, de manière similaire, "Paul"?

2. En utilisant une ou plusieurs boucles, créer la liste suivante :

```
liste_2 = [  
    [1, 2, 3, 4, 5, 6, 7, 8],  
    [2, 4, 8, 10, 12, 14, 16],  
    [3, 6, 9, 12, 15, 18, 21],  
    [4, 8, 12, 16, 20, 24, 28, 32]  
]
```

QUIZZ 4

QUIZZ 4

1. Écrire une fonction `contient_valeur()` qui prend en paramètres une liste d'entiers et un entier `n`, et qui renvoie **True** si `n` figure dans la liste, **False** sinon.
2. Écrire une fonction `positions_valeur()` qui prend en paramètres une liste d'entiers et un entier `n`, et qui renvoie la liste des positions auxquelles `n` apparaît dans la liste (la fonction renvoie une liste vide si l'entier `n` n'apparaît pas dans la liste).

3. Préciser liste après les lignes suivantes :

```
liste = []  
for i in range(3):  
    ligne = []  
    for j in range(4):  
        ligne.append(i - j)  
    liste.append(ligne)
```