

# Evaluation cours « Programmation Python »

## Prérequis:

- Python en version >3.
- Utilisation d'un éditeur de texte ou de l'idle, à votre convenance.
- Durée : 2 Heures
- Accès documentation, internet : Autorisé
- Accès IA type chatgpt : Non Autorisé

## Engagement d'intégrité :

- En commençant cet examen, vous vous engagez à ne pas utiliser chatgpt ou toute autre outil similaire .

## Rendu:

- Pour chaque étudiant, 1 fichier au format [Nom\\_Prenom.zip](#) est attendu. Celui-ci devra contenir 2 fichiers correspondant chacun à :
  - 1 Fichier [qcm.py](#) contenant les réponses aux QCM.
  - 1 Fichier [exercices.py](#) contenant les codes des exercices.
  - Commenter chaque début d'exercice `#exercice1`, `#exercice2`, etc.
- Dépôt directement sur moodle

## Partie 1 : QCM (Estimation : 30 min)

### Question 1 :

Le code ci-dessous permet d'importer:

- A. L'entité **c** du module **b** du package **a**
- B. L'entité **a** du module **b** du package **c**
- C. L'entité **c** du module **a** du package **b**
- D. L'entité **b** du module **a** du package **c**

```
1 from a.b import c
2
3
```

### Question 2 :

Que va afficher le code suivant et pourquoi ?

```
1 x = "\\\\"
2 print(len(x))
3
```

### Question 3 :

Lesquelles des affirmations suivantes sont vraies à propos du code ci-dessous?

- A. **nums** à la même longueur que **vals**
- B. **nums** et **vals** sont des noms différents pour la même liste
- C. **nums** et **vals** sont des listes différentes
- D. **vals** est plus long que **nums**

```
4
5 nums = [1, 2, 3]
6 vals = nums
7
```

### Question 4 :

Quelle est la différence principale entre une liste et un tuple en Python ?

- A. Les listes sont immuables, tandis que les tuples sont mutables.
- B. B. Les tuples sont immuables, tandis que les listes sont mutables.
- C. C. Les listes peuvent contenir uniquement des types de données identiques, tandis que les tuples peuvent contenir différents types de données.
- D. D. Il n'y a aucune différence; les termes sont interchangeables.

## Question 5 :

Quelle est la différence entre break et continue dans une boucle en Python ?

- A. break termine la boucle, tandis que continue passe à l'itération suivante de la boucle.
- B. break passe à l'itération suivante, tandis que continue termine la boucle.
- C. Il n'y a pas de différence; ils peuvent être utilisés de manière interchangeable.
- D. break est utilisé dans les boucles for, et continue dans les boucles while.

## Question 6 :

Comment vérifier si une clé existe dans un dictionnaire ?

- A. "key" in dict.keys()
- B. dict.has\_key("key")
- C. "key" in dict
- D. dict.exists("key")

## Question 7 :

Quelle situation justifie l'utilisation de plusieurs classes dans un programme orienté objet ? Justifiez

- A. Lorsque vous avez plusieurs variables que vous voulez regrouper dans une liste.
- B. Lorsque vous devez modéliser différents types d'objets ayant des comportements similaires mais des caractéristiques spécifiques.
- C. Lorsque vous voulez écrire un programme qui ne contient aucune répétition de code.
- D. Lorsque vous voulez simplement éviter d'écrire des fonctions.

## Question 8 :

Lors de la création d'un programme qui manipule des fichiers, quel type d'exceptions pourriez-vous devoir gérer ?

- A. `ValueError`, si un fichier contient des données invalides.
- B. `FileNotFoundError`, si le fichier à ouvrir n'existe pas.
- C. `PermissionError`, si vous n'avez pas les droits d'accès au fichier.
- D. Toutes les réponses ci-dessus.

## Question 9 :

Vous écrivez un programme pour analyser un fichier CSV contenant des données. Quels types d'exceptions pourriez-vous rencontrer et devoir gérer ?

- A. `FileNotFoundError`, si le fichier CSV spécifié n'existe pas.
- B. `UnicodeDecodeError`, si le fichier CSV contient des caractères encodés de manière incorrecte.
- C. `ValueError`, si les données dans le fichier ne correspondent pas au format attendu (par exemple, une chaîne de caractères dans une colonne numérique).
- D. Toutes les réponses ci-dessus.

## Question 10 :

Dans le contexte de la programmation orientée objet en Python, pourquoi utiliser des valeurs par défaut dans un constructeur (`__init__`) ?

- A. Pour éviter que l'utilisateur ait à fournir tous les arguments lors de la création d'une instance.
- B. Pour imposer des valeurs fixes et empêcher toute modification des attributs.
- C. Pour permettre la création d'une classe sans aucun attribut.
- D. Pour éviter de définir une méthode `__init__` dans la classe.

## Partie 2 : Exercices (Estimation : 60 min)

### Exercice 1 (10 min) :

Initialiser deux entiers :  $a = 0$  et  $b = 10$ .

- A. Écrire une boucle affichant et incrémentant la valeur de  $a$  tant qu'elle reste strictement inférieure à celle de  $b$ .
- B. Écrire une autre boucle décrémentant la valeur de  $b$  et affichant sa valeur si elle est impaire. Boucler tant que  $b$  n'est pas nul.

### Exercice 2 (10 min):

Écrire une fonction au nom de `supprimeDoublons()` qui supprime les doublons d'une liste saisie par l'utilisateur. Si l'utilisateur saisie `[1,2,3,2,4,4,5]`, la fonction retourne `[1,2,3,4,5]`. Ne pas utiliser la fonction `remove` de python.

### Exercice 3 (10 min):

Créer une fonction qui prend un argument ( $n$ ), correspondant à la taille d'une matrice carrée. Si  $n = 3$  alors taille de la matrice ( $3 \times 3$ ). La fonction doit afficher une matrice échiquier (des 1 et des 0 alternés).

### Exercice 4 (10 min):

On considère un mot `abcdef`, puis on construit un autre mot selon le schéma : 1ère lettre, dernière lettre, 2ème lettre, avant-dernière lettre, 3ème lettre, ...

Exemple 1 : `abcdef` -> `afbecd`

Exemple 2 : `kayak` -> `kkaay`

### Exercice 5 (20 min):

Un permis de voiture a 12 points au total.

On considère sur un parcours particulier les hypothèses suivantes :

- Conduire sous l'effet d'alcool entraîne la perte de 6 points.
- Un excès de vitesse à 20 km/h entraîne la perte d'un point.
- Un excès de vitesse entre 50 et 80km/h entraîne la perte de 2 points.

- Un excès de vitesse supérieur à 80km/h entraîne la perte de 6 points.
- Le défaut de port de ceinture entraîne la perte de 3 points.
- Un refus de priorité entraîne la perte de 4 points.

Jean a commis les infractions suivantes :

- Excès de vitesse à 20km/h
- Défaut de port de ceinture de sécurité

Et Nicolas a commis les infractions suivantes:

- Conduite sous l'effet d'alcool
- Excès de vitesse entre 20 et 30 km/h
- Refus de priorité

1. Ecrire une fonction qui prend en paramètre le type d'infraction et

retournant le nombre de points perdus

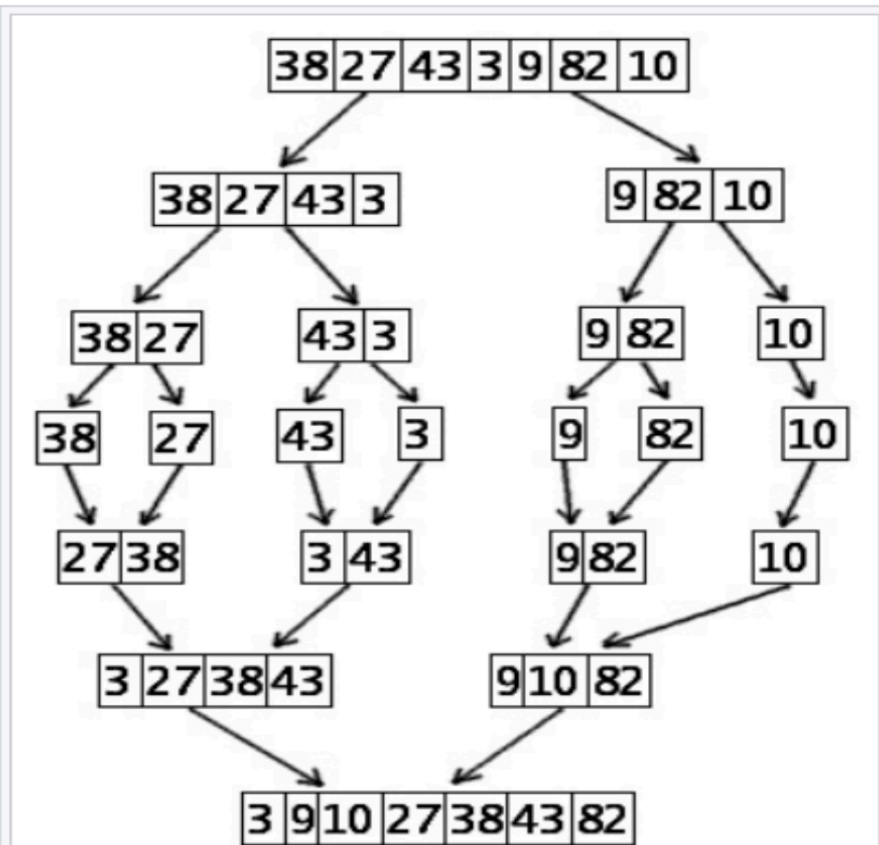
2. Trouver la meilleure façon d'afficher le nombre de points restant pour Jean et Nicolas en indiquant la perte du permis si le nombre de points est arrivé à 0.

## Exercice 6 (30 min) :

Implémentation du tri fusion en Python : Le tri fusion est un algorithme de tri comparatif qui divise récursivement la liste en deux moitiés, trie chaque moitié, puis fusionne les deux moitiés triées pour obtenir la liste triée finale. Cet algorithme utilise une approche "diviser pour régner" et est très efficace pour trier de grandes listes. Votre tâche est d'implémenter l'algorithme de tri fusion en Python.

Instructions :

- Fonction de fusion : Écrivez une fonction `merge(left, right)` qui fusionne deux listes triées `left` et `right` en une seule liste triée et la retourne..
- Fonction de tri fusion : Écrivez une fonction `merge_sort(list)` qui implémente l'algorithme de tri fusion.



Tri fusion appliqué à un tableau de 7 éléments.