

Nom, Prénom :

Quiz Python  
QZ-02

*Haute École d'Ingénierie et de Gestion du Canton de Vaud*  
*Département TIN*

Prof. Yves Chevallier

4 novembre 2025

**Consignes :**

- Écrire votre nom et votre prénom sur la première page.
- Écrire lisiblement, au stylo ou au crayon à papier gras.
- Répondre aux questions dans les espaces appropriés.
- Relire toutes vos réponses avant de rendre votre travail.
- Rendre toutes les feuilles de ce quiz.
- Aucun moyen de communication autorisé.
- Toutes les réponses concernent le langage Python 3.10.
- Cochez les questions à choix multiple avec une croix **X**.

1. Quel énoncé décrit le mieux le concept d'une blockchain ?
  - ☐ Chaque bloc contient un *hash* du bloc précédent, ce qui les relie en chaîne.
  - ☐ Chaque bloc à un lien vers ses parents et ses descendants.
  - ☐ Une blockchain est dossier spécialisé dans le stockage de fichiers.
  - ☐ La blockchain est un protocole de communication entre serveurs web.
2. Qu'appelle-t-on une transformation non bijective ?
  - ☐ Une transformation qui est à la fois injective et surjective
  - ☐ Une transformation qui n'est pas inversible car son espace d'arrivée est de dimension inférieure à son espace de départ
  - ☐ Une transformation linéaire qui conserve les distances
  - ☐ La fonction sinus ou cosinus par exemple
3. Quel est l'intérêt principal d'une fonction de hachage cryptographique comme SHA-256 ?
  - ☐ Chiffrer un message de bout en bout
  - ☐ Générer une signature de taille fixe
  - ☐ Compresser un fichier sans perte
  - ☐ Convertir un nombre en binaire
4. Quelle affirmation décrit correctement la différence entre `git merge` et `git rebase` ?
  - ☐ `git rebase` supprime l'historique, `git merge` le conserve
  - ☐ Les deux commandes produisent exactement les mêmes commits
  - ☐ `git merge` n'est disponible que sur GitHub
  - ☐ `git merge` crée un commit de fusion alors que `git rebase` réécrit l'historique pour appliquer vos commits après une autre branche
5. Dans quel cas la création d'une branche Git est-elle conseillée ?
  - ☐ Pour sauvegarder automatiquement les fichiers temporaires
  - ☐ Pour exécuter un script Python
  - ☐ Lorsqu'on démarre une nouvelle fonctionnalité ou un correctif indépendant
  - ☐ Pour supprimer rapidement des commits
6. Quel est le principe de la preuve de travail (*proof-of-work*) utilisée par Bitcoin ?
  - ☐ Les utilisateurs doivent prouver leur identité auprès d'une autorité centrale
  - ☐ Les mineurs doivent calculer un *hash* commençant par un certain nombre de zéros pour valider un bloc
  - ☐ Les transactions sont validées par un vote majoritaire non pondéré
  - ☐ Le protocole duplique chaque bloc sur plusieurs chaînes concurrentes

7. Quelle sortie affiche le code suivant ?

```
letters = ['a', 'b', 'c']  
numbers = [1, 2, 3]  
print(list(zip(letters, numbers)))
```

- ☐ ['a', 'b', 'c', 1, 2, 3]
- ☐ [( 'a', 1), ( 'b', 2), ( 'c', 3)]
- ☐ { 'a': 1, 'b': 2, 'c': 3}
- ☐ [( 'a', 'b', 'c'), (1, 2, 3)]

8. À propos de `**kwargs` et de `kwargs.get('clé', défaut)`, quelle affirmation est correcte ?

- ☐ `**kwargs` regroupe les arguments nommés dans un dictionnaire et `kwargs.get('clé', défaut)` retourne une valeur sans lever d'exception si la clé est absente
- ☐ `**kwargs` crée automatiquement des variables globales
- ☐ `**kwargs` est utilisé pour les arguments positionnels
- ☐ `**kwargs` ne peut être utilisé qu'avec des nombres

9. Un dossier peut être considéré comme un module importable en Python à condition de...

- ☐ Contenir un fichier `__init__.py` même vide
- ☐ Contenir un fichier `requirements.txt`
- ☐ Porter le même nom que le projet Git
- ☐ Contenir des fichiers `.py`

10. Comment réalise-t-on un import relatif à partir d'un module ?

- ☐ En préfixant le chemin par `import://`
- ☐ En ajoutant le chemin complet du disque dans l'instruction `import`
- ☐ En déclarant une variable globale appelée `__relative__`
- ☐ En utilisant `from .sous_module import nom` ou `from ..package import nom`

11. À quoi sert la fonction `enumerate` dans une boucle `for` ?

- ☐ Retourner à la fois l'indice et la valeur lors de l'itération
- ☐ Trier automatiquement la liste avant l'itération
- ☐ Convertir une liste en dictionnaire
- ☐ Arrêter la boucle après trois éléments

12. Quel est l'avantage principal du mot clé `with` lors de l'ouverture d'un fichier en Python ?

- ☐ Il garantit la fermeture du fichier même en cas d'exception
- ☐ Il chiffre automatiquement le contenu du fichier
- ☐ Il supprime le fichier après lecture
- ☐ Il copie le fichier dans un dossier temporaire

13. Si on souhaite surcharger l'addition pour l'objet `foo` dans le cas `42 + foo` ?

- ☐ Il faut surcharger la méthode `__add__` dans la classe de `foo`
- ☐ Il faut surcharger la méthode `__radd__` dans la classe de `foo`
- ☐ Il faut définir une fonction `add(foo)` en dehors de la classe
- ☐ Il faut utiliser la fonction intégrée `sum()`

14. Quelle est la sortie du code suivant ?

```
nums = [1, 2, 3, 4]
print(nums[1:-1])
```

- ☐ [2, 3] ☐ [1, 4]  
☐ [1, 2, 3, 4] ☐ [3]

15. Quel est l'effet d'un bloc **try ... except** bien écrit ?

- ☐ Intercepter et traiter une exception pour éviter l'arrêt brutal du programme  
☐ Accélérer l'exécution d'un programme Python  
☐ Empêcher complètement qu'une exception se produise  
☐ Convertir automatiquement une chaîne de caractères en nombre

16. Quelle est la particularité principale d'un objet **set** en Python ?

- ☐ Il ne contient pas de doublons et ne garantit pas l'ordre  
☐ Il conserve l'ordre d'insertion et autorise les doublons  
☐ Il ne peut contenir que des chaînes de caractères  
☐ Il sert uniquement à associer des clés et des valeurs

17. Quelle est la sortie de l'expression `[n * n for n in range(3)]` ?

- ☐ [1, 4, 9] ☐ [0, 1, 4]  
☐ [0, 1, 2, 3] ☐ [2, 4]

18. Que produit l'instruction `list(range(5))` ?

- ☐ [1, 2, 3, 4, 5] ☐ [5, 4, 3, 2, 1]  
☐ [0, 5] ☐ [0, 1, 2, 3, 4]

19. À quoi sert le test `if __name__ == "__main__":` dans un module Python ?

- ☐ À exécuter un bloc de code seulement lorsque le module est lancé comme script principal  
☐ À empêcher toute importation du module  
☐ À définir le point d'entrée obligatoire d'une bibliothèque  
☐ À vérifier la version de Python installée

20. Quelle affirmation décrit correctement `*args` dans la définition d'une fonction Python ?

- ☐ Il transforme automatiquement les arguments en dictionnaire  
☐ Il capture les arguments positionnels supplémentaires dans un tuple  
☐ Il impose que la fonction n'accepte qu'un seul argument  
☐ Il convertit les paramètres en chaînes de caractères