

4.2

Quiz Python
QZ-01

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

Prof. Yves Chevallier

8 octobre 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.

- C'est mieux que la dernière fois !*
- ✗ 1. Python est un langage qui a été créé par Guido van Rossum en 1991. Guido van Rossum est souvent appelé le BDFL de Python. Que signifie BDFL ?
- ☐ Big Data Friendly Language
 - ☐ Bold Decision for Life
 - ☐ Best Data Format Language
 - ☐ Basic Data Flow Language
 - ☐ Benevolent Dictator For Life
 - ☐ Built-in Data Function Library

- Une belle tentative.*
- ✓ 2. Que signifie REPL en Python ?
- ☐ Runtime Environment for Python Libraries, utilisé pour exécuter des modules Python
 - ☐ Resource Evaluation Processing Loop, principalement utilisé dans les systèmes embarqués
 - ☐ Read-Eval-Print Loop, utile pour tester rapidement des instructions Python
 - ☐ Repetitive Execution Program Library, déconseillé pour les débutants
 - ☐ Real-time Execution Programming Language, utilisé pour les applications temps réel
 - ☐ Built-in Data Function Library

- Très bonne idée, mais mal appliquée.*
- ✗ 3. Quel est le programme utilisé préférentiellement pour utiliser Python comme calculatrice en ligne de commande ?
- ☐ calc
 - ☐ ipython
 - ☐ jupyter
 - ☐ python
 - ☐ qpython
 - ☐ notebook

- Attention à la syntaxe...*
- ✓ 4. Inscrire après chaque syntaxe Python le type correspondant à l'expression.
- | | |
|-----------------------------|---------------------------------------------|
| 1. <code>[1]</code> _____ | 4. <code>(1,2)</code> _____ |
| 2. <code>{1}</code> _____ | 5. <code>(i for i in range(3))</code> _____ |
| 3. <code>{1:2}</code> _____ | 6. <code>'abc'</code> _____ |

- Très bonne idée, mais mal appliquée.*
- ✓ 5. Quel est la sortie du code suivant `"12314231423225".split("3")` ?
- ☐ `['12314231421225']`
 - ☐ `{1,2,3,4,2,3,1,4,2,3,2,2,5}`
 - ☐ `['1', '2', '4', '2', '1', '2', '25']`
 - ☐ Segmentation fault
 - ☐ `['12', '142', '142', '225']`

- Une belle tentative.*
- ✓ 6. Je souhaite importer la fonction `sqrt` sans polluer l'espace de nom courant de mon programme quelle est la bonne commande ?
- ☐ `import sqrt`
 - ☐ `using math.sqrt`
 - ☐ `from math import sqrt`
 - ☐ `from math import *`
 - ☐ `from sqrt import math`

✓ 7. Python est un langage à typage dynamique. Qu'est-ce que cela signifie ?

C'est mieux que la dernière fois !

- ☐ Les variables doivent avoir un type spécifié avant utilisation
- ☐ Le type d'une variable est déterminé automatiquement au moment de l'exécution
- ☐ Python n'alloue que des variables de type entier
- ☐ Python utilise des types fixes comme dans C

✓ 8. Les modèles de données en Python définissent des comportements spécifiques pour les objets. Quelle méthode spéciale est typiquement appelée à l'initialisation d'une instance de classe ?

Attention à la syntaxe...

✗ 9. Quelles sont les deux méthodes spéciales utilisées pour obtenir un itérateur et l'utiliser ?

Excellent raisonnement !

1. _____
2. _____

✗ 10. Quels sont les noms souvent utilisés pour définir 1. les actions d'un objet et 2. les variables d'état d'un objet ?

Attention aux détails.

1. _____
2. _____

✓ 11. Quelle est la meilleure définition de `self` en Python ?

Presque parfait... encore un petit effort.

- ☐ C'est un mot clé qui permet de définir une fonction
- ☐ C'est un mot clé qui permet de définir une variable
- ☐ C'est un mot clé utilisé dans les décorateurs
- ☐ C'est une convention pour référencer l'instance de la classe actuelle dans une méthode

✓ 12. Le mot clé `next` est utilisé dans quel contexte en Python ?

Presque parfait... encore un petit effort.

- ☐ Pour sauter une itération dans une boucle `for`
- ☐ Pour obtenir le prochain élément d'un itérateur
- ☐ Pour obtenir le prochain élément d'un dictionnaire
- ☐ Pour arrêter une boucle `for`

✗ 13. Quelle est la valeur de `x` après l'exécution du code suivant ?

Tu peux le faire !

```
u = {1: 20, 3: 40, 5: 60}
x = len(u)
```

- | | | | |
|--------------------------|-------------------------|--------------------------|--------------------------|
| <input type="radio"/> 20 | <input type="radio"/> 4 | <input type="radio"/> 6 | <input type="radio"/> 60 |
| <input type="radio"/> 3 | <input type="radio"/> 5 | <input type="radio"/> 40 | |

✓ 14. Que sera affiché sur la sortie standard ?

Très bonne idée, mais mal appliquée.

```
x = [1, 2, 3]
y = x
y.append(4)
print(x)
```

- | | |
|------------------------------------|------------------------------------------------------------|
| <input type="radio"/> [3, 2, 1] | <input type="radio"/> [1, 2, 3, 4, 4] |
| <input type="radio"/> [1, 2, 3] | <input type="radio"/> [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3] |
| <input type="radio"/> [1, 2, 3, 4] | <input type="radio"/> None |

✓ 15. Quelle est la valeur affichée sur la sortie standard ?
Attention aux détails.

```
def foo(a, b=2, c=3):  
    return a - b * c
```

```
result = foo(5, c=10)  
print(result)
```

✗ 16. Que voyez-vous sur la sortie standard ?
Presque parfait... encore un petit effort.

```
u = [1, 2, 3, 4, 5]  
print([n**2 for n in u if n % 2 == 0])
```

☐ [1, 4, 9, 16, 25]

☐ [1, 2, 3, 4, 5]

☐ [1, 9, 25]

☐ [4, 16]

✓ 17. À quoi sert le module pip en Python ?
Presque parfait... encore un petit effort.

- ☐ Compiler du code Python
- ☐ Gérer les dépendances et automatiser la création d'environnements virtuels pour Python
- ☐ Installer des modules Python depuis le terminal
- ☐ Optimiser les performances du code
- ☐ Relier la sortie standard d'un programme à un autre

✓ 18. Quel est le type de `**kwargs` dans la signature d'une fonction Python ?
Attention aux détails.

☐ Une liste

☐ Un set

☐ Un dictionnaire

☐ Un tuple

✗ 19. Quelle est la différence principale entre une liste et un tuple en Python ?
Presque parfait... encore un petit effort.

- ☐ Une liste est immuable, un tuple est mutable
- ☐ Un tuple est immuable, une liste est mutable
- ☐ Une liste peut contenir des types différents, un tuple ne peut contenir qu'un seul type
- ☐ Un tuple peut contenir des types différents, une liste ne peut contenir qu'un seul type

✓ 20. Quel est l'intérêt principal d'utiliser des environnements virtuels en Python ?
Bien joué !

- ☐ Améliorer les performances d'exécution des scripts Python
- ☐ Isoler les dépendances de projets Python pour éviter les conflits entre versions de paquets
- ☐ Permettre l'exécution de scripts Python sur des systèmes d'exploitation différents
- ☐ Faciliter le déploiement de scripts Python sur des serveurs distants