

QCM : Programmation Python & Framework Django

1°) Quelle est la sortie de `print([1, 2][1:])` ?

- [1]
- [2] **Correct**
- [1, 2]
- Erreur de syntaxe

2°) Quelle instruction permet de sortir immédiatement d'une boucle ?

- pass
- **break** **Correct**
- continue
- exit

3°) Quelle instruction permet de sortir immédiatement d'une boucle ?

- pass
- **break** **Correct**
- continue
- exit

4°) Quelle signature permet de gérer un nombre variable d'arguments ?

- `*def f(a, args)`
- `def f(**kwargs)` **Correct**
- `def f(a=5)`
- `def f(*, a)`

(Note : *args gère un nombre variable d'arguments positionnels, **kwargs gère les arguments par mots-clés).

5°) En POO, quelle méthode est appelée à l'instanciation ?

- `__del__`
- `__init__` **Correct**
- `__str__`
- `__call__`

6°) Que fait try-except sans spécifier l'exception ?

- **Capture toutes les exceptions**
- Génère une erreur

- Ignore les erreurs
- Nécessite un finally

7°) Quelle expression génère [2, 4] ?

- **[x*2 for x in range(1, 3)]**
- (x**2 for x in [1, 2])
- list(map(lambda x: x+1, [1, 3]))
- [x+1 if x>1 else x for x in [1, 2, 3]]

8°) Quel type est mutable ?

- tuple
- str
- frozenset
- **list**

9°) À quoi sert self dans une méthode de classe ?

- Référence la classe parente
- **Désigne l'instance courante**
- Indique une méthode statique
- Permet l'héritage multiple

10°) Que fait super().__init__() ?

- Initialise les variables globales
- **Appelle le constructeur de la classe parente**
- Crée une sous-classe
- Importe un module

11°) Quel import permet d'utiliser sqrt(4) ?

- import math
- **from math import sqrt**
- import sqrt from math
- import math.sqrt

12°) Quel mécanisme permet de modifier le comportement des opérateurs pour une classe ?

- Les décorateurs de classe
- **Les méthodes spéciales (add, sub, etc.)**
- Les métaclasses

- Les descriptors

13°) Que retourne (x:=5) en Python 3.8+ ?

- **5**
- (5, 5)
- Erreur de syntaxe
- None

14°) Quelle est la complexité de dict.keys() en CPython ?

- **O(1)**
- O(n)
- O(log n)
- Dépend de la taille des clés

15°) Quelle technique permet de contourner le GIL ?

- **Utiliser multiprocessing**
- Implémenter des coroutines
- Désactiver le GC
- Utiliser Jython

16°) Quelle est la sortie de :

Python

```
def f():
    yield from (i*i for i in range(3))
print(list(f()))
```

- **[0, 1, 4]**
- (0, 1, 4)
- [1, 4, 9]
- Erreur

17°) Quelle syntaxe est valide pour le pattern matching (Python 3.10+) ?

- case [x, 0]:
- case { 'key': value }:
- case Point(x=0, y=y):
- **Toutes ces réponses**

18°) Quelle méthode est appelée par with ?

- **__enter__ et __exit__**
- __open__ et __close__

- `__start__` et `__stop__`
- `__init__` et `__del__`

19°) Quelle bibliothèque permet de créer des extensions C pour Python ?

- `ctypes`
- Cython
- `pybind11`
- **Toutes ces réponses**

20°) Quelle assertion sur les coroutines est fausse ?

- Utilisent `async/await`
- Peuvent être annulées
- Exécutées par l'Event Loop
- **Nécessitent un GIL distinct**