Bing

D'accord, je vais essayer de vous aider. Voici un QCM de 10 questions sur les opérations en Python, avec des réponses et des feedbacks. Bonne chance!

- Quel est le résultat de l'expression suivante?

[python]

a = 5

b = 3

c = a // b + a % b

print(c)

A) 2

B) 3

C) 4

D) 5

Réponse: B) 3

Feedback: L'opérateur // signifie la division entière, qui renvoie le quotient sans le reste. L'opérateur % signifie le modulo, qui renvoie le reste de la division. Donc, a // b vaut 1 et a % b vaut 2. Ensuite, on ajoute ces deux valeurs pour obtenir 3.

- Quel est le résultat de l'expression suivante?

[python]

x = 2 \*\* 3 \*\* 2

y = (2 \*\* 3) \*\* 2

print(x, y)

A) 64, 64

B) 512, 64

C) 64, 512

D) 512, 512

Réponse: B) 512, 64

Feedback: L'opérateur \*\* signifie la puissance, qui élève un nombre à une autre puissance. L'ordre d'évaluation de cet opérateur est de droite à gauche, donc on commence par calculer 3 \*\* 2 qui vaut 9, puis on élève 2 à cette puissance pour obtenir 512. Par contre, les parenthèses changent l'ordre de priorité, donc on commence par calculer 2 \*\* 3 qui vaut 8, puis on élève ce résultat à la puissance 2 pour obtenir 64.

- Quel est le résultat de l'expression suivante?

[python]

s = "Python"

t = s \* 3 + s[-1]

print(t)

A) PythonPythonPythonn

B) PythonPythonPytho

C) PythonnPythonnPythonn

D) PythonPythonPython

Réponse: A) PythonPythonPythonn

Feedback: L'opérateur \* signifie la répétition d'une chaîne de caractères, donc s \* 3 renvoie "PythonPythonPython". L'opérateur [] signifie l'accès à un élément d'une séquence par son indice, qui peut être négatif pour compter à partir de la fin. Donc s[-1] renvoie le dernier caractère de s, qui est "n". Ensuite, on concatène ces deux chaînes avec l'opérateur + pour obtenir "PythonPythonPythonn".

- Quel est le résultat de l'expression suivante?

[python]

u = [1, 2, 3]

v = [4, 5, 6]

w = u + v \* 2

print(w)

A) [1, 2, 3, [4, 5, 6], [4, 5, 6]]

B) [1, 2, 3, [8, 10, 12]]

C) [1, 2, 3, [4], [5], [6], [4], [5], [6]]

D) TypeError

Réponse: D) TypeError

Feedback: L'opérateur + signifie la concaténation de deux listes, donc u + v renvoie [1, 2, 3, 4, 5,

6]. Par contre, l'opérateur \* ne peut pas être appliqué entre une liste et une autre liste,

car il n'y a pas de sens à répéter une liste par un nombre de fois égal à une autre liste.

Donc v \* u renvoie une erreur de type (TypeError), et l'expression entière échoue.

- Quel est le résultat de l'expression suivante?

[python]

m = {"a":1, "b":2}

n = {"b":3, "c":4}

o = m.update(n)

print(o)

A) {"a":1, "b":3, "c":4}

B) {"a":1, "b":2}

C) None

D) AttributeError

Réponse: C) None

Feedback: La méthode update() modifie un dictionnaire en ajoutant ou remplaçant les clés et les valeurs d'un autre dictionnaire. Donc, m.update(n) change le contenu de m en {"a":1, "b":3, "c":4}, mais ne renvoie rien (None). La variable o reçoit donc la valeur None, et c'est ce qui est affiché.