# QCM PYTHON

**LIRE ET COMPRENDRE DU CODE • 15 QUESTIONS**

Nom : _____ Classe : _____ Date : _____

Une seule réponse par question. Chaque programme est indépendant.
Coche a, b, c ou d. Barème : 1 point par bonne réponse, soit 15 points.

## 1. Afficher un texte

```
print("Bonjour")
```

Que voit-on à l'écran ?

- ☐ **a.** "Bonjour" avec les guillemets
- ☐ **b.** Bonjour
- ☐ **c.** print Bonjour
- ☐ **d.** Rien ne s'affiche

## 2. Afficher une variable

```
age = 16
print("Âge :", age)
```

Que voit-on à l'écran ?

- ☐ **a.** Âge : age
- ☐ **b.** 16
- ☐ **c.** Âge : 16
- ☐ **d.** Âge : "16"

## 3. Récupérer une réponse

```
prenom = input("Ton prénom ? ")
print("Bonjour", prenom)
```

L'élève saisit Léa. Qu'affiche la dernière ligne ?

- ☐ **a.** Bonjour Léa
- ☐ **b.** Bonjour prenom
- ☐ **c.** Ton prénom ?
- ☐ **d.** Léa Bonjour

# QCM PYTHON

**LIRE ET COMPRENDRE DU CODE • 15 QUESTIONS**

Nom : _____ Classe : _____ Date : _____

Une seule réponse par question. Chaque programme est indépendant.
Coche a, b, c ou d. Barème : 1 point par bonne réponse, soit 15 points.

## 4. Convertir une saisie

```
age = int(input("Ton âge ? "))
age_futur = age + 1
print(age_futur)
```

L'élève saisit 16. Qu'affiche la dernière ligne ?

- ☐ **a.** 16
- ☐ **b.** 161
- ☐ **c.** age + 1
- ☐ **d.** 17

## 5. Choisir entre deux possibilités

```
age = 18

if age >= 18:
    print("Majeur")
else:
    print("Mineur")
```

Que voit-on à l'écran ?

- ☐ **a.** Mineur
- ☐ **b.** Majeur
- ☐ **c.** Majeur, puis Mineur
- ☐ **d.** Rien ne s'affiche

## 6. Tester plusieurs conditions

```
note = 17

if note >= 16:
    print("Très bien")
elif note >= 10:
    print("Moyenne atteinte")
else:
    print("Moyenne non atteinte")
```

Que voit-on à l'écran ?

- ☐ **a.** Moyenne atteinte
- ☐ **b.** Très bien, puis Moyenne atteinte
- ☐ **c.** Très bien
- ☐ **d.** Moyenne non atteinte

# QCM PYTHON

**LIRE ET COMPRENDRE DU CODE • 15 QUESTIONS**

Nom : _____ Classe : _____ Date : _____

Une seule réponse par question. Chaque programme est indépendant.
Coche a, b, c ou d. Barème : 1 point par bonne réponse, soit 15 points.

## 7. Déterminer un tarif

```
age = 12
if age < 12:
    tarif = 5
elif age < 18:
    tarif = 7
else:
    tarif = 10
print(tarif)
```

Que voit-on à l'écran ?

- ☐ **a.** 5
- ☐ **b.** 10
- ☐ **c.** 12
- ☐ **d.** 7

## 8. Comprendre range()

```
for i in range(3):
    print(i)
```

Quelle suite s'affiche, à raison d'un nombre par ligne ?

- ☐ **a.** 0, puis 1, puis 2
- ☐ **b.** 1, puis 2, puis 3
- ☐ **c.** 0, puis 1, puis 2, puis 3
- ☐ **d.** 3 uniquement

## 9. Calculer dans une boucle

```
total = 0

for nombre in range(1, 4):
    total = total + nombre

print(total)
```

Que voit-on à l'écran ?

- ☐ **a.** 4
- ☐ **b.** 10
- ☐ **c.** 6
- ☐ **d.** 3

# QCM PYTHON

**LIRE ET COMPRENDRE DU CODE • 15 QUESTIONS**

Nom : _____ Classe : _____ Date : _____

Une seule réponse par question. Chaque programme est indépendant.
Coche a, b, c ou d. Barème : 1 point par bonne réponse, soit 15 points.

## 10. Suivre une boucle while

```
compteur = 1

while compteur <= 3:
    print(compteur)
    compteur = compteur + 1
```

Quelle suite s’affiche, à raison d’un nombre par ligne ?

- ☐ **a.** 0, puis 1, puis 2
- ☐ **b.** 1, puis 2, puis 3
- ☐ **c.** 1, puis 2, puis 3, puis 4
- ☐ **d.** Le nombre 1 indéfiniment

## 11. Définir une fonction

```
def dire_bonjour():
    print("Bonjour !")

print("Fin")
```

Que voit-on à l’écran ?

- ☐ **a.** Bonjour !
- ☐ **b.** Bonjour !, puis Fin
- ☐ **c.** Rien ne s’affiche
- ☐ **d.** Fin

## 12. Réutiliser un résultat

```
def double(nombre):
    return nombre * 2

resultat = double(4)
print(resultat + 1)
```

Que voit-on à l’écran ?

- ☐ **a.** 9
- ☐ **b.** 8
- ☐ **c.** 5
- ☐ **d.** None

# QCM PYTHON

**LIRE ET COMPRENDRE DU CODE • 15 QUESTIONS**

Nom : _____ Classe : _____ Date : _____

Une seule réponse par question. Chaque programme est indépendant.
Coche a, b, c ou d. Barème : 1 point par bonne réponse, soit 15 points.

## 13. Lire un élément d’une liste

```
fruits = ["pomme", "banane", "kiwi"]
print(fruits[1])
```

Que voit-on à l’écran ?

- ☐ **a.** pomme
- ☐ **b.** kiwi
- ☐ **c.** banane
- ☐ **d.** 1

## 14. Ajouter un élément

```
notes = [12, 15, 9]
notes.append(18)

print(len(notes))
```

Que voit-on à l’écran ?

- ☐ **a.** 3
- ☐ **b.** 4
- ☐ **c.** 18
- ☐ **d.** 54

## 15. Modifier puis parcourir une liste

```
fruits = ["pomme", "banane", "kiwi"]
fruits[1] = "poire"

for fruit in fruits:
    print(fruit)
```

Quelle suite s’affiche, à raison d’un fruit par ligne ?

- ☐ **a.** pomme, puis banane, puis kiwi
- ☐ **b.** poire uniquement
- ☐ **c.** pomme, puis banane, puis kiwi, puis poire
- ☐ **d.** pomme, puis poire, puis kiwi