Python : variables, tests et fonctions – exercices

Mathématiques – Seconde

Algorithmique et programmation · exercices ★ à ★★★, pièges et QCM

Nom : _____ Date : _____

Durée : environ 1 h. *Calculatrice interdite, sauf mention contraire. Les exercices vont de ★ (automatismes) à ★★★ (niveau d'un contrôle exigeant). Fais les séries dans l'ordre ; si une série ★ est réussie sans erreur, passe à la suite. Corrige-toi avec le corrigé, puis refais chaque calcul faux sans regarder.*

Les pièges de cette fiche

- `=` range une valeur, `==` compare deux valeurs.
- `/` donne toujours un flottant ; `//` donne le quotient entier et `%` le reste.
- La puissance s'écrit `**`, et elle passe avant `*` et `+`.
- Le membre de droite se calcule avec les valeurs **du moment** : après `x = y`, l'ancienne valeur de `x` est perdue.
- `print` affiche, `return` renvoie : sans `return`, la fonction renvoie `None`.
- Avec `if / elif`, la première condition vraie gagne : le cas le plus restrictif d'abord.

Exercice 1 – Que contiennent les variables ? ★

Pour chaque programme, donne la valeur de chaque variable à la fin.

a)

```
a = 5
b = a + 3
a = 2 * b
```

b)

```
x = 7
x = x - 10
x = x * x
```

c)

```
a = 3
b = 8
a = b
b = a
```

d)

```
a = 3
b = 8
c = a
a = b
b = c
```

e)

```
n = 4
n = n + 1
m = n * 2
n = m - n
```

f)

```
x = 2
y = x ** 3
x = y % 5
```

g)

```
a = 10
b = a / 4
c = a // 4
```

h)

```
s = "4"
t = s + s
u = 2 * s
```

Exercice 2 – Opérations et types ★

Donne la valeur que Python calcule pour chaque expression.

a) `17 // 5`

b) `17 % 5`

c) `2 ** 5`

d) `7 / 2`

e) `8 / 2`

f) `20 % 4`

g) `3 * 2 ** 2`

h) `(10 - 3) * 2`

i) `30 // 7`

j) `5 == 5.0`

k) `7 % 2 == 1`

l) `"3" + "5"`

Exercice 3 – Instructions conditionnelles ★

Qu'affiche chaque programme pour la valeur de `n` indiquée (placée en première ligne) ?

Programme P1

```
if n >= 10:
    r = n - 10
elif n >= 0:
    r = 2 * n
else:
    r = -n
print(r)
```

Programme P2

```
if n % 2 == 0 and n % 3 == 0:
    print("A")
elif n % 3 == 0:
    print("B")
else:
    print("C")
```

a) P1, `n = 15`

b) P1, `n = 10`

c) P1, `n = 4`

d) P1, `n = 0`

e) P1, `n = -3`

f) P1, `n = 9.5`

g) P2, `n = 12`

h) P2, `n = 9`

i) P2, `n = 8`

j) P2, `n = 0`

Exercice 4 – Appeler une fonction ★★

Pour chaque fonction, donne ce que renvoient les appels.

a)

```
def f(x):
    return 3 * x - 5
```

Que renvoient `f(4)` et `f(-1.5)` ?

b)

```
def g(x):
    return x ** 2 - 2 * x
```

Que renvoient `g(-3)` et `g(0.5)` ?

c)

```
def tarif(age):
    if age < 12:
        return 5
    else:
        return 9
```

Que renvoient `tarif(11)` et `tarif(12)` ?

d)

```
def moyenne(a, b):
    return (a + b) / 2
```

Que renvoient `moyenne(7, 12)` et `moyenne(moyenne(4, 8), 10)` ?

e)

```
def double(x):
    print(2 * x)
y = double(5)
```

Qu'affiche ce programme ? Que contient `y` ?

Exercice 5 – Écrire une fonction ★★

Écris en Python la fonction décrite, avec `def` et `return`, puis donne le résultat du test.

a) `f(x)` : l'image de x par $f(x) = 2x^2 - 3x + 1$. Teste avec $x = 2$.

b) `coef(t)` : le coefficient multiplicateur d'une évolution de t % (t négatif pour une baisse). Teste avec une baisse de 20 %.

c) `taux(v1, v2)` : le taux d'évolution, en %, de la valeur v_1 à la valeur v_2. Teste de 50 à 60.

d) `distance(xA, yA, xB, yB)` : la distance AB dans un repère orthonormé. Teste avec $A(1\,;2)$ et $B(4\,;6)$.

e) `milieu(xA, yA, xB, yB)` : les coordonnées du milieu de $[AB]$. Teste avec $A(-1\,;3)$ et $B(4\,;5)$.

f) `est_pair(n)` : `True` si l'entier n est pair, `False` sinon. Teste avec 7.

Exercice 6 – Programmes à trous ★★

Complète chaque `...` pour que la fonction fasse ce qui est annoncé : `va` = valeur absolue, `maxi` = le plus grand des deux et `sur_droite` = le point est-il sur la droite d'équation $y = 2x + 1$.

a)

```
def va(x):
    if x ... 0:
        return ...
    else:
        return x
```

b)

```
def maxi(a, b):
    if ...:
        return a
    else:
        return ...
```

c)

```
def sur_droite(x, y):
    if ... == ...:
        return True
    return False
```

Exercice 7 – La livraison ★★★

Un site de vente en ligne facture 4,90 € de livraison pour une commande de moins de 50 €. La livraison est gratuite à partir de 50 €. À partir de 100 €, la livraison est gratuite et le client a en plus 10 % de remise. On note m le montant de la commande, en euros.

1. Calcule le prix payé pour une commande de 30 €, de 60 €, puis de 120 €.
2. Écris une fonction `total(m)` qui renvoie le prix payé.
3. Un élève écrit le même programme en testant d'abord `m >= 50`, puis `m >= 100`. Que renvoie sa fonction pour 120 € ? Explique.

Exercice 8 – Le disque ★★★

Dans un repère orthonormé, $\mathscr{D}$ est le disque de centre $\Omega(2\,;-1)$ et de rayon 5 : un point M est dans $\mathscr{D}$ quand $\Omega M \leqslant 5$.

1. Calcule ΩA pour $A(5\,;3)$. A est-il dans le disque ?
2. Écris une fonction `dans_disque(x, y)` qui renvoie `True` si le point $M(x\,;y)$ est dans le disque, `False` sinon.
3. Que renvoient `dans_disque(7, 2)` et `dans_disque(0, 0)` ? Justifie par un calcul.

Exercice 9 – Trouve l'erreur ★★

Chaque programme fait autre chose que ce qui est annoncé en dessous. Trouve l'erreur, puis corrige.

a)

```
n = 14
if n % 2 = 0:
    print("pair")
```

Doit afficher pair si n est pair.

b)

```
def aire(r):
    print(3 * r ** 2)
A = aire(10) + 1
```

Doit mettre 301 dans A.

c)

```
def prix(age):
    if age < 18:
        return 6
    elif age < 12:
        return 4
    else:
        return 10
p = prix(8)
```

Moins de 12 ans : 4 €.

d)

```
oeufs = 45
boites = oeufs / 6
```

Nombre de boîtes de 6 pleines.

e)

```
c = 5
A = c ^ 2
```

Aire d'un carré de côté 5.

f)

```
xA, yA, xB, yB = 0, 1, 5, 13
d = (xB - xA) ** 2 + (yB - yA) ** 2 ** 0.5
```

Distance AB.

QCM – Six questions comme en contrôle

Une seule bonne réponse par question. Note ta réponse, puis vérifie dans le corrigé.

1. En Python, `19 % 4` vaut :

   A `4` **B** `4.75` **C** `3` **D** `76`

2. Pour tester si `n` vaut 10, on écrit :

   A `if n == 10:` **B** `if n = 10:` **C** `if n == 10` **D** `if (n = 10):`

3. Après `a = 4`, `b = 7`, `a = a + b`, `b = a - b`, la variable `b` vaut :

   A `7` **B** `11` **C** `−3` **D** `4`

4. `2 + 3 ** 2` vaut :

   A `25` **B** `11` **C** `10` **D** `8`

5. Une fonction sans `return` qui se termine par `print(x)` renvoie :

   A la valeur de `x` **B** `0` **C** `True` **D** `None`

6. Avec `def f(x):` qui renvoie `2 * x` si `x > 0`, et `x - 1` sinon, `f(0)` vaut :

   A `−1` **B** `0` **C** `1` **D** `2`

Ce document est la propriété intellectuelle de Ben, professeur particulier (benprofparticulier.fr), protégée par le Code de la propriété intellectuelle (art. L. 111-1 et L. 122-4). Il est mis à disposition pour un usage personnel de révision. Toute reproduction, diffusion, mise en ligne ou revente, totale ou partielle, sans autorisation écrite est interdite et constitue une contrefaçon (art. L. 335-2).