Python : boucles for et while – 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(a, b)` va de a à $b-1$: elle fait $b-a$ tours.
- Somme à `0`, produit à `1`, compteur à `0` : **avant** la boucle, jamais dedans.
- L'indentation décide : une ligne décalée est répétée, une ligne non décalée s'exécute une fois après la boucle.
- La condition du `while` est le **contraire** de la condition d'arrêt.
- Pas de `!=` dans un `while` qui avance par sauts : la boucle risque de ne jamais s'arrêter.
- Hausse de t % : on multiplie par $1 + t/100$; baisse de t % : par $1 - t/100$.

Exercice 1 – Les valeurs de k ★

Pour `for k in ...`, donne les valeurs prises par `k` et le nombre de tours.

a) `range(4)`

b) `range(1, 5)`

c) `range(3, 8)`

d) `range(10)`

e) `range(5, 6)`

f) `range(5, 5)`

g) `range(-2, 3)`

h) `range(1, 10, 3)`

i) `range(0, 20, 5)`

j) `range(9, 4, -1)`

Exercice 2 – Qu'affiche le programme ? ★

Écris ce que chaque programme affiche, dans l'ordre.

a)

```
for k in range(4):
    print(k * k)
```

b)

```
for k in range(1, 4):
    print(2 * k + 1)
```

c)

```
x = 1
for k in range(3):
    x = x * 2
print(x)
```

d)

```
x = 1
for k in range(3):
    x = x * 2
    print(x)
```

e)

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

f)

```
n = 10
for k in range(3):
    n = n - 3
print(n)
```

g)

```
for k in range(2, 7):
    if k % 2 == 0:
        print(k)
```

h)

```
s = 0
for k in range(1, 5):
    s = s + k
    print(s)
```

Exercice 3 – Accumulateurs ★

Donne la valeur de la variable de la première ligne à la fin du programme.

a)

```
S = 0
for k in range(1, 6):
    S = S + k
```

b)

```
P = 1
for k in range(1, 5):
    P = P * k
```

c)

```
S = 0
for k in range(1, 6):
    S = S + k * k
```

d)

```
c = 0
for k in range(1, 21):
    if k % 3 == 0:
        c = c + 1
```

e)

```
P = 0
for k in range(1, 5):
    P = P * k
```

f)

```
S = 0
for k in range(5):
    S = S + 2
```

g)

```
u = 8
for k in range(3):
    u = u * 1.5
```

h)

```
S = 0
for k in range(1, 11):
    S = S + 2 * k - 1
```

Exercice 4 – Tableau de suivi ★★

Donne chaque variable avant la boucle, puis à la fin de chaque tour (des colonnes peuvent rester vides).

a)

```
u = 3
n = 0
while u < 100:
    u = 2 * u + 1
    n = n + 1
```

Tour	Avant	1	2	3	4	5	6
u	3						
n	0						

b)

```
a = 1
b = 1
for k in range(5):
    c = a + b
    a = b
    b = c
```

Tour	Avant	1	2	3	4	5	6
k							
a	1						
b	1						

Exercice 5 – while : au bout de combien… ? (calculatrice autorisée) ★★

a)

Une ville a 5000 habitants ; sa population augmente de 3 % par an. Que vaut n à la fin ? Que représente-t-il ?

```
P = 5000
n = 0
while P < 6000:
    P = P * 1.03
    n = n + 1
```

b)

Une voiture coûte 20 000 € ; sa valeur baisse de 15 % par an. Que vaut n à la fin ? Que représente-t-il ?

```
V = 20000
n = 0
while V >= 10000:
    V = V * 0.85
    n = n + 1
```

c) On place 1000 € à 2 % par an (intérêts composés). Écris un programme qui donne le nombre d'années pour atteindre au moins 1200 €.

d) Une culture de 500 bactéries double chaque heure. Écris un programme qui donne le nombre d'heures pour dépasser un million de bactéries.

Exercice 6 – Programmes à trous ★★

Complète les `...` : a) $1 + 2 + \cdots + 50$; b) nombre de multiples de 7 de 1 à 100 ; c) $1 \times 2 \times \cdots \times 6$; d) nombre de divisions par 2 pour que $u < 1$.

a)

```
S = 0
for k in range(..., ...):
    S = ...
```

b)

```
c = 0
for k in range(1, 101):
    if ...:
        c = c + 1
```

c)

```
P = ...
for k in range(1, ...):
    P = P * k
```

d)

```
u = 1000
n = 0
while ...:
    u = u / 2
    n = ...
```

Exercice 7 – Deux placements ★★★

Léa place 2000 €. Placement A : le capital augmente de 3 % chaque année. Placement B : le capital augmente de 80 € chaque année. Elle veut savoir lequel dépasse le premier 3000 €.

1. Placement A : par quel nombre multiplie-t-on le capital chaque année ? Calcule-le après 1 an, puis 2 ans.
2. Complète ce programme : il doit donner le nombre d'années pour que le capital A atteigne 3000 €.

```
C = 2000
n = 0
while ...:
    C = ...
    n = n + 1
```

3. Le programme complété donne `n` $= 14$. Écris le programme du placement B et donne son résultat.
4. Lequel dépasse le premier 3000 € ? Lequel est le plus intéressant au bout de 20 ans ?

Exercice 8 – Le lac ★★★

Un lac contient 5000 tonnes d'algues. Chaque année, 10 % des algues disparaissent, puis 200 tonnes nouvelles apparaissent. On note `u` la masse d'algues (en tonnes) et on étudie le programme ci-contre.

```
u = 5000
n = 0
while u > 2500:
    u = 0.9 * u + 200
    n = n + 1
```

1. Calcule la masse au bout de 1 an, puis de 2 ans.
2. Que calcule ce programme ? On trouve `n` $= 18$: interprète.
3. Un élève remplace `2500` par `1500`. Montre que $u - 2000$ est multiplié par 0,9 chaque année, puis explique pourquoi son programme ne s'arrête jamais.

Exercice 9 – Trouve l'erreur ★★

Chaque programme ne fait pas ce qui est annoncé. Trouve l'erreur, puis corrige.

a)

```
P = 1
for k in range(1, 4):
    P = P * 2
```

But : calculer 2^4.

b)

```
for k in range(1, 9):
    S = 0
    S = S + k
```

But : $1 + 2 + \cdots + 8$.

c)

```
u = 1
n = 0
while u != 100:
    u = u * 3
    n = n + 1
```

But : tripler u jusqu'à dépasser 100.

d)

```
c = 0
for k in range(1, 31):
    if k % 4 == 0:
        print(k)
    c = c + 1
```

But : compter les multiples de 4.

e)

```
u = 500
n = 0
while u > 1000:
    u = u * 1.1
    n = n + 1
```

But : années pour que u atteigne 1000.

f)

```
S = 0
for k in range(1, 11):
    S = k
```

But : $1 + 2 + \cdots + 10$.

QCM – Six questions comme en contrôle

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

1. `for k in range(2, 9):` fait :

   A 8 tours **B** 9 tours **C** 7 tours **D** 6 tours

2. Après `S = 0` et `for k in range(1, 4): S = S + k * k`, `S` vaut :

   A 14 **B** 6 **C** 30 **D** 5

3. Pour « tant que le capital C n'a pas atteint 5000 », on écrit :

   A `while C >= 5000:` **B** `while C != 5000:` **C** `while C == 5000:` **D** `while C < 5000:`

4. Une quantité baisse de 8 % par an. Dans la boucle, on écrit :

   A `Q = Q * 0.08` **B** `Q = Q * 0.92` **C** `Q = Q - 0.08` **D** `Q = Q * 1.08`

5. Pour calculer un produit dans une boucle, on initialise `P` :

   A à 0, avant la boucle **B** à 1, dans la boucle **C** à 0, dans la boucle **D** à 1, avant la boucle

6. Après `x = 2` et `for k in range(3): x = x * x`, `x` vaut :

   A 256 **B** 8 **C** 16 **D** 64

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).