Exercice 1 — Les types de variables ●○○

Donne le type de chaque variable après ces affectations : entier (`int`), flottant (`float`), chaîne de caractères (`str`) ou booléen (`bool`).

```
a = 5
b = 2.5
c = "5"
d = a > 3
e = a / 2
```

Exercice 2 — Des affectations successives ●○○

Que contiennent les variables `a` et `b` à la fin du programme ?

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

Exercice 3 — Une instruction conditionnelle ●○○

Un magasin offre $10\,\%$ de réduction pour un achat d'au moins 100 €.

```
def prix_final(p):
    if p >= 100:
        return p * 0.9
    else:
        return p
```

Que renvoient `prix_final(80)`, `prix_final(100)` et `prix_final(250)` ?

Exercice 4 — Une boucle for ●○○

Qu'affiche ce programme ? Modifie-le pour calculer la somme des entiers de 1 à 100.

```
s = 0
for k in range(1, 11):
    s = s + k
print(s)
```

Exercice 5 — Une boucle while ●●○

Que renvoie `seuil(1000)` ? À quoi sert cette fonction ?

```
def seuil(m):
    n = 0
    p = 1
    while p <= m:
        p = 2 * p
        n = n + 1
    return n
```

Exercice 6 Écrire une fonction ●○○

a. Écris une fonction `aire_rectangle(L, l)` qui renvoie l'aire d'un rectangle.

b. Écris une fonction `moyenne(a, b, c)` qui renvoie la moyenne de trois notes.

Exercice 7 Tester si un nombre est multiple ●●○

En Python, `a % b` donne le reste de la division euclidienne de `a` par `b`.

a. Que valent `17 % 5` et `24 % 6` ?

b. Écris une fonction `est_multiple(a, b)` qui renvoie `True` si `a` est un multiple de `b`, `False` sinon.

Exercice 8 — Le plus grand multiple ●●○

Complète cette fonction pour qu'elle renvoie le plus grand multiple de `a` inférieur ou égal à `b` (avec a et b entiers positifs).

```
def plus_grand_multiple(a, b):
    m = 0
    while m + a <= b:
        m = 
    return m
```

Que renvoie `plus_grand_multiple(7, 100)` ?

Exercice 9 — Doubler son capital ●●○

Un capital de $1\,000$ € rapporte $5\,\%$ par an.

```
c = 1000
n = 0
while c < 2000:
    c = c * 1.05
    n = n + 1
print(n)
```

a. Que représentent `c` et `n` ?

b. Le programme affiche `15`. Interprète ce résultat.

Exercice 10 Simuler un dé ●●○

La fonction `randint(1, 6)` du module `random` renvoie un entier au hasard entre 1 et 6.

```
from random import randint

def frequence_six(n):
    compteur = 0
    for i in range(n):
        if randint(1, 6) == 6:
            compteur = compteur + 1
    return compteur / n
```

a. Explique ce que renvoie `frequence_six(1000)`.

b. Vers quelle valeur ce résultat doit-il se rapprocher quand n devient grand ?

Exercice 11 Comprendre une fonction ●●○

Que renvoie `mystere(3, 8)` ? Et `mystere(-2, -7)` ? Que fait cette fonction ?

```
def mystere(a, b):
    if a > b:
        return a
    return b
```

Exercice 12 — Chercher un minimum par balayage ●●●

On cherche le minimum de $f(x) = x^2 - 4x + 1$ sur $[0\,;\,5]$ en testant les valeurs de x de $0{,}1$ en $0{,}1$.

```
def f(x):
    return x**2 - 4*x + 1

xmin = 0
for k in range(51):
    x = k / 10
    if f(x) < f(xmin):
        xmin = x
print(xmin, f(xmin))
```

a. Quelles valeurs de x le programme teste-t-il ?

b. Le programme affiche `2.0 -3.0`. Démontre ce résultat avec $f(x) = (x-2)^2 - 3$.

CAMPUS XYZ

Corrigés

Exercices sur l'algorithmique et Python en seconde

Exercices sur l'algorithmique et Python en seconde : corrigés

CAMPUS XYZ

Exercice 1 — Les types de variables ●○○

- `a` : **entier** ; `b` : **flottant** (nombre à virgule, écrit avec un point).
- `c` : **chaîne de caractères** (entre guillemets, ce n'est pas un nombre).
- `d` : **booléen**, il vaut `True` car $5 > 3$.
- `e` : **flottant**, il vaut `2.5` (la division `/` donne toujours un flottant).

Exercice 2 — Des affectations successives ●○○

On suit le programme ligne par ligne : une affectation remplace l'ancienne valeur.

- `a = 3` ; puis `b = 5` ; puis `a = 10` ; puis `b = 10 - 5 = 5`.

À la fin : **`a = 10` et `b = 5`**.

Exercice 3 — Une instruction conditionnelle ●○○

- $80 < 100$: pas de réduction, **`80`**.
- $100 \geqslant 100$: **`90.0`**.
- **`225.0`**.

> `>=` signifie « supérieur ou égal » : le cas $p = 100$ bénéficie de la réduction.

Exercice 4 — Une boucle for ●○○

`range(1, 11)` donne les entiers de 1 à 10 (le 11 est exclu). Le programme ajoute $1 + 2 + \cdots + 10$: il affiche **`55`**.

Pour aller jusqu'à 100, on écrit `range(1, 101)` : le programme affiche alors **`5050`**.

Exercice 5 — Une boucle while ●●○

À chaque tour, `p` double et `n` augmente de 1 : après n tours, $p = 2^n$. La boucle s'arrête dès que $p > m$.

$2^9 = 512 \leqslant 1\,000$ et $2^{10} = 1\,024 > 1\,000$: `seuil(1000)` renvoie **`10`**. La fonction donne le plus petit entier n tel que $2^n > m$.

Exercice 6 — Écrire une fonction ●○○

Une fonction reçoit des arguments et renvoie un résultat avec `return`.

```
def aire_rectangle(L, l):
    return L * l

def moyenne(a, b, c):
    return (a + b + c) / 3
```

> Attention aux parenthèses : `a + b + c / 3` ne diviserait que `c`.

Exercice 7 — Tester si un nombre est multiple ●●○

a. $17 = 5 \times 3 + 2$: **2** ; $24 = 6 \times 4$: **0**.

b. a est un multiple de b quand le reste est nul :

```
def est_multiple(a, b):
    return a % b == 0
```

Exercice 8 — Le plus grand multiple ●●○

On part de 0 et on ajoute a tant qu'on ne dépasse pas b : il faut écrire **`m = m + a`**.

`plus_grand_multiple(7, 100)` renvoie **98** (7×14). Autre écriture possible : `return (b // a) * a`.

Exercice 9 — Doubler son capital ●●○

a. `c` est le capital après `n` années : chaque année, il est multiplié par $1{,}05$.

b. Il faut 15 **ans** pour que le capital atteigne au moins $2\,000$ € : $1\,000 \times 1{,}05^{14} \approx 1\,980$ et $1\,000 \times 1{,}05^{15} \approx 2\,079$.

Exercice 10 — Simuler un dé ●●○

a. On lance $1\,000$ fois un dé virtuel, on compte les 6, puis on divise par $1\,000$: c'est la **fréquence d'apparition du 6**.

b. Pour un dé équilibré, elle se rapproche de $\frac{1}{6} \approx 0{,}167$ (loi des grands nombres). Chaque exécution donne un résultat un peu différent.

Exercice 11 — Comprendre une fonction ●●○

- `mystere(3, 8)` : $3 > 8$ est faux, on renvoie **8**.
- `mystere(-2, -7)` : $-2 > -7$ est vrai, on renvoie **−2**.

La fonction renvoie **le plus grand des deux nombres**. Le second `return` n'est atteint que si la condition est fausse.

Exercice 12 — Chercher un minimum par balayage ●●●

a. `k` va de 0 à 50, donc $x = \dfrac{k}{10}$ prend les valeurs $0\,;\,0{,}1\,;\,0{,}2\,;\,\ldots\,;\,5$.

b. $(x-2)^2 - 3 = x^2 - 4x + 4 - 3 = x^2 - 4x + 1$. Un carré est positif, donc $f(x) \geqslant -3$, avec égalité pour $x = 2$: **le minimum de f est -3, atteint en $x = 2$**. Le balayage donne ici la valeur exacte car 2 fait partie des valeurs testées.

LES FICHES MAGIQUES SECONDE

Révise toute la seconde avec les fiches magiques



- **Plus de 40 fiches claires et illustrées** : tout le programme de seconde, une notion par page.
- **Des vidéos méthode** accessibles par QR code sur les fiches.
- **Des exercices corrigés** après chaque fiche pour vérifier que tout est acquis.

campusxyz.fr/produit/les-fiches-magiques-2de