



TD Python 1 Premiers pas : variables, conditions et boucles

Tous les scripts s'écrivent et s'exécutent dans Basthon (<https://basthon.fr>) ou Capytale. Pour chaque exercice, on écrit d'abord ce que l'on attend, puis on exécute, puis on compare.

Exercice 1 (★☆☆☆) obligatoire Prédire la sortie

Pour chacun des scripts suivants, écrire sur le papier ce qu'il affiche, en suivant l'évolution de chaque variable ligne par ligne. Vérifier ensuite en l'exécutant.

```
1. a = 7
2  b = 3
3  a = a + b
4  b = a - b
5  a = a - b
6  print(a, b)
```

```
2. n = 2025
2  q = n // 100
3  r = n % 100
4  print(q, r, q * 100 + r)
5  print(n // 10 % 10)
```

```
3. x = 2
2  y = x ** 3 - 3 * x ** 2 + 1
3  x = y
4  print(x, y)
5  print(-7 // 2, -7 % 2, 7 // -2)
```

Exercice 2 (★☆☆☆) Types, conversions et erreurs

1. Pour chacune des expressions suivantes, donner sa valeur et son type, ou indiquer le type de l'erreur qu'elle provoque. Vérifier en console.

- | | |
|------------------------------------|--------------------------------|
| a) <code>7 / 2</code> | f) <code>5 // 0</code> |
| b) <code>7 // 2 * 2 + 7 % 2</code> | g) <code>1e3 == 1000</code> |
| c) <code>"3" + "4"</code> | h) <code>0.1 * 3 == 0.3</code> |
| d) <code>int("3") + 4</code> | i) <code>type(2 ** 100)</code> |
| e) <code>"3" + 4</code> | j) <code>int(-3.7)</code> |

2. Chacun des trois scripts suivants provoque une erreur. Sans l'exécuter, donner le type de l'erreur (`SyntaxError`, `NameError`, `TypeError`, `ZeroDivisionError`), puis le corriger.

```
a) prix = input("Prix ? ")
2  print(prix * 1.2)

b) for k in range(5)
2  print(k)

c) total = 0
2  for k in range(1, 5) :
3      total = total + k
4  print(totale)
```

Exercice 3 (★☆☆☆) obligatoire **Multiples et diviseurs**

1. Écrire une boucle `for` qui affiche, sur une seule ligne, tous les multiples de 7 compris entre 1 et 100. On utilisera `range` avec un pas, et `print(k, end=" ")` pour rester sur la même ligne.
2. Écrire une boucle qui affiche tous les diviseurs positifs de 84.
3. Combien y a-t-il d'entiers de $\llbracket 1, 1000 \rrbracket$ divisibles à la fois par 4 et par 6 ? Écrire un script qui les compte, puis retrouver ce nombre par un raisonnement.

Exercice 4 (★★☆☆) obligatoire **Tables de vérité**

On rappelle qu'une boucle sur (`True`, `False`) imbriquée dans une autre parcourt les quatre lignes d'une table de vérité à deux propositions.

1. Le *ou exclusif* de deux propositions P et Q est vrai lorsqu'exactement l'une des deux est vraie. Écrire un script qui affiche, sur chacune des quatre lignes, les valeurs de P , Q et des deux expressions

$$P \neq Q \quad \text{et} \quad (P \text{ or } Q) \text{ and not } (P \text{ and } Q).$$

Que constate-t-on ?

2. Vérifier par un script que l'implication $P \implies Q$, programmée par `(not P) or Q`, a la même table de vérité que sa contraposée $\bar{Q} \implies \bar{P}$.
3. Avec trois boucles imbriquées, vérifier sur les huit cas la distributivité

$$P \wedge (Q \vee R) \iff (P \wedge Q) \vee (P \wedge R).$$

On utilisera un accumulateur booléen `tout`, initialisé à `True`, qui reste vrai tant que tous les cas testés sont vrais, et l'on affichera seulement sa valeur finale.

Exercice 5 (★★☆☆) **Sommes et conjectures**

1. Écrire un script qui calcule la somme $1 + 3 + 5 + \dots + (2n - 1)$ des n premiers entiers impairs pour $n = 10$.
2. Modifier le script pour afficher, pour chaque n de 1 à 10, la valeur de n , la somme, et n^2 . Conjecturer une formule.
3. Vérifier par un script que, pour tout $n \in \llbracket 1, 100 \rrbracket$, on a $\sum_{k=1}^n k^3 = \left(\frac{n(n+1)}{2}\right)^2$. On n'affichera qu'un seul booléen. Cette vérification constitue-t-elle une démonstration ?

Exercice 6 (★★☆☆) **Une suite définie par récurrence**

Soit (u_n) la suite définie par $u_0 = 3$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = 3u_n - 4$.

1. Écrire un script qui calcule et affiche u_{10} .
2. Écrire un script qui détermine le plus petit entier n tel que $u_n > 10^6$, et affiche n et u_n .
3. Vérifier par un script que $u_n = 2 + 3^n$ pour $n \in \llbracket 1, 10 \rrbracket$.
4. On remplace $u_0 = 3$ par $u_0 = 2$ dans le script de la question 2. Que se passe-t-il ? Expliquer, puis ajouter un garde-fou.

Exercice 7 (★★☆☆) obligatoire **La suite de Syracuse**

Partant d'un entier $n \geq 1$, on applique la règle suivante : si n est pair, on le remplace par $n/2$; sinon, on le remplace par $3n + 1$. On s'arrête lorsque n vaut 1. On conjecture (sans qu'aucune démonstration ne soit connue) que l'on atteint toujours 1.

1. Écrire un script qui, partant de $n = 27$, compte le nombre d'étapes nécessaires pour atteindre 1.
 2. Le modifier pour afficher aussi la plus grande valeur atteinte au cours du vol.
 3. Parmi les entiers de départ de 1 à 100, lequel demande le plus d'étapes ? Combien ?
-

Exercice 8 (★★☆☆) Nombres parfaits

Un entier $n \geq 2$ est *parfait* s'il est égal à la somme de ses diviseurs positifs autres que lui-même. Par exemple $6 = 1 + 2 + 3$.

1. Écrire un script qui calcule la somme des diviseurs de 28 autres que 28, et affiche si 28 est parfait.
 2. Écrire un script qui affiche tous les nombres parfaits inférieurs à 10 000.
 3. Un diviseur de n autre que n est au plus égal à $n/2$. En déduire une amélioration du script.
-

Exercice 9 (★★★★☆) Nombres premiers

1. Soit $n \geq 2$. Écrire un script qui cherche le plus petit diviseur $d \geq 2$ de n à l'aide d'une boucle **while**, en s'arrêtant dès que $d^2 > n$ (on justifiera pourquoi c'est suffisant). Le script affiche si n est premier et, sinon, le diviseur trouvé. Tester avec $n = 221$.
 2. En imbriquant ce test dans une boucle **for**, afficher tous les nombres premiers inférieurs à 100 et les compter.
 3. Combien y a-t-il de nombres premiers inférieurs à 10 000 ?
-

Exercice 10 (★★★★☆) Additionner des dixièmes

1. Écrire un script qui part de $x = 0$ et ajoute 0,1 dix fois de suite, puis affiche x et le booléen $x == 1$. Commenter.
 2. Écrire une boucle **while** qui ajoute 0,1 à x tant que $x < 1$ et compte les tours. Combien de tours obtient-on ? Pourquoi ?
 3. Que se passe-t-il si la condition est $x != 1$? Écrire cette boucle avec un garde-fou de 100 tours et observer.
 4. Proposer deux corrections : l'une avec une tolérance, l'autre en ne manipulant que des entiers.
-

Exercice 11 (★★★★☆) Les chiffres d'un entier

Pour un entier $n \geq 1$ écrit en base dix, $n \% 10$ est son chiffre des unités et $n // 10$ est l'entier obtenu en effaçant ce chiffre.

1. Écrire un script qui calcule la somme des chiffres de $n = 2025$ à l'aide d'une boucle **while**.
 2. Écrire un script qui compte le nombre de chiffres de n .
 3. Vérifier par un script que, pour tout $n \in \llbracket 1, 10\,000 \rrbracket$, n est divisible par 9 si et seulement si la somme de ses chiffres l'est. On n'affichera qu'un booléen.
 4. Écrire un script qui construit le *miroir* de n , c'est-à-dire l'entier obtenu en lisant ses chiffres de droite à gauche : le miroir de 12345 est 54321.
-

Exercice 12 (★★★★) Le problème des pièces

On dispose de pièces de 7 et de 5 unités, en quantité illimitée. Un montant $n \in \mathbb{N}$ est *atteignable* s'il existe des entiers naturels x et y tels que $n = 7x + 5y$.

1. Écrire un script qui, pour $n = 23$, parcourt tous les couples (x, y) possibles avec deux boucles imbriquées et affiche une décomposition si elle existe, ou indique que n n'est pas atteignable. On précisera les bornes des boucles.
2. Afficher tous les montants non atteignables entre 0 et 40. Quel est le plus grand ?

- 3.** Reprendre la question précédente avec des pièces de a et de $b = a + 1$ unités pour a allant de 2 à 7, en cherchant le plus grand montant non atteignable parmi $\llbracket 0, ab - 1 \rrbracket$. Conjecturer une formule en fonction de a et b .
-