

Correction TP Python sur France-ioi.org

2GT10

27 octobre 2021



---

# Table des matières

---

<b>Table des matières</b>	<b>3</b>
<b>1 Niveau 1</b>	<b>5</b>
1 Affichage de texte, suite d'instructions . . . . .	5
2 Répétitions d'instructions . . . . .	6
3 Calculs et découverte des variables . . . . .	10
4 Lecture de l'entrée . . . . .	12



---

# CHAPITRE 1:

## Niveau 1

---

### 1 Affichage de texte, suite d'instructions

**Exercice 1.1.** Écrivez un programme permettant au robot d'afficher précisément le texte "Hello world!".

```
1 print("Hello world!")
```

**Exercice 1.2.** Camthalion souhaite que son programme affiche :

Coucou!  
Je m'appelle Camthalion  
Ma devise est 'Parler peu mais parler bien'.

```
1 print("Coucou !")  
2 print("Je m'appelle Camthalion")  
3 print("Ma devise est 'Parler peu mais parler bien'.")
```

**Exercice 1.3.** Écrivez un programme qui affiche exactement le texte qui suit :

Tout droit tu grimperas,  
La clé tu trouveras,  
Habile tu seras,  
Quand tu les porteras,  
Et avec le chef tu reviendras!

```
1 print("Tout droit tu grimperas,")  
2 print("La clé tu trouveras,")  
3 print("Habile tu seras,")  
4 print("Quand tu les porteras,")  
5 print("Et avec le chef tu reviendras !")
```

**Exercice 1.4. Dans le fourré**

Diriger le robot à travers le labyrinthe

```
1 from robot import *
2
3 haut()
4 haut()
5 haut()
6 droite()
7 droite()
8 bas()
9 bas()
10 droite()
```

### Exercice 1.5. Empilement de cylindres

```
1 from robot import *
2
3 deplacer(1, 2)
4 deplacer(1, 3)
5 deplacer(2, 3)
6 deplacer(1, 2)
7 deplacer(3, 1)
8 deplacer(3, 2)
9 deplacer(1, 2)
10 deplacer(1, 3)
11 deplacer(2, 3)
12 deplacer(2, 1)
13 deplacer(3, 1)
14 deplacer(2, 3)
15 deplacer(1, 2)
16 deplacer(1, 3)
17 deplacer(2, 3)
```

### Exercice 1.6 (Recette secrète). Obtenir 4 litres avec 1 bol de 5 litres et 1 bol de 3 litres.

```
1 from robot import *
2
3 remplir(5)
4 transferer(5, 3)
5 vider(3)
6 transferer(5, 3)
7 remplir(5)
8 transferer(5, 3)
```

## 2 Répétitions d'instructions

### Exercice 2.1 (Punition).

```
1 for loop in range(135):
2     print("Je dois respecter le Grand Sorcier.")
```

**Exercice 2.2** (Mathématiques de base). Correction des erreurs contenues dans le code :

```
1 for loop in range(13)
2     print("9 * 8 = 72")
```

On observe plusieurs problèmes :

- le ":" est manquant à la fin de la première ligne
- il manque la tabulation de la deuxième ligne. Rappelons que la tabulation sert à montrer que la deuxième ligne appartient à la boucle.
- Dans le "print" de la deuxième ligne les "double quotes" ou "guillemets" ne sont pas fermés.

```
1 for loop in range(13):
2     print("9 * 8 = 72")
```

**Exercice 2.3** (Transport d'eau).

```
1 from robot import *
2
3 gauche()
4 gauche()
5 print("Bonjour, laissez-moi vous aider")
6 ramasser()
7 for loop in range(32):
8     droite()
9     déposer()
```

**Exercice 2.4** (Le secret du Goma).

```
1 from robot import *
2
3 for loop in range(0, 15):
4     # ceci est un commentaire
5     droite()
6     ramasser()
7
8 droite()
9 déposer()
```

**Exercice 2.5** (Sisyphé). *Point culture* : Le rocher de Sisyphé est une expression bien ancrée dans la langue française qui signifie « un travail difficile, toujours recommencé, interminable ». Elle provient de la mythologie grecque où Sisyphé, pour avoir trompé la mort, est condamné à pousser jusqu'au sommet d'une colline un lourd rocher qui en redescend chaque fois.

En bonus pour le BAC de Français/Philosophie : Le Mythe de Sisyphé est un essai écrit par Albert Camus en 1942.

```
1 from robot import *
2
3 for montee in range(0, 21):
4     haut()
5     droite()
```

```

6
7 for descente in range(0, 21):
8     gauche()
9     bas()

```

### Exercice 2.6. Page d'écriture



#### Astuce:

*Dans cet exercice, il faut créer une chaîne de caractères contenant les 30 lettres puis l'afficher une fois celle-ci entièrement constituée*

Une première solution possible

```

1 ligne_a = ""
2 for a in range(0, 30):
3     ligne_a = ligne_a + "a_"
4     print(ligne_a)
5
6 ligne_b = ""
7 for b in range(0, 30):
8     ligne_b = ligne_b + "b_"
9     print(ligne_b)
10
11 ligne_c = ""
12 for c in range(0, 30):
13     ligne_c = ligne_c + "c_"
14     print(ligne_c)

```

Une solution en utilisant les listes en Python (pour les pros)

```

1 for lettre in ['a', 'b', 'c']:
2     ligne = ""
3     for loop in range(0, 30):
4         ligne = ligne + lettre + "_"
5
6     print(ligne)

```

### Exercice 2.7 (Jeu de Dames).

```

1 for loop in range(20):
2     ligne1 = ""
3     for x in range(20):
4         ligne1 = ligne1 + "OX"
5
6     ligne2 = ""
7     for x in range(20):
8         ligne2 = ligne2 + "XO"
9
10 print(ligne1)
11 print(ligne2)

```

### Exercice 2.8 (Mont Kailash).

```

1 from robot import *
2
3 for tour in range(108):
4
5     for up in range(13):

```



```
6     haut ()
7
8     for right in range(13):
9         droite()
10
11    for down in range(13):
12        bas()
13
14    for left in range(13):
15        gauche()
```

### | Exercice 2.9 (Vendanges).

```
1  from robot import *
2
3  for loop in range(20):
4      ramasser()
5      for right in range(15):
6          droite()
7
8      déposer()
9      for left in range(15):
10         gauche()
```

### | Exercice 2.10 (Le Grand Evènement).

```
1  from robot import *
2
3  haut ()
4  for loop in range(4):
5
6      for up in range(8):
7          haut ()
8          # ...
9      droite()
10     for down in range(8):
11         bas()
12         # ...
13
14     droite()
15     # ...
16
17     # dernier aller-retour en hauteur
18     for up in range(8):
19         haut ()
20         # ...
21     droite()
22     for down in range(9):
23         bas()
24         # ...
25
```

```

26 for left in range(9):
27     gauche()
28     # ...

```

### 3 Calculs et découverte des variables

**Exercice 3.1** (Réponds!). Afficher le nombre 42

```

1 print(42)

```

**Exercice 3.2** (L'éclipse). Vous savez que l'éclipse aura lieu le 12581e jour et que la date actuelle est le 11937e jour. Votre programme doit calculer et afficher le nombre de jours qu'il faudra patienter avant de pouvoir admirer l'éclipse.

```

1 print(12581 - 11937)

```

**Exercice 3.3** (Bonbons pour tout le monde!). L'école est formée de 4 classes, constituées respectivement de 25, 30, 27 et 22 élèves. Cependant, 8 élèves sont absents aujourd'hui. Sachant que chaque élève présent doit recevoir 3 bonbons, écrivez un programme qui calcule puis affiche le nombre total de bonbons nécessaires.

```

1 print(3*(25+27+30+22-8))

```

Un autre programme python un peu plus élaboré :

```

1 nb_eleves = 25 + 30 + 27 + 22
2 nb_absents = 8
3 nb_bonbons = 3 * (nb_eleves - nb_absents)
4 print(nb_bonbons)

```

**Exercice 3.4** (L'algoréathlon).

```

1 distance = 2 + 34 + 6 # distance parcourue en 1 jour
2 print(distance, distance*2, distance*3)

```

**Exercice 3.5** (Cour de récréation).

```

1 longueur_cote = 5*17 + 2*7 + 5 + 2*2
2 aire_cour = longueur_cote * longueur_cote
3 perimetre_cour = 4 * longueur_cote
4
5 print(aire_cour)
6 print(perimetre_cour)

```

**Exercice 3.6** (Une partie de cache-cache).

```
1 for compt in range(1, 101):
2     print(compt)
3
4     print("J'arrive !")
```

**Exercice 3.7** (Progresser par l'erreur).

```
1 print("V")
2 print("V")
3 print("I")
4 print("I")
5 print("V")
6 print("I")
7 print("I")
```

**Exercice 3.8** (Décollage de fusée).

```
1 for compt in range(0, 101):
2     print(100 - compt)
3
4     print("Décollage !")
```

**Exercice 3.9** (Invasion de batraciens). Sachant qu'il y a actuellement 1337 crapauds et que leur nombre double chaque semaine, votre programme devra afficher le nombre de crapauds qu'il y aura après la 12e semaine.

```
1 nb_crapauds = 1337
2 for week in range(0, 12):
3     nb_crapauds = 2*nb_crapauds
4
5     print(nb_crapauds)
```

**Exercice 3.10** (Kermesse).

```
1 bonbons = 0
2 for tir in range(1, 51):
3     bonbons = bonbons + tir
4     print(bonbons)
```

**Exercice 3.11** (Course avec les enfants).

```
1 from robot import *
2
3 for rank in range(1, 11):
4
5     for right in range(0, rank):
6         droite()
7     # ...
```

```

8     ramasser()
9     for left in range(0, rank):
10         gauche()
11         # ...
12         déposer()
13         # ...

```

**Exercice 3.12** (Construction d’une pyramide). L’objectif est de construire une tour à l’aide de petits cubes en bois, sachant que la forme de cette tour consiste en un ensemble de grands cubes placés les uns au-dessus des autres. La base de la tour est un cube de taille  $17 \times 17 \times 17$ , c’est-à-dire composé de  $17 \times 17 \times 17 = 4913$  petits cubes. Sur ce cube est posé un autre cube de taille  $15 \times 15 \times 15$ . Au-dessus de ce dernier se trouve un cube de taille  $13 \times 13 \times 13$ . La tour continue ainsi jusqu’à atteindre le sommet, qui consiste en un cube de taille  $1 \times 1 \times 1$ .

Votre programme doit calculer et afficher le nombre total de petits cubes nécessaires pour construire la pyramide. Effectuez les calculs dans le programme en y intégrant une boucle.

```

1  cubes = 0
2  for elem in range(0, 9): # on utilise la def d'un nombre impair p = 2k + 1
3      cubes = cubes + (2*elem + 1) * (2*elem + 1) * (2 *elem + 1)
4      # ...
5  print(cubes)

```

**Exercice 3.13** (Table de multiplication). Ecrivez un programme qui affiche une table de multiplication allant jusqu’à 20 fois 20

```

1  for base in range(1, 21):
2      for num in range(1, 21):
3          print(base*num, end=' ')
4          # ...
5      print()
6      # ...

```

## 4 Lecture de l’entrée

**Exercice 4.1.** Récoltes Votre programme doit lire un entier, qui représente la longueur du côté d’un champ carré en mètres. Il doit ensuite afficher la masse que l’on pourra récolter de ce champ si l’on suppose que la production sera de 23 kg par mètre carré.

**Solution 4.1.1.** Ici il faut se souvenir de l’aire d’un carré qui est  $\mathcal{A} = c \times c$ . et on demande à l’utilisateur d’entrer la valeur de  $c$ .

```

1  longueur = int(input())
2  print(longueur*longueur*23)

```

**Exercice 4.2.** Retraite spirituelle Votre programme devra lire un entier : le nombre de jours que dure la randonnée. Ensuite, il devra afficher le nombre de fois que l’incantation est répétée, sachant qu’elle est prononcée une fois par seconde pendant 16 heures par jour (les 8 autres heures, on dort!).

On rappelle qu’il y a 3600 seconde dans 1 heure.

```

1 jours = int(input())
2 print(jours*3600*16)

```

**Exercice 4.3.** Age des petits-enfants Votre programme doit être une version corrigée du programme ci-dessous, sachant qu'il vous faut changer le moins de choses possible.

```

1 âgeCadet = int(input)
2 âgeAîné = input()
3 différence = âgeAîné - âgeDuCadet
4 print(différence)

```

```

1 ageCadet = int(input())
2 ageAine = int(input())
3 difference = ageAine - ageCadet
4 print(difference)

```

**Exercice 4.4.** Encore des punitions Votre programme doit lire un entier, le nombre de lignes souhaité, et écrira autant de fois que demandé la phrase « Je dois suivre en cours ».

```

1 nb_lines = int(input())
2 for i in range(0, nb_lines):
3     print("Je dois suivre en cours")

```

**Exercice 4.5.** Graduation de thermomètres Étant données deux températures entières tempMin et tempMax, votre programme doit afficher toutes les températures (entières) comprises entre les deux, bornes incluses.

```

1 tempMin = int(input())
2 tempMax = int(input())
3
4 for temp in range(tempMin, tempMax + 1):
5     print(temp)

```

**Exercice 4.6.** Jeu de calcul mental Un nombre de départ va être donné par le chef du village. La personne qui suit doit le multiplier par 2, puis la suivante doit multiplier le nombre obtenu par 3, celle d'encore après doit multiplier le résultat par 4... jusqu'à ce que les nbNombres calculs aient été effectués.

Le chef a choisi le nombre 66 pour démarrer le jeu. Votre programme lira l'entier nbNombres, la quantité de nombres attendue par le jeu (nombre de départ inclus). Il devra ensuite afficher tous les nombres de la partie afin de vous rendre imbattable !

```

1 nbNombres = int(input())
2 depart = 66
3
4 for mult in range(1, nbNombres + 1):
5     depart = depart * mult
6     print(depart)

```

**Exercice 4.7.** La Grande Braderie Il y a trois entiers à lire : la position de départ positionDepart, la largeur d'un emplacement largeurEmplacement et le nombre de vendeurs nbVendeurs.

Vous devez afficher une suite de nombres, partant de positionDepart et augmentant de largeurEmplacement à chaque fois. Il y a au total nbVendeurs augmentations à faire. Vous devez afficher la valeur de chacun des nombres de la suite.

```

1 positionDepart = int(input())
2 largeurEmplacement = int(input())
3 nbVendeurs = int(input())
4
5 print(positionDepart)
6 for i in range(0, nbVendeurs):
7     positionDepart = positionDepart + largeurEmplacement
8     print(positionDepart)

```

**Exercice 4.8.** Bétail Votre programme doit lire 20 entiers puis afficher la somme de tous ces entiers.

```

1 somme = 0
2 for i in range(0, 20):
3     entree = int(input())
4     somme = somme + entree
5
6 print(somme)

```

**Exercice 4.9.** Socles pour statues Un socle est ainsi constitué d'étages, chaque étage ayant une hauteur égale à une unité et une base carrée. Le côté des carrés diminue de une unité à chaque étage.

Votre programme doit lire deux entiers, représentant respectivement la largeur du socle au niveau du sol et la largeur du socle au niveau de la face supérieure du socle. Il doit ensuite calculer et afficher le volume du socle.

Rappelons le volume d'un socle à base carrée :  $\mathcal{V} = c \times c \times l$

```

1 coteSocle1 = int(input())
2 coteSocleFin = int(input())
3 volume = 0
4
5 for coteSocle in range(coteSocleFin, coteSocle1 + 1):
6     volume = volume + coteSocle * coteSocle * 1
7
8 print(volume)

```

**Exercice 4.10.** Le plus beau Karva Votre programme doit d'abord lire le nombre de Karvas en compétition. Ensuite, pour chaque Karva, il doit :

- lire 4 entiers : son poids, son âge, la longueur de ses cornes et la hauteur au garrot ;
- afficher sa note, sachant qu'elle s'obtient en multipliant la longueur des cornes par la hauteur au garrot, valeur à laquelle on ajoute le poids.

1