

## Programmer en Python.

### I Le langage Python pour les maths.

Il est possible de tester les instructions en allant sur le [laboratoire du Livrescolaire](#).

Python est un langage informatique qui permet, entre autres, de faire des mathématiques.

#### Addition.

```
>>>2+3  
5
```

#### Soustraction.

```
>>>2-3  
-1
```

#### Multiplication.

```
>>>2*(-3)  
-6
```

#### Division.

Pas question de fraction pour Python les divisions s'écrivent en ligne.

```
>>>2/3  
0.6666666666666666
```

Remarques.

1. Python ne donne pas d'écriture sous forme de fraction irréductible.
2. L'indicateur de position dans l'écriture décimale n'est pas la virgule, comme en français, mais le point comme en anglais.

3. Lorsque Python utilise la notation décimale cela signifie qu'il utilise des valeurs approchées.

```
>>>3*2/3  
2.0
```

### Puissance.

```
>>>3**2  
9
```

### Affectation.

Il est possible de donner un nom à un calcul cela s'appelle l'*affectation*. L'affectation s'indique avec le signe égal. Pour afficher le résultat du calcul il faut utiliser la fonction `print()`.

```
>>>A=3*(5-1)  
>>>print(A)  
12
```

Remarques.

1. Affecter c'est enregistrer le calcul en lui donnant un nom.
2. L'affectation n'indique pas le résultat.
3. Il est alors possible d'utiliser A dans d'autres calculs.

```
>>>2-A/3  
-2.0
```

4. La lettre A dans laquelle est enregistré le résultat du calcul est appelée une *variable*.

## Exercices.

### Exercice 1.

Donnez le résultat retourné par Python avec les instructions suivantes.

1. `2 - (3 - 1)`

4. `1/3+1/3+1/3`

2. `4 / (6 - 1)`

5. `(2 - 3) * 2 ** 3`

3. `3+4**2`

6. `1 . 1 * 12`

### Exercice 2.

Dites pourquoi les instructions suivantes conduiront à message d'erreur et proposez une correction.

1. `>>>3/0`

3. `>>>2+1=A`

2. `>>>-3 -*2`

4. `>>>A=2+1  
>>>2A`

## II Un programme.

Nous aurons parfois besoin de faire plusieurs calculs pour arriver à nos fins. Nous dirons qu'il s'agit d'un algorithme.

Pour réaliser un algorithme nous devons faire diverses instructions ce qui en informatique s'appelle un programme.

Le fichier informatique qui contient un programme est appelé un script.

Dans un script Python on indique une nouvelle instruction en passant à la ligne.

### Exercice 3.

Exercices 29, 30 et 32 page 31 du manuel Indice.

## III Les résultats renvoyés par un programme.

Les programmes informatiques ne renvoient pas des réponses exactes mais des valeurs approchées. Le mathématicien doit donc toujours rester circonspect quand au résultat donné par une machine (ordinateur ou calculatrice).

Nous dirons que les résultats informatiques sont des *conjectures* : il ne sont pas absolument fiables et doivent être démontrés.

## IV Choisir un calcul.

Il est possible de prévoir deux calculs possibles dans un programme et de n'effectuer l'un ou l'autre que si une certaine condition est vérifiée. On retrouve la même formulation qu'en langage courant « si ..., alors, ..., sinon ... ».

Exemples.

1. On souhaite obtenir 1 si le nombre enregistré dans la lettre `A` est strictement supérieur à 2 et sinon `-1`.

```
if A>2:
    print(1)
else:
    print(-1)
```

2. Voici un programme qui additionne les entiers naturels jusqu'à 5.

Remarques.

1. La condition vérifiée est indiquée juste après le `if` et est suivie de « : ».
2. Le `else` est suivi de « : ».
3. Les instructions conditionnelles (celles qui ne sont pas forcément effectuées) sont décalées par rapport à la marge. Ce décalage est appelé en informatique une *indentation*.

### Exercice 4.

Voici un programme écrit en Python.

```
if n<200:
    M=0.11*n
else:
    M=1.1*n
```

Donnez la valeur enregistrée dans `M` lorsque

1.  $n = 10$ ,
2.  $n = -120$ ,
3.  $n = 199,99$ ,
4.  $n = 200,0001$ .
5.  $n = 200$ .

Exercice 5.

Un manuel numérique est proposé au tarif de 8 euros par élève si le lycée a moins de 300 élèves et 7,5 euros sinon.

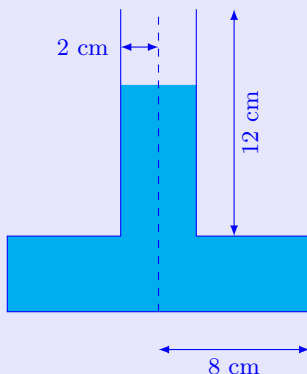
Complétez le programme suivant dans lequel  $N$  est le nombre d'élèves et  $P$  est le prix correspondant payé pour le manuel numérique.

```
if n<301:  
    P= .....  
else:  
    P= .....
```

Exercice 6.

### Exercice 7.

Une carafe est constituée d'un cylindre de révolution de hauteur 4 cm et de rayon 8 cm, surmonté d'un autre cylindre de révolution de hauteur 12 cm et de rayon 2 cm.



1. Calculez, en  $\text{cm}^3$ , le volume d'eau dans la carafe quand la hauteur d'eau est  $x$  centimètres, où  $x$  est un nombre réel compris entre 0 et 4.
2. Montrez que le volume d'eau dans la carafe quand la hauteur d'eau dans la carafe est  $x$  centimètres, où  $x$  est un nombre réel compris entre 4 et 16, est égale à  $4\pi x + 240\pi$  (en  $\text{cm}^3$ ).
3. Complétez le programme ci-dessous afin qu'il retourne le volume d'eau dans la carafe (en  $\text{cm}^3$ ) selon la hauteur d'eau  $h$  versée.

```

if .....:
    print(.....)
else:
    print(.....)
    
```

### Correction exercice 7

1. Calculons le volume  $V_1(x)$  d'eau dans la carafe si  $x \leq 4$ .

Si  $x \leq 4$  alors seul le cylindre du bas est rempli. Puisque son rayon est  $r_1 = 8$  cm et sa hauteur  $h_1 = x$  cm son volume est :

$$\begin{aligned}
 V_1(x) &= \pi \times r_1^2 \times h_1 \\
 &= \pi \times 8^2 \times x \text{ cm}^3
 \end{aligned}$$

$$V_1(x) = 64\pi x \text{ cm}^3.$$

2. Calculons le volume  $V(x)$  d'eau dans la carafe si  $x > 4$ .

Si  $x > 4$  alors le cylindre du bas est entièrement rempli et celui du haut est rempli jusqu'à une hauteur  $h_2 = x - 4$  cm. Puisque le rayon du cylindre du haut est  $r_3 = 2$  cm et sa hauteur  $h_2 = x - 4$  cm son volume est :

$$\begin{aligned} V_2(x) &= \pi \times r_2^2 \times h_2 \\ &= \pi \times 2^2 \times (x - 4) \text{ cm}^3 \\ &= 4\pi(x - 4) \text{ cm}^3 \\ &= (4\pi \times x - 4\pi \times 4) \text{ cm}^3 \\ &= 4\pi x - 16\pi \text{ cm}^3 \end{aligned}$$

Nous en déduisons le volume total d'eau en  $\text{cm}^3$  dans la carafe :

$$\begin{aligned} V &= V_1(4) + V_2(x) \\ &= 64\pi \times 4 + 4\pi x - 16\pi \\ &= 4\pi x + 256\pi - 16\pi \end{aligned}$$

Si  $x > 4$ , alors :  $V(x) = 4\pi x + 240\pi$ .

3.

```
if x<=4:
    print (64*3.1416*x)
else:
    print (4*3.1416*x
+240*3.1416)
```

## V Répéter des calculs.

Un des intérêts majeur de l'informatique est de permettre la répétition d'un grand nombre de calculs numériques.

Nous allons voir deux façons de procéder à de telles répétitions.

### Boucle for.

Une *boucle for* (ou boucle bornée) permet de recommencer une instruction un nombre de fois choisi à l'avance.

La boucle for utilise une variable, le plus souvent notée  $k$ , qui sert de compteur de répétition.

Exemples.

1. Un salaire de 1200 € augmente de 30 € par mois pendant un an. Voici un algorithme calculant le salaire  $S$  au bout d'un an.

```
S=1200
for k in range(1,12+1):
    S=S+30
print(S)
```

Pour vérifier que cet algorithme fonctionne nous allons faire le tableau d'état des variables.

2. Nous souhaitons maintenant calculer la somme des 6 premiers entiers naturels.

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

### Exercice 8.

En prévision d'une course de vélo, Fanny suit le programme d'entraînement suivant sur douze samedi : elle parcourt 25 kilomètres le premier samedi, puis augmente chaque semaine de 11 kilomètres la distance parcourue.

1. Déterminez la distance  $d$  parcourue le deuxième samedi et la distance totale  $t$  parcourue au bout de deux samedis d'entraînement.
2. Complétez l'algorithme ci-dessous afin que la variable  $t$  contiennent en fin d'algorithme la distance totale parcourue au bout des douze samedis d'entraînement.

```
d=.....
t=.....
for k in range(2,12+1):
    d=.....
    t=.....
print(t)
```



## Exercice 9.

On empile des sphères formant ainsi une pyramide base carrée.

1. Déterminez le nombre de sphères nécessaires pour constituer une pyramide de deux niveaux.
2. Déterminez le nombre de sphères nécessaires pour constituer une pyramide de cinq niveaux.
3. Sachant qu'il faut 140 sphères pour construire une pyramide de sept niveaux, déterminez le nombre de sphères nécessaire pour construire une pyramide de huit niveaux.
4. Complétez le programme ci-dessous pour qu'il retourne le nombre  $b$  de sphères nécessaire pour constituer une pyramide comportant  $n$  niveaux.

```
n = .....
for i in range(1, n+1):
    .....
.....
```

**Boucle While.**

Une *boucle while* (ou boucle non bornée) permet de recommencer une instruction autant de fois que nécessaire pour qu'une condition (choisie à l'avance) devienne fausse. Exemple de l'élève : tant que ses notes sont faibles l'élève doit augmenter son travail.

Une boucle while n'a pas de compteur de répétition ; nous serons parfois amenés à en créer un. Si la condition est mal choisie la boucle est infinie (« infinite loop ») ; on parle alors de bug.

La condition, qui décide si la répétition doit se poursuivre ou pas, est une phrase (mathématique) qui est soit vraie soit fausse. Cette condition est appelée un *booléen*. Si la phrase (le booléen) est vraie on lui associe la valeur 1 sinon on lui associe la valeur 0.

**Exemples.**

1. On considère un carré d'aire  $1 \text{ cm}^2$  dont la surface est doublée autant de fois que nécessaire pour qu'elle dépasse  $30 \text{ cm}^2$ .

```
S=1
while S<30:
    S=S*2
print(S)
```

2. Si dans l'exemple précédent nous souhaitons maintenant savoir combien de fois le carré a doublé pour dépasser les 30 cm<sup>2</sup> nous pouvons ajouter un compteur dans la boucle.

```
S=1
k=0
while S<30:
    S=S*2
    k=k+1
print(S)
```

### Exercice 10.

Dans l'algorithme ci-dessous, la variable  $m$  contient un nombre entier. Selon la question posée, elle prendra différentes valeurs en début d'algorithme.

```
m=.....
while m>9:
    m=m-9
```

- Si en début d'algorithme la variable  $m$  reçoit 5, quelle valeur contient-elle en fin d'algorithme?
  - Si en début d'algorithme, la variable  $m$  reçoit 115, quelle valeur contient-elle en fin d'algorithme?
- Dans cette question, la valeur que reçoit la variable  $m$  en début d'algorithme est 30. Complétez le tableau suivant et déduisez-en la valeur  $m$  après l'exécution de cet algorithme.

Étapes	$m$	Condition vérifiée?
Avant la boucle	30	oui
1er passage dans la boucle	.	.....
2e passage dans la boucle	.	.....
.....	.	.....

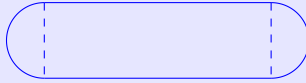
En fin d'algorithme la valeur de  $m$  est : .....

- La variable  $m$  peut-elle contenir 21 en fin d'algorithme?
- Quelles valeurs peut recevoir la variable  $m$  en début d'algorithme pour que la boucle while soit parcourue?

## Exercice 11.

Une ampoule destinée à recevoir du sérum est constituée d'un corps cylindrique de hauteur 100 mm et de deux demi-sphères de rayon  $r$  millimètres. On veut déterminer à partir de quelle valeur entière du rayon, exprimée en millimètres, le volume d l'ampoule dépassera 20 centilitres.

1. Montrez que le volume de l'ampoule, exprimée en  $\text{mm}^3$ , est égale à  $100\pi r^2 + \frac{4}{3}\pi r^3$ .



2. Complétez l'algorithme ci-dessous afin qu'en fin d'algorithme, la variable  $r$  ait la valeur cherchée.

```
r=0
v=0
while ...:
    r= ...
    v= ...
print( ... )
```

3. Programmez le précédente algorithme et répondez à la question.

## Exercice 12.

### Exercice 13.

Une entreprise de forage creuse des puits dans le désert afin d'atteindre le nappe d'eau phréatique. Cette entreprise facture le premier mètre creusé 100 €, le seconde mètre 140 € et ainsi de suite en augmentant le prix de chaque nouveau mètre creusé de 40 €.

1. Calculez le prix  $M$  du troisième mètre creusé, puis le prix total  $s$  d'un puits de trois mètres de profondeur.
2. Recopiez puis complétez le programme ci-dessous, dans lequel variable  $p$  désigne la profondeur du puits (en mètres), afin qu'il retourne le prix en euros de ce puits. Utilisez ce programme pour déterminer le prix d'un puits de 8 mètres de profondeur, puis celle d'un puits de 12 mètres de profondeur.

```
p = .....
M = 100
s = 100
n = 1
while n < p:
    M = M + 40
    s = .....
    n = .....
print (...)
```

3. Une organisation humanitaire dispose d'un budget de 4000 €.
  - (a) En utilisant le programme de la question 2, déterminez la profondeur maximale d'un puits que peut financer l'organisation.
  - (b) Recopiez et complétez la fonction ci-dessous afin qu'elle renvoie cette profondeur maximale en prenant pour argument la somme maximale disponible.

```
max = .....
M = ...
s = ...
n = 0
while ...:
    .....
    .....
print(n)
```