
Algorithme et programmation

Python

maths-mde.fr

Table des matières

1	Mise en route	2
1.1	Qu'est-ce qu'un algorithme?	2
1.2	Les variables	2
1.2.1	Valeurs numériques	2
1.2.2	Chaînes de caractères	2
1.2.3	Les booléens	3
1.3	La syntaxe	3
2	Scratch avec Python	5
3	Travaux Pratiques	7
3.1	Afficher et affecter des valeurs	7
3.2	Objets de type simple	9
3.3	Objets de type composé	11
3.4	Fonctions	13
3.5	Structure conditionnelle	15

1 Mise en route

1.1 Qu'est-ce qu'un algorithme ?

Définition 1.1

Un **algorithme** est une structure logique comprenant une **succession d'instructions** à enchaîner dans l'ordre.

Les instructions peuvent être données en utilisant plusieurs types de langages.

Définition 1.2

Le **langage naturel** (c'est-à-dire en français avec peu de syntaxe spécifique) est utilisé notamment par le logiciel Algobox.



Les langages plus spécifiques sont nombreux et ont leur spécificité, parmi eux on peut citer, Scilab, Java, C++, PHP et **Python** que nous allons utiliser ici.

1.2 Les variables



Un programme informatique permet de manipuler des données, celles-ci consistant en une **suite finie d'instructions binaires**. Le programme utilise des variables de différents types afin de pouvoir mémoriser des informations qu'il utilisera plus tard.

À la variable, il faut affecter un nom, type (nombre, chaîne de caractères, booléen), une valeur et une adresse (l'emplacement dans la mémoire).

Pour affecter une valeur à une variable en Python, on écrit le nom, le symbole « = » et la valeur.

| **Exemple.** L'instruction `âge = 15` affecte à la **variable** « âge » la valeur « 15 ».

1.2.1 Valeurs numériques

Définition 1.3

La valeur peut être un nombre entier, **int**, ou réel (équivalent à « flottant »), **float**. On peut spécifier ou non le type de nombre à saisir, selon les besoins.

Exemple.

$x = y$	→ La valeur de droite y calculée et affectée dans la valeur de gauche x .
$x == y$	→ les nombres x et y sont égaux.
$(x, y) = (4, 7)$	→ On peut affecter deux valeurs à deux lettres simultanément.
$a = 8$	
$b = a$	→ b prend la valeur 8.
$c = b + 7$	→ c prend la valeur 15.



La virgule des décimaux est un point en langage Python.

1.2.2 Chaînes de caractères

Définition 1.4

Il s'agit d'une liste de lettres et/ou de chiffres, toujours écrite entre guillemets. Pour être affichée à l'écran, elle doit être introduite par la commande `input()`.

| **Exemple.** `input(« I est le milieu de [BC] »)`

On peut accoler deux chaînes de caractères en utilisant le symbole +, le faire se répéter en utilisant le symbole *.

Exemple.

```

abc = "code"
print(abc)      → affiche : code.
print("abc")    → affiche : abc.
abc + "d"       → b affiche : coded.

```

1.2.3 Les booléens

Définition 1.5

Le booléen est un type de variables qui ne peut prendre que deux valeurs : vrai (true) et faux (false). Il permet de tester des résultats.

Exemple.

```

abc = 5
print(a==5)    → affiche : true.
print(b==6)    → affiche : false.

```

Quelques opérations utilisées en Python

```

x = 3
y = 7          ou (x,y) = (3,7)
x + y          → somme
x - y          → différence
x*y            → produit
x/y = 0,428571 → division
y % x = 4      → reste de la division euclidienne de y par x
y // x = 1     → quotient de la division euclidienne de y par x
sqrt(x) = 1,732 → racine carrée de x
x**2 = 9       → x puissance 2.

```



En langage Python, certaines fonctions spécifiques telles que la racine carrée dans des bibliothèques. Il faut alors les importer avec la commande `from math import*`.

Exemple. `from math import sqrt` importe la fonction racine carrée.

1.3 La syntaxe

Python

Instructions	Affichage
<code>int = 25</code> → int indique un nombre entier	
<code>print(int)</code>	25
<code>print("int")</code>	int
→ les guillemets demandent ici d'afficher une chaîne de caractères et non une valeur	
<code>n = int(input("entrer votre âge :"))</code>	entrer votre âge :
→ la phrase "entrer votre âge" apparaît à l'affichage grâce à la fonction <code>input()</code>	
<code>x = float(input("entrer votre taille :"))</code>	entrer votre taille :
→ On demande ici un nombre réel (float)	

Exemple. L'algorithme ci-dessous demande la valeur d'un nombre et donne la valeur de son double.

```

x= float(input("entrer x"))
y=2*x
print(y)

```

Si on rentre la valeur 7, l'algorithme affichera 14.

Exercice 1 :

Qu'affichera le programme suivant ?

```
a=1
b=1
b=a+b
a=a-b
print(a)
```

Exercice 2 :

Expliquer de façon claire et concise ce que fait le programme suivant.

```
a=int(input("Entrez un entier"))
b=a
a=a*b
print(a)
```

Exercice 3 :

Qu'affichera le programme suivant ?

```
a=1
b=1
c=a+b
a=b+c
b=c+a
print(b)
```

Exercice 4 :

Que fait le programme suivant ?

```
a=int(input("Entrez un entier"))
b=a+a
a=a+b
print(a)
```

Exercice 5 :

Qu'affichera le programme suivant ?

```
a=3
for i in range(1,1001):
    a=a+0.5
print(a)
```

Exercice 6 :

Que fait le programme suivant ?

```
b=1
a=int(input("Entrez un nombre"))
for i in range(1,4):
    b=b*a
print(b)
```

Exercice 7 :

Complétez le programme suivant pour qu'il calcule la somme des inverses des entiers de 1 à 1000 :

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{1000}$$

```
somme=0
for i in range(1,1001):
    somme=somme+ .....
print(somme)
```

Exercice 8 :

Écrivez un programme qui demande à l'utilisateur d'entrer entier naturel non nul n et qui affiche la factorielle de n . La factorielle d'un entier n , que l'on note en mathématiques avec un point d'exclamation : « $n!$ » (lu « factorielle n ») est le produit de tous les entiers de 1 à n inclus.

Ainsi $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$.

Exercice 9 :

Qu'affichera le programme suivant ?

```
a=0
b=0
for i in range(0,4):
    a=a+b
    b=b+1
print(a)
```

Exercice 10 :

Qu'affichera le programme suivant ?

```
a=1
q=1+0.1**6
for i in range(1,10**6+1):
    a=q*a
print(a)
```

Exercice 11 :

Que fait le programme suivant ?

```
s=0
n=int(input("Entrez un entier naturel non nul."))
for i in range(1,n+1):
    s=s+i
print(s)
```

Exercice 12 :

Que fait le programme suivant ?

```
s=0
n=int(input("Entrez un entier naturel non nul."))
p=int(input("Entrez un second entier naturel non nul."))
for i in range(1,n+1):
    for j in range(1,p+1):
        s=s+1
print(s)
```

2 Scratch avec Python

Le module turtle permet de tracer facilement des dessins en Python. Il s'agit de commander une tortue à l'aide d'instructions simples comme « avancer », « tourner ». C'est le même principe qu'avec Scratch, avec toutefois des différences : tu ne déplaces plus des blocs, mais tu écris les instructions ; et en plus les instructions sont en anglais !

Principales commandes : `from turtle import *`

- `forward(longueur)` avance d'un certain nombre de pas.
- `backward(longueur)` recule.
- `right(angle)` tourne vers la droite (sans avancer) selon un angle donné en degrés.
- `left(angle)` tourne vers la gauche.
- `setheading(direction)` s'oriente dans une direction (0 = droite, 90 = haut, -90 = bas, 180 = gauche).
- `goto(x,y)` se déplace jusqu'au point (x, y).
- `setx(newx)` change la valeur de l'abscisse.
- `sety(newy)` change la valeur de l'ordonnée
- `down()` abaisse le stylo.
- `up()` relève le stylo.
- `width(epaisseur)` change l'épaisseur du trait.
- `color(couleur)` change la couleur : "red", "green", "blue", "orange", "purple"...
- `position()` renvoie la position (x, y) de la tortue.
- `heading()` renvoie la direction angle vers laquelle pointe la tortue
- `towards(x,y)` renvoie l'angle entre l'horizontale et le segment commençant à la tortue et finissant au point (x, y)
- `exitonclick()` termine le programme dès que l'on clique.

Activité 1 :

1. Ouvrez le logiciel Python.
2. Transcrivez le programme ci-dessous, l'exécutez et observez le dessin obtenu.

```
from turtle import *  
forward(120)  
left(90)  
forward(50)
```

3. Modifiez votre programme pour dessiner un carré.
4. Tapez le programme ci-dessous et comparez avec la question précédente. Selon vous, à quoi sert l'instruction sur la 2e ligne du programme ci-dessous qui s'appelle une boucle.

```
from turtle import *  
for i in range(4):  
    forward(120)  
    left(90)
```



ATTENTION à l'indentation des lignes (le décalage vers la droite) qui doit être respectée !

5. Modifiez votre programme pour dessiner un hexagone régulier, un triangle équilatéral et un octogone régulier.

Activité 2 :

1. Ouvrez le logiciel Python.
2. Transcrivez puis exécutez le programme ci-dessous. Détaillez l'action de chaque instruction en s'aidant des précités principales commandes.

```
from turtle import *
def triangle(longueur):
    for i in range(3):
        forward(longueur)
        left(120)
    up()
    goto(100,100)
    down()
    fillcolor('red')
    begin_fill()
    triangle(60)
    end_fill()
    up()
    goto(-100,-100)
    pensize(10)
    pencolor('green')
    down()
    triangle(120)
```

3. Créer des fonctions qui dessine un carré, un parallélogramme, un hexagone et un octogone.

3 Travaux Pratiques

3.1 Afficher et affecter des valeurs

Activité 1 :

1. Que veut dire le mot anglais print ?
2. (a) Écrire et exécuter le programme 1 et le programme 2 ci-après

Programme 1

```
a=2
b=3*a+5
print(b)
```

Programme 2

```
a=2
b=3*a+5
print("b")
```

- (b) Expliquer l'affichage du programme 1.....
- (c) Pourquoi n'a-t-on pas le même affichage pour le programme 1 et pour le programme 2? Que permettent les guillemets?
.....

- (d) Modifier la dernière ligne du programme 2 en `print("b=",b)` puis expliquer ce que permet la virgule.
.....

3. (a) Sans l'écrire, dire ce que va afficher le programme 3 ci-dessous.

```
x=5
y=12
x=3*x+2*y
y=5*y-12*x
print(y)
```

- (b) L'écrire, l'exécuter et vérifier l'affichage.

Activité 2 :

1. (a) Sans l'écrire, donner l'affichage du programme ci-dessous.

```
a="30"
print(a)
a=a*2
print(a)
```

- (b) Écrire le programme et vérifier la réponse à la question précédente.
.....

2. Modifier le susdit programme, en remplaçant la première ligne par `a = "bonjour"`. Qu'obtient-on?
.....

Remarque : Dans le langage PYTHON, lorsque l'on écrit une valeur *z* entre guillemets " (ou apostrophes) lors d'une affectation, alors cette variable est de type *str* pour chaîne de caractères (string en anglais) c'est-à-dire que sa valeur *z* est le texte entre guillemets (même si ce texte est un nombre, comme dans la première version du programme ci-dessus).

3. (a) Écrire et exécuter les trois programmes suivants.

```
a="3"
a=int(a)
b=a*2
print(b)
```

```
a=3
a=str(a)
b=a*2
print(b)
```

```
a="bonjour"
a=int(a)
b=a*2
print(b)
```

.....
.....
(b) Expliquer chacun des affichages.
.....
.....
.....

Activité 3 :

1. (a) Écrire et exécuter le programme suivant.

```
for i in range(1,6) :  
    print("Je répète 6 fois un message")
```

(b) Quelle incohérence semble-t-il y avoir entre le susdit programme et son affichage ?
.....

2. Écrire et exécuter le programme ci-dessous.

```
for i in range(1,6) :  
    print("Je répète 6 fois un message",i)
```

3. L'un des deux algorithmes ci-dessous correspond au dernier programme. Lequel ?

Algorithme 1

Pour i allant de 1 à 6
Afficher "Je répète 6 fois un message"
Fin pour

Algorithme 2

Pour i allant de 1 à 6
Afficher "Je répète 6 fois un message"
Fin pour

.....
.....
4. Soit a et b deux entiers, recopier et compléter les pointillés dans le tableau suivant.

En langage naturel	En PYTHON
Pour i allant de ... à ...	for i in range(a,b)

5. Dans le dernier programme, remplacer tous les i par des k et exécuter de nouveau le programme. Que remarque-t-on ?
.....

6. (a) Sans utiliser l'ordinateur, dans le programme 16, quel serait l'affichage si l'on ajoutait une 3ème ligne print("au revoir") indentée comme la 2ème ligne ? non indentée ?

(b) Vérifier avec l'ordinateur.
.....
.....
.....

7. Écrire un programme qui affiche tous les carrés des nombres entiers de 0 à 1000, c'est-à-dire 0 ; 1 ; 4 ; 9 ; ; 1000000).

3.2 Objets de type simple

Exercice 1 :

Que renvoie la console Python quand on saisit ?

`type(1)`

`type(3.14)`

`type(-2020)`

`type(1+1==2)`

`type(1.0)`

`type(5<4)`

Quels sont les 3 types d'objets découverts à travers ces exemples ?

.....

Exercice 2 :

Que renvoie la console Python lorsqu'on saisit les opérations suivantes ?

`20+3`

`0//3`

`20-3`

`20%3`

`20/3`

`round(20/3,2)`

`20*3`

`20**3`

Retrouver la syntaxe Python des opérations arithmétiques élémentaires sur les nombres en complétant le tableau suivant :

Opérations	Syntaxe Python
Addition de a et b	
Multiplication de a par b	
Division de a par b	
Exponentiation ab	
Quotient de la division euclidienne de a par b / Reste de la division euclidienne de a par b	
Valeur arrondie de a à p décimales près	

Exercice 3 :

Quels sont les résultats des opérations suivantes dans la console Python ?

`2+3-5`

`(2+3)/5`

`2+3*5`

`2/(3+5)`

`2*3+5`

`5+2**3`

`(2+3)*5`

`5*2**3`

`2*3/5`

`(5+2)**3`

`2+3/5`

`(5*2)**3`

Exercice 4 :

Que renvoie la console Python quand on saisit les booléens suivants ?

<code>1==0</code>	<code>6/2==3</code>
<code>1!=0</code>	<code>2021%2==0</code>
<code>5>4</code>	<code>3**2==9</code>
<code>-5>-4</code>	<code>5*2**3</code>
<code>2**3==6</code>	<code>(5+2)**3</code>
<code>6%2==3</code>	<code>(5*2)**3</code>

Exercice 5 :

Donner le contenu des variables a et b à la fin des 3 programmes suivants :

<pre>a=2 b=3 a=a+1 b=2*a+3 b=a+b print(a) print(b)</pre>	<pre>a=2 b=3 a=a+1 b=a+b b=2*a+3 print(a) print(b)</pre>	<pre>a=2 b=3 b=2*a+3 a=a+1 b=a+b print(a) print(b)</pre>
--	--	--

.....

Les 3 programmes présentent les mêmes 5 affectations mais dans un ordre différent. Cela affecte-t-il le résultat final des contenus des variables a et b?

3.3 Objets de type composé

Exercice 1 :

Que renvoie la console Python quand on saisit ?

```
Prenom='Albert'  
Prenom[0]
```

```
L=[0,3,6,9,12]  
L[1]+L[2]
```

```
Prenom='Albert'  
Prenom[0]+'.'+'Einstein'?
```

```
L=[0,3,6,9,12]  
L[3]+L[5]
```

```
Prenom='Albert'  
T in Prenom
```

```
L=[0,3,6,9,12]  
2*L[1]==L[2]?
```

```
L=[0,3,6,9,12]  
L+[15]
```

```
L=[0,3,6,9,12]  
2*L[1]==L[2]
```

```
L=[0,3,6,9,12]  
type(L[3])
```

```
L=[0,3,6,9,12]  
L[1]+L[3] in L
```

Remarque : Le mot réservé *in* permet de tester si un élément (de type simple) est dans un objet de type composé.

Exercice 2 :

Trouver les valeurs inconnues.

```
range(4,20,4)
```

```
[a,b,c,d,e]
```

```
range(a,b,c)
```

```
[-6,-5,-4,-3,-2,-1]
```

```
range(1,10,2)
```

```
[a,b,c,d,e]
```

```
range(a,b,c)
```

```
[0,1,2,3,4,5]
```

Remarque : On rappelle que la fonction *list* permet d'afficher le contenu de *range* sous forme de liste.

Exercice 3 :

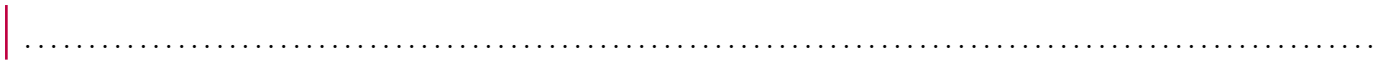
Donner le contenu des variables a, b et c à la fin des 5 affectations suivantes.

```
a=[5,10,15,20]  
b=a[1]+a[3]  
b=b%5  
a=[b]+a  
c=(b in a)
```

Exercice 4 :

Donner les valeurs prises par les différentes variables à la fin de ce programme.

```
x = 1  
y = 2  
z = 3  
x = x*y*z  
z = x + y + z  
y = x y z
```



3.4 Fonctions

Exercice 1 :

On peut écrire une fonction qui prend en entrées les trois notes en Mathématiques d'un élève (affectés d'un même coefficient) à un trimestre, et renvoie sa moyenne.

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

moyenne(10,9,14) Que renvoie la console Python dans ce cas ?

```
moyenne(10,9,14)
```

Exercice 2 :

On peut écrire une fonction qui prend en entrée un entier et renvoie vrai s'il est divisible par 7 et faux sinon.

```
def test(n):  
    r=n%7  
    return r==0
```

Que renvoie la console Python dans les deux cas ?

```
test(63)
```

```
test(22)
```

Exercice 3 :

On peut écrire une fonction qui prend en entrée un prix hors-tax (H.T.) en euros et renvoie la liste contenant : le prix H.T. pris en entrée, le montant d'une T.V.A. à 20% (taxe sur la valeur ajoutée qui correspond à 20% du prix), et le prix T.T.C. (toutes taxes comprises).

```
def prix(p):  
    t=0.2*p  
    TTC=p+t  
    return [p,t,TTC]
```

Que renvoie la console Python dans ce cas ?

```
taxe(60)
```

Exercice 4 :

1. Écrire dans l'éditeur la fonction.

```
def somme(a,b):  
    return a+b
```

2. Que renvoie la console Python dans les deux cas ?

```
somme(3,4)
```

```
Et somme(120,45)?
```

3. Quel est le rôle de cette fonction ?

Exercice 5 :

1. Écrire une fonction Python produit qui prend en entrées deux entiers et renvoie leur produit.
2. Tester-la dans la console en observant le résultat renvoyé par :

```
produit(3,4)
```

3. A-t-on besoin de la fonction produit pour connaître le résultat de produit(1234,1) ?

Exercice 6 :

1. Écrire une fonction Python `oppose` qui prend en entrée un entier relatif et renvoie son opposé.
2. Tester-la dans la console sur des entiers relatifs positifs et négatifs.
3. Que renvoie dans la console `oppose(oppose(1789))` ?

3.5 Structure conditionnelle

Exercice 1 :

On rappelle qu'à l'issue des épreuves du Bac, un candidat est :

- admis si sa moyenne générale est supérieure ou égale à 10 ;
- convoqué à l'oral de rattrapage si sa moyenne générale est inférieure à 10 mais supérieure ou égale à 8 ;
- recalé si sa moyenne générale est inférieure à 8.

On a écrit ci-dessous une fonction Python incomplète qui prend en entrée la moyenne générale à l'examen d'un candidat au Bac et renvoie son statut : admis , oral , recalé .

1. Compléter le programme suivant.

```
def Bac(m):  
    if      :  
        return "admis"  
    if      and      :  
        return "oral"  
    if      :  
        return "recalé"
```

2. Que renvoie la console lorsqu'on saisit ?

Bac(11.5)

3. Que renvoie la console lorsqu'on saisit ?

Bac(8.2)

4. Que renvoie la console lorsqu'on saisit ?

Bac(7.8)

Il est possible d'écrire une autre fonction, avec des instructions un peu différentes, mais qui renvoie les mêmes résultats :

```
def Bac2(m):  
    if m>=10:  
        return "admis"  
    else:  
        if m>=8:  
            return "oral"  
        else:  
            return "recalé"
```

5. Que signifie le mot réservé *else*?
6. Après le premier *else*, on se place dans le cas où m est
7. Après le second *else*, on se place dans le cas où m est

Exercice 2 :

Le but du TP est d'aider un vigneron du Jurançon à programmer une facturation. Connaissant le prix d'une bouteille, le nombre de bouteilles achetées, les éventuelles remises effectuées en fonction du nombre de bouteilles achetées, le programme doit renvoyer le montant total à payer pour le client.

Cette partie se fait en autonomie.

Données :

- Le vigneron met à la vente deux types de bouteilles : celles de la cuvée Automne à 9,20 € la bouteille, et celles de la cuvée Prestige à 13,85 € la bouteille
- A partir de 12 bouteilles achetées, et ce quel que soit la cuvée, le vigneron effectue une remise de 10% sur le montant total de la facture.

1. Un client achète 5 bouteilles de la cuvée Automne . Quel est le montant de la facture ?

.....
.....

2. Un client achète 7 bouteilles de la cuvée Prestige . Quel est le montant de la facture ?
.....
.....
3. Un client achète 3 bouteilles de la cuvée Automne et 6 bouteilles de la cuvée Prestige . Quel est le montant de la facture ?
.....
.....
4. Un client achète 6 bouteilles de la cuvée Automne et 10 bouteilles de la cuvée Prestige . Quel est le montant de la facture ?
5. Écrire une fonction Python qui prend en entrées le nombre de bouteilles achetées de la cuvée Automne et nombre de bouteilles achetées de la cuvée Prestige , et qui renvoie le montant de la facture. 11
.....
.....
.....
.....
.....
.....
.....
.....

On contrôlera que la fonction est correcte en vérifiant les résultats qu'elle propose sur les 4 premières questions.