

Initiation à Python :

Des fonctions en cascades pour faire de la géométrie...

Objectif : réaliser des fonctions successives sur Python et les réutiliser si besoin dans les autres programmes

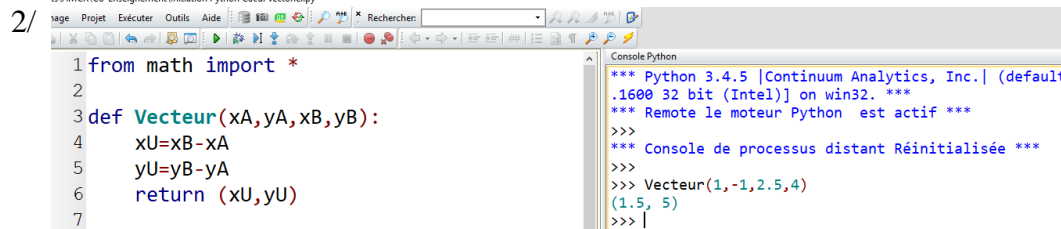
Ressource :

$\text{sqrt}(3)$	$\sqrt{3}$
$10^{**}3$	10^3

Partie A : S'appropriier l'environnement Python.

1/ Rappeler la formule du cours qui donne les coordonnées du vecteur \overline{AB} à partir de $A(x_A; y_A)$ et $B(x_B; y_B)$:

.....
.....



(a) Il y a une fenêtre pour écrire le programme et une autre pour le compiler (la console) : repérer ces deux fenêtres ci-dessus.

(b) A l'aide de la fiche ressource, identifiez :

- Le nom de la fonction programmée :
- Les arguments :
- Les variables :
- Ce que retourne la fonction :
- Que signifie « from math import * » ?
- Que signifie « indenter » ?

(c) Quel est le test réalisé dans la console ?
Vérifier, à la main, le résultat affiché :


.....
.....
.....
.....

Faire valider par le professeur.

Partie B : Premier programme

Réaliser la fonction « Vecteur » qui donne ses coordonnées à partir de deux points et la tester dans la console pour les points A(1 ;2) et B(3 ;4)

```
3 def Vecteur(xA,yA,xB,yB):
4     xU=xB-xA
5     yU=yB-yA
6     return (xU,yU)
```

Ne pas oublier de compiler avec le bouton  avant de faire les tests sinon les modifications apportées ne seront pas prises en compte.

```
*** Console de processus distant Réinitialisée ***
>>>
>>> Vecteur(1,2,3,4)
(2, 2)
>>>
```

Faire un nouveau test avec A(0 ;1) et B(0 ;5) :

Faire valider par le professeur.

Partie C : Prise d'initiative

1/ Rappeler la formule du cours qui donne la distance entre deux points sur dans un plan orthonormé :

.....
.....

2/ Construire une fonction « Distance », à la suite de celle que vous avez créé, qui permet de donner la distance entre deux points.

```
19 def Distance(xA,yA,xB,yB):
```

3/ Faire un test avec A(-4;10) et B(7;30) :

Faire valider par le professeur.

Partie D : Nature d'un triangle

Voici un algorithme qui réutilise des fonctions programmées auparavant. C'est d'ailleurs ce qui fait que Python est un logiciel de programmation puissant.

```
12 def TriangleIsocèle(xA, yA, xB, yB, xC, yC):
13     a=Distance(xB, yB, xC, yC)
14     b=Distance(xA, yA, xC, yC)
15     c=Distance(xA, yA, xB, yB)
16     if a==b or b==c or a==c:
17         return(" ")
18     else:
19         return(" ")
20
```

1/ Que représente chacune des variables a, b et c ?

a :
b :
c :

2/ Compléter les blancs de l'algorithme et donner l'objectif de cet algorithme ?
.....

3/ Faire un test avec A(1;0), B(2;1) et C(0;1):

Faire un test avec A(1;0), B(2;1) et C(0;2):

Faire valider par le professeur.

Partie E : Triangle rectangle

```
22 def TriangleRectangle(xA, yA, xB, yB, xC, yC):
```

Créer un programme qui permet de déterminer si un triangle est rectangle ou non.

Tester avec A(-1 ;1) B(1 ;2) C(3 ;-4) :

Faire un test avec A(1;0), B(2;1) et C(0;1):

Faire valider par le professeur

Partie F : Cartésien en polaire

Réaliser la fonction « CartésienPolaire » qui donne les coordonnées polaires à partir des coordonnées cartésiennes :

```
def CartésienPolaire(xA, yA):
```

Faire un test avec A(1 ;1) :

Faire un test avec B(-1 ;1) :

Faire valider par le professeur.