

Objectif principal : <ul style="list-style-type: none"> ➤ Comprendre l'utilité des fonctions en Python 	Contenus Mathématiques : <ul style="list-style-type: none"> ➤ Formule du milieu dans un repère ➤ Propriétés sur les parallélogrammes
Compétences Mathématiques : <ul style="list-style-type: none"> ➤ Calculer : mettre en œuvre des algorithmes ➤ Chercher : expérimenter à l'aide d'outils logiciel Compétences Algorithmiques : <ul style="list-style-type: none"> ➤ Découper un problème en sous-problèmes ➤ Réutiliser des algorithmes déjà programmés 	Matériel & Organisation : <ul style="list-style-type: none"> ➤ Logiciel de programmation Python (EduPython) ➤ En salle informatique ➤ En groupe de 2 ou 3 élèves

Voici ci-dessous un programme écrit en langage Python :

```
a=float(input("entrer l'abscisse du 1er point : "))
b=float(input("entrer l'ordonnée du 1er point : "))
c=float(input("entrer l'abscisse du 2eme point : "))
d=float(input("entrer l'ordonnée du 2eme point : "))
x=(a+c)/2
y=(b+d)/2
print("Le milieu de ces 2 points a pour coordonnées (", x , ";" , y , ")")
```

1. A quoi sert ce programme ?

.....

2. A l'aide de votre Mémo Python, expliquer pourquoi avoir mis la fonction float() pour enregistrer les variables a , b , c et d .

.....

.....

3. Tester le script **script_milieu.py** sous Python avec quelques valeurs.

Appeler le professeur pour vérification



Voici ce même programme écrit sous forme de **fonction** qui :

- prend en arguments les coordonnées (a ; b) et (c ; d) de deux points ;
- renvoie les coordonnées (x ; y) du milieu de ces points :

```
def milieu(a,b,c,d):
    x=(a+c)/2
    y=(b+d)/2
    return(x,y)
```

Taper cette fonction sous python puis enregistrer le fichier sous le nom **fonctions.py** (ce fichier vous servira tout le long du TP. Vous y enregistrerez toutes les fonctions que vous créerez).

Vous allez maintenant pouvoir exécuter votre fonction milieu.

Pour cela, vous devez taper dans la console : **milieu(a,b,c,d)** avec (a,b) les coordonnées de votre 1^{er} point et (c,d) les coordonnées de votre 2nd point.

Quel(s) avantage(s) à créer une fonction plutôt qu'un script ?

Des avantages, il y en a essentiellement deux : On évite les lignes de code d'entrée/sortie et on peut faire appel à une fonction déjà créée dans d'autres scripts et/ou fonctions !

A vous de créer une fonction !

Ecrire, dans votre fichier **fonctions.py**, une nouvelle fonction **parallelogramme(a,b,c,d,e,f,g,h)** qui :

- prend en arguments les coordonnées des quatre points $A(a ; b)$, $B(c ; d)$, $C(e ; f)$ et $D(g ; h)$ du quadrilatère ABCD ;
- affiche si le quadrilatère est un parallélogramme ou pas.

⚠ Vous pouvez faire appel à votre fonction milieu ainsi : **M=milieu(a,b,c,d)**. La variable M contiendra alors le couple de coordonnées du milieu calculé.

Ma fonction parallelogramme :

```
# Ceci est un commentaire :  
# si votre quadrilatère s'appelle ABCD, il faudra rentrer les coordonnées  
# des points A(a;b), B(c;d), C(e;f) puis D(g;h) en respectant bien cet ordre !
```

```
def parallelogramme(a,b,c,d,e,f,g,h) :
```

Appeler le professeur pour vérification



Pour aller plus loin...

1. Ecrire maintenant une fonction « **distance** » qui calculera la distance entre deux points du plan (*que vous écrirez dans le fichier **fonctions.py***)

2. Ecrire une fonction « **nature_parallelogramme** », utilisant les fonctions précédentes, qui prend en arguments les coordonnées de quatre points A, B, C, D et qui affiche la nature du quadrilatère ABCD.

Vous pourrez tester votre fonction avec les points suivants :

Parallélogramme	A(-2 ; 2)	B(3 ; 1)	C(-1 ; -3)	D(-6 ; -2)
Carré	A(4 ; 1)	B(2 ; 5)	C(-2 ; 3)	D(0 ; -1)
Losange	A(8 ; 5)	B(4 ; 2)	C(4 ; -3)	D(8 ; 0)
Quelconque	A(2 ; 5)	B(-1 ; 4)	C(-2 ; -3)	D(-5 ; -3)
Rectangle	A(1 ; -2)	B(2 ; 1)	C(-4 ; 3)	D(-5 ; 0)