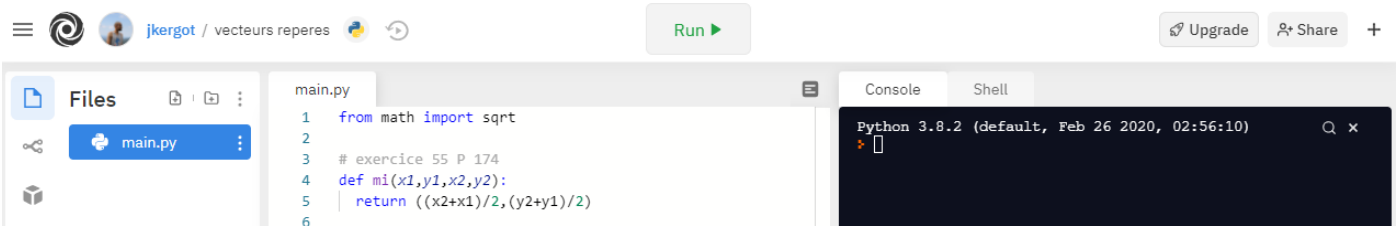
www.dimension-k.com

Programmation python Partie 1

Environnement de travail

On utilise le site repl.it qui nous permettra de créer autant de projets (repl) que l'on veut. Et qui en facilite le partage et l'exécution. Ça permet de commencer un travail sur un ordinateur et de le terminer sur un téléphone, une tablette ou un autre ordinateur sans avoir à utiliser de clé USB, d'installation ou autre. Le partage d'un projet se fait au travers de lien que l'on peut copier dans la barre de navigation.

L'interface :



Au niveau du ruban vous pouvez lire le nom de l'utilisateur , le nom du repl (modifiable en cliquant dessus)
Le bouton run permet de lancer le code qui est dans la partie centrale de l'écran.
A gauche de cette partie il y a un espace où l'ont peut stocker différents fichiers pour un projet.
A droite, on a la console qui est un espace où l'on peut utiliser des commandes python et voir le résultat de l'exécution du code au centre de la page (quand on appuie sur « Run »)

Premiers pas

Dans la console :

assignation

Python est muni des 4 opérations habituelles, pour les puissances on utilisera **, par exemple 2^4 s'écrira 2**4.
Pour stocker une information et l'utiliser plus tard, on utilise des variables dont on choisira le nom, qu'on remplira à notre guise.

```
> nombre=5*4
> texte="bonjour !"
```

va stocker le résultat de 5*4 dans la variable nommée « nombre »
va stocker la chaine de caractères ''bonjour !'' dans la variable nommée « texte »

Fonctions

Python contient tout un lot de fonctions qui vont avoir des effets paramétrables.
Chaque fonction est suivie de parenthèses qui contiendront (ou non) les paramètres de la fonction.
print() est l'une d'entre elle.

```
> print("petite phrase")
petite phrase
> print(texte," les gens")
bonjour !  les gens
```

Dans la partie programme

Création de fonction

Ça commence toujours par def suivi du nom de la fonction suivie de parenthèses contenant les noms des paramètres, et cette première ligne se termine toujours par « : »
Le reste de la définition de la fonction sera décalée d'une tabulation (deux espaces) vers la droite.
Exemple 1 (voir image en haut du document)
La fonction appelée mi a quatre paramètres qui sont les coordonnées de deux points.
« return » indique que la fonction donne (renvoie) au programme une petite liste contenant les résultats paramètre
Exemple 2

```
7  def par(xA,yA,xB,yB,xC,yC,xD,yD):
8    if mi(xA,yA,xC,yC)==mi(xB,yB,xD,yD) :
9      para=True
10   else :
11     para=False
12   return para
```

la fonction s'appelle par elle a 8 paramètres qui sont les coordonnées de 4 points.
Elle utilise la fonction de l'exemple 1 :
mi(xA,yA,xC,yC) correspond aux coordonnées du milieu de [AC] et mi(xB,yB,xD,yD) à celle du milieu de [BD].
S'il y a égalité entre les deux on va effectuer la ligne 9 sinon la ligne 11. Une fois que ça sera fait on passe à la ligne 12 qui renverra au programme le contenu de la variable para.
Utilisation de fonction

Pour pouvoir utiliser une fonction que l'on créé il faut exécuter le code (on utilise le bouton run à cet effet).
Une fonction est comme un théorème, c'est un outil potentiel, et elle a besoin d'être appelé sur une situation concrète pour donner un résultat utilisable.
Si je veux connaitre les coordonnées du milieu du segment [AB] avec A(5 ;11) et B(-1 ;7) on tapera :

```
> print(mi(5,11,-1,7))
(2.0, 9.0)
```

On sait donc que le segment [AB] aura pour coordonnées (2 ;9)

Assignations multiples :

On peut remplir plusieurs variables avec une seule ligne de commande :

```
x1,y1=xB-xA,yB-yA
x2,y2=xC-xA,yC-yA
```

A gauche du signe d'égalité on a une série de variables et à gauche une série de calculs , la premières variable sera remplie avec le premier calcul, la seconde variable avec le second calcul etc…

Indentation

Quand on veut indiquer que certaines instructions ne soient exécutées sous certaines conditions : on va les indenter, ce qui veut dire qu'elles seront écrites sous la condition et décalées vers la droite par rapport à celle-ci. Ce décalage est mis en évidence par Repl avec un trait vertical.
Par exemple :

```
def milieu(xA,yA,xB,yB):
  xI=(xA+xB)/2
  yI=(yA+yB)/2
  return (xI,yI)
```

- Les commandes d'une fonction ne seront exécutées que si celle-ci est utilisées, on les décale donc par rapport à celle-ci.
- Pour les informations exécutées si une condition est réalisée (avec if) ou quand celle-ci ne l'est pas (avec else) on les décalera.

Dans tous les cas, le décalage est précédé par « : »

```
if x>=0 :
  print("x est positif")
  print("sa racine est ",sqrt(x))
else:
  print("x est négatif, il n'a pas de racine")
```

Bonus : Equations de droite

Exemple / méthode

Soit I(5 ;-7) et J(-3 ;2).

1) Déterminer une équation cartésienne de la droite (IJ)
2) Dire si les points V(13 ;-16) et W(2 ;3) sont sur la droite.

1)

$M \in (IJ) \Leftrightarrow \overrightarrow{IJ}\begin{pmatrix}-3-5\\2-(-7)\end{pmatrix}$ et $\overrightarrow{IM}\begin{pmatrix}x-5\\y-(-7)\end{pmatrix}$ sont colinéaires $\Leftrightarrow \det\left(\overrightarrow{IJ}\begin{pmatrix}-8\\2+7\end{pmatrix};\overrightarrow{IM}\begin{pmatrix}x-5\\y+7\end{pmatrix}\right)=0$

$\Leftrightarrow \begin{vmatrix}-8 & x-5\\ 9 & x+7\end{vmatrix}=0 \Leftrightarrow -8(y+7)-9(x-5)=0 \Leftrightarrow -8y-56-9x+45=0$

$\Leftrightarrow -9x-8y-11=0$ est une équation cartésienne de la droite (IJ)

2)

Pour le point V(13 ;-16)

$-9x-8y-11=-9(13)-8(-16)-11=-117+128-11=0$ le point V est bien sur la droite (IJ)

Pour le point W(2 ;3)

$-9x-8y-11=-9(2)-8(3)-11=-18-24-11=-53\neq 0$ le point W n'est pas sur la droite (IJ)