Info 9 - Boucles non bornées- *while*

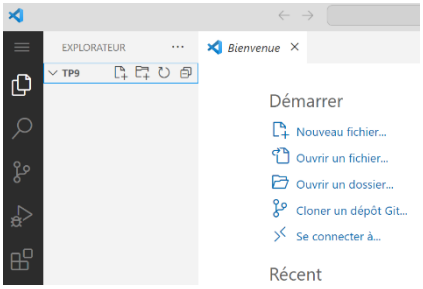
Dans ce TP on voit 2 codes utilisant une syntaxe *while*. Ces codes seront écrits en utilisant *Visual Studio Code* comme environnement de développement.

On demande de rédiger un compte-rendu au format *.doc* ou *.odt* , à transférer en fin d'activité par l'intermédiaire de l'onglet ***Mon Compte*** du site https://nsibranly.fr en utilisant le code : **tp9** . Ce compte-rendu contiendra :

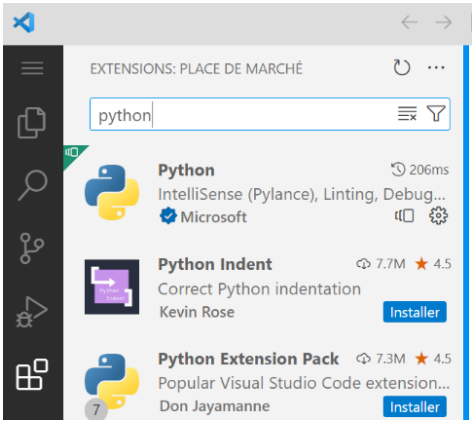
- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes** écrits **et** celles **des résultats des exécutions**. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows (touches clavier *windows+Shift+s*)

Attention à repérer correctement les titres de paragraphe.

1- Utilisation de Visual Studio Code :

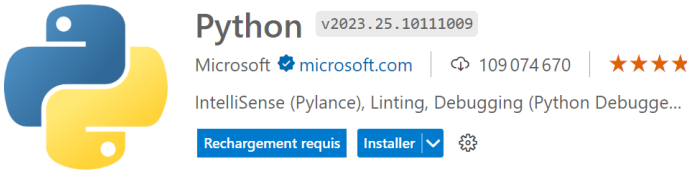


⇨ Dans Visual Studio Code, ouvrir un nouveau dossier, nommé par exemple *tp9*, dans votre espace de travail, sur le serveur ***U:*** du lycée.



⇨ Dans la barre de menu verticale située à gauche, cliquer sur l'onglet « *extensions* »

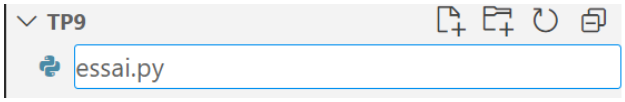
⇨ Dans la ligne de recherche, écrire « *python* » …. une liste d'extensions disponibles apparait …. choisir la 1ère « *Python de Microsoft* » …



⇨ Cliquer sur « *Installer* ».

⇨ Vous pouvez également écrire dans la ligne de recherche « *french* » … et installer l'extension de vsc en français.

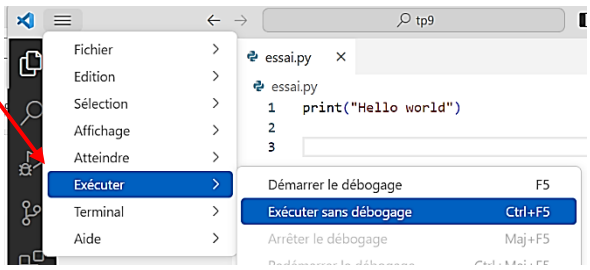
⇨ Créer un nouveau fichier nommé « *essai.py* » par exemple.



⇨ Ecrire dans ce fichier, par exemple, l'instruction python « `print("Hello world")` » .

⇨ Exécuter ce fichier soit en cliquant sur l'icone  , soit en allant dans le menu principal et en utilisant l'onglet « *Exécuter* »

Soit en appuyant sur la touche *F5* du clavier.



2- PLACEMENT FINANCIER A 10 % :

Mr Musk Junior achète 1 000 € d'actions d'une start-up qui travaille dans l'I.A. La rénumération annuelle promise est de 10% . La valeur de son portefeuille d'actions est de 1 000 € en janvier 2024. En janvier 2025, sa valeur sera normalement de 1 100 €, puis de 1 210 € en janvier 2026, 1 331 € en janvier 2027, etc ….

⇨ Ouvrir un nouveau fichier que vous nommerez *placement.py* . Y copier-coller le code python ci-dessous et enregistrer le fichier.

```
def portefeuille(somme) :
    s = 1000
    annee = 2024
    while s < somme :
        interet = s * 10/100
        s = s + interet
        annee = annee + 1
        print(f"En janvier {annee}, la valeur du portefeuille est : {s:.1f} €")
    return annee

# Programme principal
a = portefeuille(10000)
print(f"Mr Musk Junior aura plus de {10000} € en janvier {a}")
```

⇨ Exécuter ce fichier …. on obtient normalement dans le Terminal :

```
En janvier 2026, la valeur du portefeuille est : 1210.0 €
En janvier 2027, la valeur du portefeuille est : 1331.0 €
En janvier 2028, la valeur du portefeuille est : 1464.1 €
En janvier 2029, la valeur du portefeuille est : 1610.5 €
En janvier 2030, la valeur du portefeuille est : 1771.6 €
En janvier 2031, la valeur du portefeuille est : 1948.7 €
En janvier 2032, la valeur du portefeuille est : 2143.6 €
En janvier 2033, la valeur du portefeuille est : 2357.9 €
En janvier 2034, la valeur du portefeuille est : 2593.7 €
En janvier 2035, la valeur du portefeuille est : 2853.1 €
En janvier 2036, la valeur du portefeuille est : 3138.4 €
En janvier 2037, la valeur du portefeuille est : 3452.3 €
En janvier 2038, la valeur du portefeuille est : 3797.5 €
En janvier 2039, la valeur du portefeuille est : 4177.2 €
En janvier 2040, la valeur du portefeuille est : 4595.0 €
En janvier 2041, la valeur du portefeuille est : 5054.5 €
En janvier 2042, la valeur du portefeuille est : 5559.9 €
En janvier 2043, la valeur du portefeuille est : 6115.9 €
En janvier 2044, la valeur du portefeuille est : 6727.5 €
En janvier 2045, la valeur du portefeuille est : 7400.2 €
En janvier 2046, la valeur du portefeuille est : 8140.3 €
En janvier 2047, la valeur du portefeuille est : 8954.3 €
En janvier 2048, la valeur du portefeuille est : 9849.7 €
En janvier 2049, la valeur du portefeuille est : 10834.7 €
Mr Musk junior aura plus de 10000 € en janvier 2049
```

Ainsi ce code permet ici de donner l'année à laquelle, la valeur du portefeuille dépassera les 10 000 €.

Dans Visual Studio Code, on peut créer des points d'arrêtes pour pouvoir ensuite réaliser une exécution avec « *Débogage* » :

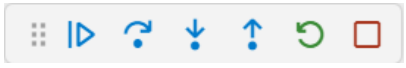
⇨ Avec la souris, ajouter un point d'arrêt sur la ligne 5, en cliquant sur la gauche du chiffre 5.

```
placement.py > portefeuille
1   def portefeuille(somme) :
2       s = 1000
3       annee = 2024
4       while s < somme :
5           interet = s * 10/100
6           s = s + interet
7           annee = annee + 1
8           print(f"En janvier {annee}, la valeur du portefeuille est : {s:.1f} €")
9       return annee
10
11  # Programme principal
12  a = portefeuille(10000)
13  print(f"Mr Musk Junior aura plus de {10000} € en janvier {a}")
```

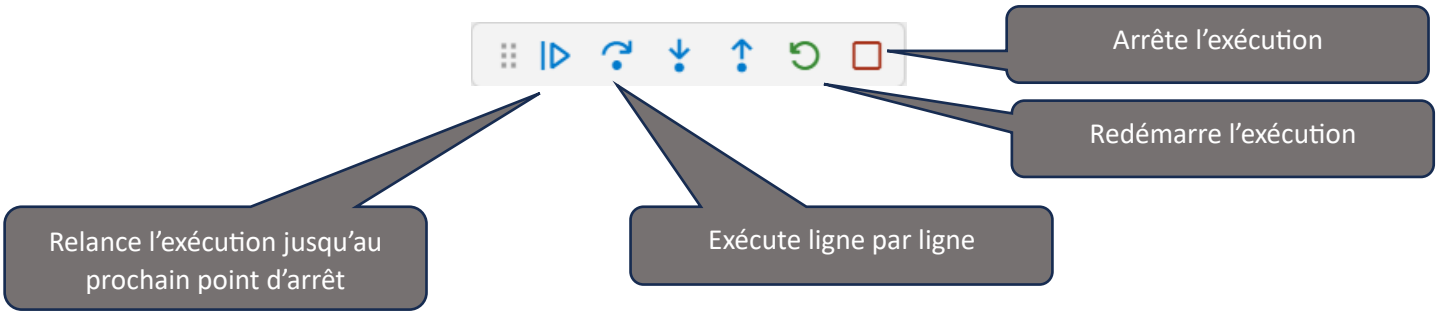
⇨ Lancer une exécution avec débogage en appuyant sur la touche F5 du clavier …

…. on voit que les valeurs des variables utilisées dans le code, sont affichées dans la menu latéral de gauche :

….. la barre d'outils ci-contre

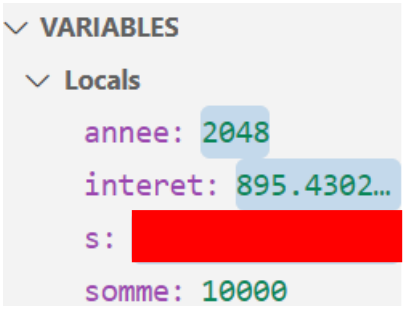


s'ouvre et propose des outils qui permettent de réaliser une exécution étapes par étapes :

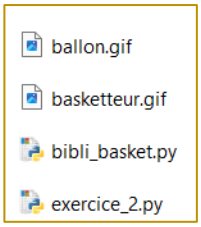


Les valeurs des variables sont constamment réactualisées tout au long de cette exécution « *avec débogage* ».

⇨ La copie d'écran ci-contre est ici incomplète, une valeur a été cachée. Afin de montrer que vous avez bien réalisé cette exécution « «*avec débogage* », on vous demande de copier-coller dans votre compte-rendu, cette même copie d'écran, par-contre avec la valeur non cachée.



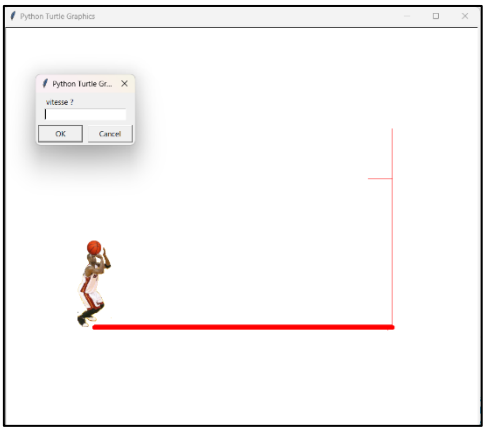
3- SHOOT AU BASKET :



⇨ Télécharger sur *nsibranly.fr*, le dossier zip proposé avec ce tp. Il contient les fichiers ci-contre. Extraire ces 4 fichiers dans votre répertoire de travail *tp9*.

⇨ Ouvrir avec *Visual Studio Code*, le fichier *exercice_2.py* et l'exécuter ….

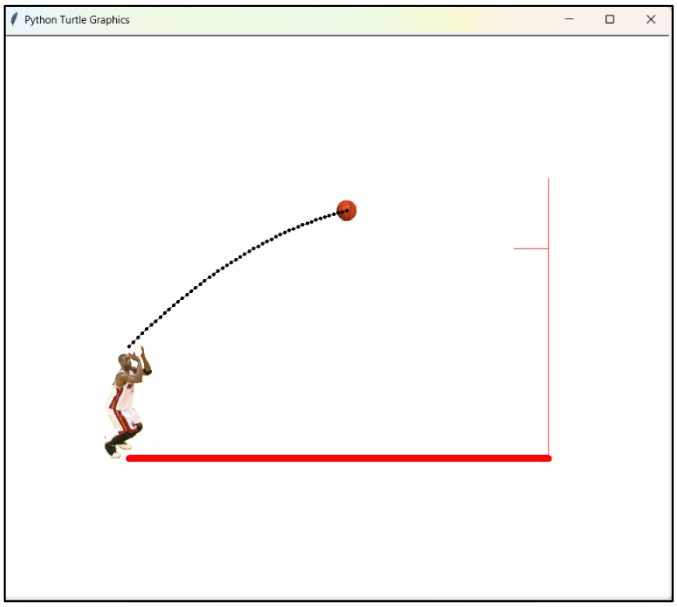
….. la fenêtre *turtle* ci-dessous est créée :



⇨ Saisir par exemple, une vitesse de 90 cm/s

puis un angle de 45° …

….. la position du ballon est alors calculée et tracée toutes les 0,1 seconde, ceci 50 fois seulement.



⇨ Refaire une exécution avec une vitesse de 120 cm/s puis un angle de 75° …

⇨ Refaire une exécution avec une vitesse de 50 cm/s puis un angle de 20° …

Pour l'instant, ce fichier contient le code suivant :

```
1   # importation des bibliothèques
2   from turtle import *
3   from math import cos,sin,pi
4   from bibli_basket import decor
5
6   # construction du décor par l'appel de la fonction decor()
7   ballon = decor()
8   xb = -300
9   yb = -40
10  # saisi vitesse et angle
11  v = int(numinput("", "vitesse ?",minval = 0, maxval = 500))
12  angle = int(numinput("", "angle ?",minval = 0, maxval = 90))
13  angle = angle * pi / 180
14
15  # lancer
16  x = xb
17  y = yb
18  t = 0
19  for i in range(50) :
20      x = v*cos(angle)*t + xb
21      y = -9.81/2*t**2 + v*sin(angle)*t + yb
22      ballon.goto(x,y)
23      ballon.dot(5)
24      t = t + 0.1
25
26  # à laisser
27  exitonclick()
```

La fonction decor() est définie dans le fichier bibli_basket.py . Elle permet d'initialiser la partie graphique : NE PAS MODIFIER CE FICHIER

La fonction decor() est exécutée. Elle renvoie un objet turtle nommé ballon

L'utilisateur donne une valeur aux variables v et angle.

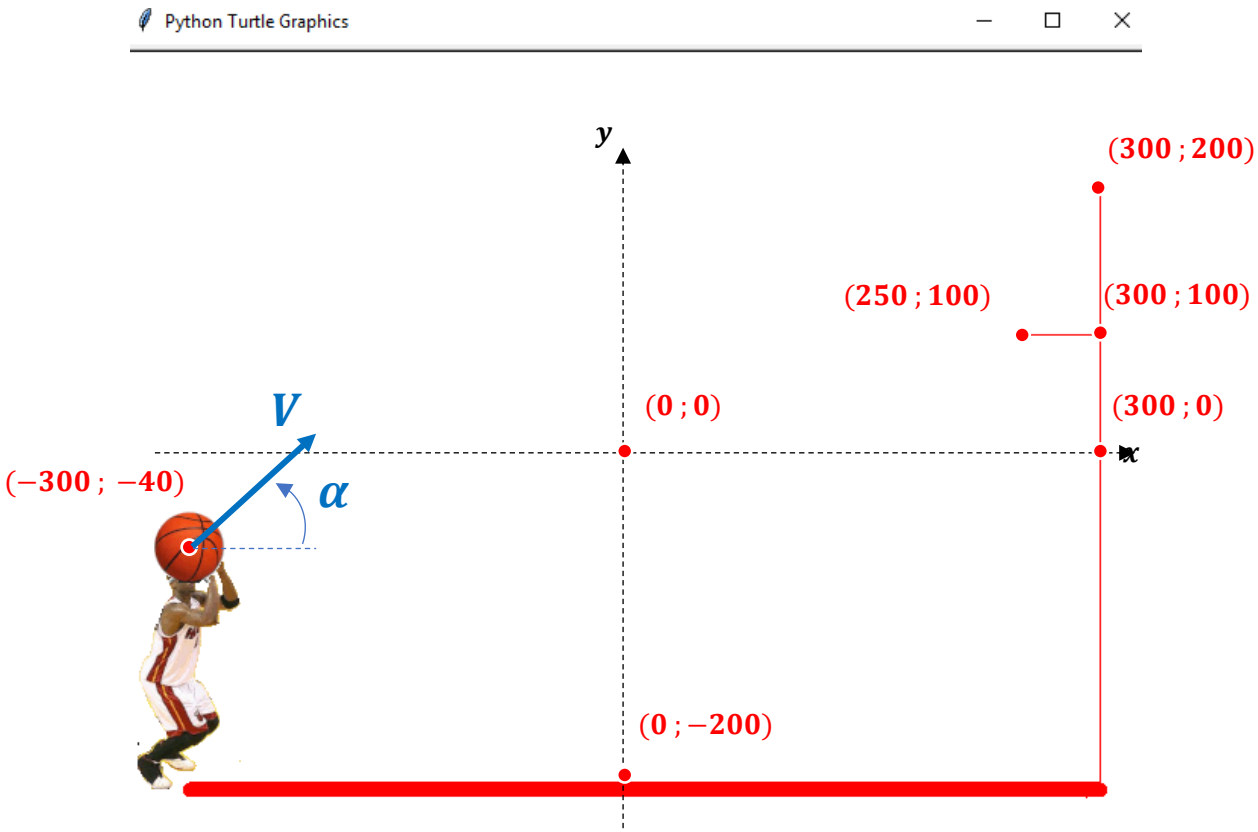
La valeur de v est convertie en radians

Une boucle bornée de 50 itérations est réalisée. A chaque fois les coordonnées x et y du ballon sont recalculées en utilisant des relations de physique qui font intervenir le temps t, les coordonnées, vitesse et angle initiaux et la gravité.

Trace un rond de rayon 5px

Déplace le ballon sur les coordonnées x et y

Le repère mathématique de la fenêtre turtle est centré. L'unité est le pixel, mais on considère que 1px sur l'écran correspond à 1 cm en réalité. Les coordonnées des points importants sont données sur la figure ci-dessous :



L'objectif de cet exercice est de modifier ce fichier *exercice_2.py* afin d'utiliser une boucle *while* qui permette de poursuivre le tracé de la trajectoire jusqu'à ce que le ballon :

- soit arrive sur la droite, à la verticale du panneau,
- soit touche le sol en retombant.

On demande également de repérer si le shoot est réussi ou pas. Il est considéré réussi si le centre du ballon passe entre les extrémités gauche et droite de l'anneau. En fin de trajectoire, on aura ainsi l'exécution de la ligne `write("GAGNE",font = ("Arial", 30, "bold"))` ou `write("PERDU",font = ("Arial", 30, "bold"))`

4- AMELIORATION DU CODE PRECEDENT :

Reprendre le code de l'exercice précédent pour y apporter les modifications suivantes :

⇨ Placer une partie du code dans une autre boucle *while* afin de pouvoir shooter plusieurs fois de suite. On pourra utiliser les instructions suivantes :

```
rep = 1
while rep == 1 :
```

```
ballon.clear()
ballon.up()
ballon.goto(xb,yb)
ballon.down()
```

```
rep = int(numinput("", "On rejoue ? 1:oui, 0:non",minval = 0, maxval = 1))
```

```
write("GAGNE",font = ("Arial", 30, "bold"))
```

- Inclure dans le code un script qui comptabilise le nombre de paniers tirés.