Etape 6 : Calcul des coordonnées d'un vecteur vitesse

Notions abordées

- Boucle for
- Ajout d'une valeur dans une liste
- Parcours des valeurs d'une liste
- Longueur d'une liste
- Création et intérêt d'une fonction

Références pyspc

- Les boucles
- Les listes
- Les fonctions

Consigne :

Etudier les programmes ci-dessous puis effectuer la mise en situation présentée dans la dernière cellule.

Etude préliminaire

Exécuter la cellule ci-dessous puis analyser ce qu'elle renvoie afin de comprendre les différentes instructions posées.

In [1]:

```
Liste=["a",11,12,13,14,15,"b",17,18,"c"]

print(Liste)
print('')
print("nombre d'éléments dans la liste=",len(Liste))
print('\n')
print(Liste[0])
print(Liste[3],'\n')
for i in range(10):
    print(i,'',Liste[i])
print('\n')
for i in range (2,9):
    print(i,'',Liste[i])
Liste.append("d")
print("")
print(Liste)
```

```
['a', 11, 12, 13, 14, 15, 'b', 17, 18, 'c']

nombre d'éléments dans la liste= 10


a
13

0  a
1  11
2  12
```

```
3  13
4  14
5  15
6  b
7  17
8  18
9  c


2  12
3  13
4  14
5  15
6  b
7  17
8  18

['a', 11, 12, 13, 14, 15, 'b', 17, 18, 'c', 'd']
```

Exécuter maintenant les deux cellules ci-dessous afin de comprendre comment créer puis appeler une fonction.

In [2]:

```
# Création de la fonction f dont les paramètres d'entrée ordonnés
# sont x et y et qui retourne le paramètre de sortie z

def f(x,y):
    z=2*x+y
    return z
```

In [3]:

```
# Différents appels possibles de la fonction

u=f(2,3)
print(u)

toto=3
titi=4
v=f(toto,titi)
print (v)

# ci-dessous, la valeur retournée par la fonction est affichée
# mais n'est pas affectée à une variable et ne peut
# donc pas être réutilisée plus loin dans le programme

print (f(titi,toto))
```

```
7
10
11
```

Détermination des coordonnées d'un vecteur vitesse

In [4]:

```
# Création des listes contenant les valeurs du temps
# et des coordonnées du vecteur position

t=[0.0, 0.04, 0.08, 0.12, 0.16, 0.2, 0.24, 0.28, 0.32,
   0.36, 0.4, 0.44, 0.48, 0.52, 0.56, 0.6, 0.64, 0.68, 0.72]

x=[-0.003, 0.065, 0.140, 0.214, 0.287, 0.362, 0.435,
   0.514, 0.584, 0.663, 0.739, 0.815, 0.890, 0.9662,
   1.039, 1.115, 1.191, 1.270, 1.340]

y=[0.0, 0.143, 0.267, 0.376, 0.472, 0.553, 0.618,
   0.666, 0.694, 0.713, 0.713, 0.696, 0.660, 0.618,
   0.553, 0.469, 0.374, 0.261, 0.135]
```

In [5]:

```
# Création d'une liste contenant les valeurs de
# la coordonnée vx du vecteur vitesse

vx=[]
for i in range (len(x)-1):
    vxi=(x[i+1]-x[i])/(t[i+1]-t[i])
    vx.append(vxi)
print(vx)
```

```
[1.7000000000000002, 1.8750000000000002, 1.8499999999999999, 1.8249999999999993, 1.875,
1.825000000000001, 1.9749999999999985, 1.7499999999999996, 1.9750000000000028,
1.8999999999999972, 1.9, 1.8750000000000027, 1.9049999999999967, 1.8199999999999978,
1.9000000000000052, 1.9, 1.9749999999999972, 1.7500000000000049]
```

In [6]:

```
# Création d'une liste contenant les valeurs
# de la coordonnée vy du vecteur vitesse

vy=[]
for i in range (len(y)-1):
    vyi=(y[i+1]-y[i])/(t[i+1]-t[i])
    vy.append(vyi)
print(vy)
```

```
[3.5749999999999997, 3.1000000000000005, 2.725, 2.399999999999999, 2.0250000000000012,
1.6249999999999996, 1.2, 0.6999999999999982, 0.47500000000000064, 0.0,
-0.4250000000000006, -0.8999999999999985, -1.05, -1.6249999999999973, -2.100000000000006,
-2.3749999999999973, -2.824999999999997, -3.150000000000006]
```

Mise en situation

Faire une copie de ce notebook puis modifier le programme en créant une fonction permettant d'éviter la répétition des lignes de code lors du calcul des coordonnées du vecteur vitesse