

UE523 – Traitement de données sur μ Contrôleur

Rapport TP 2- Programmation Python pour Sense HAT

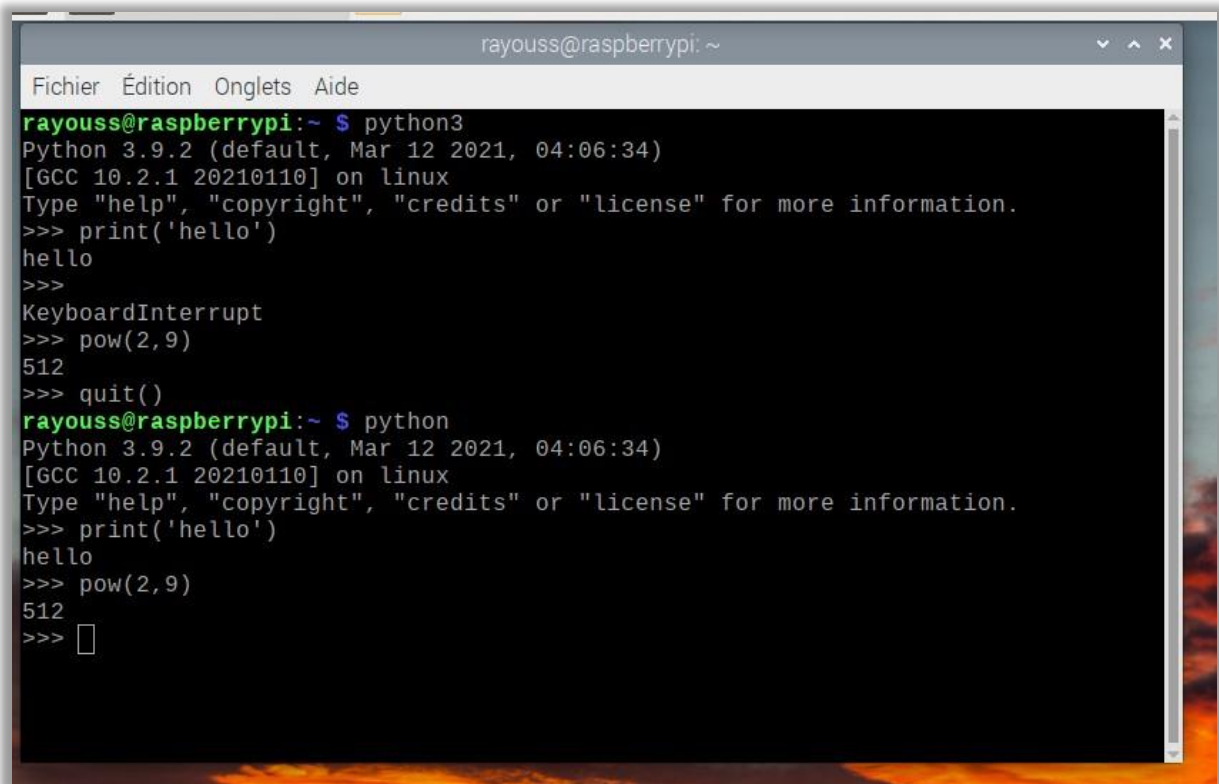
Fait par **AMIRAT** Samy & **SARDOU** Rayan

Etudiants en L3-EEA

Sous la direction de Mr **Talotta F** .

1. Exercice 1

1.1. Question 1



```
rayouss@raspberrypi: ~  
Fichier Édition Onglets Aide  
rayouss@raspberrypi:~ $ python3  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print('hello')  
hello  
>>>  
KeyboardInterrupt  
>>> pow(2,9)  
512  
>>> quit()  
rayouss@raspberrypi:~ $ python  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print('hello')  
hello  
>>> pow(2,9)  
512  
>>> 
```

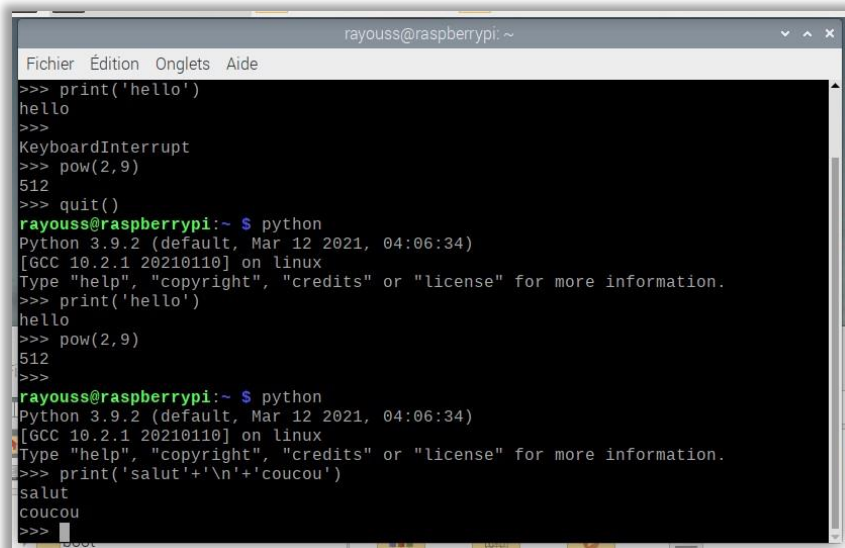
La version exacte de python est ici la version 3.9.2.

On remarque qu'il se passe **exactement la même chose** que l'on écrive python ou python3

1.2. Question 2

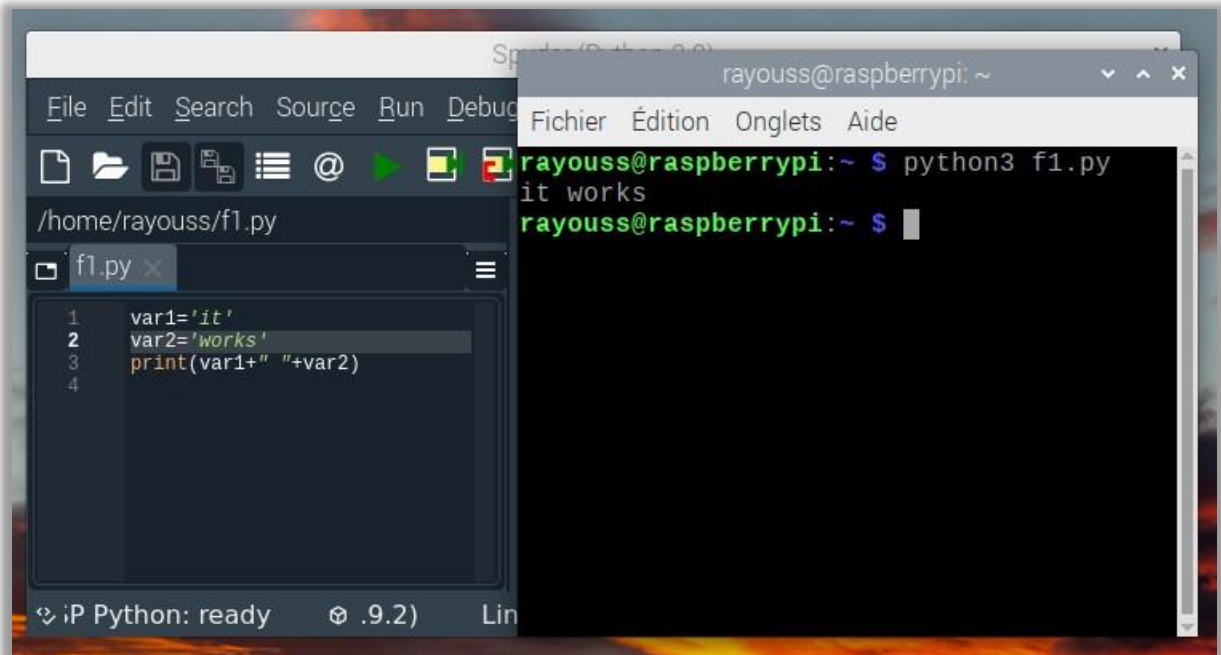
On utilise la commande **pow(2,9)**.

1.3. Question 3 : Le caractère spécial \n sert à faire un retour chariot.



```
rayouss@raspberrypi: ~  
Fichier Édition Onglets Aide  
>>> print('hello')  
hello  
>>>  
KeyboardInterrupt  
>>> pow(2,9)  
512  
>>> quit()  
rayouss@raspberrypi:~ $ python  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print('hello')  
hello  
>>> pow(2,9)  
512  
>>>  
rayouss@raspberrypi:~ $ python  
Python 3.9.2 (default, Mar 12 2021, 04:06:34)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print('salut'+'\n'+ 'coucou')  
salut  
coucou  
>>> 
```

1.4.Question 4 : Le nom du fichier python est ici f1.py et on utilise la commande python3 f1.py pour l'exécuter.



The screenshot shows a Raspberry Pi desktop environment. A code editor window titled 'f1.py' is open, displaying the following Python code:

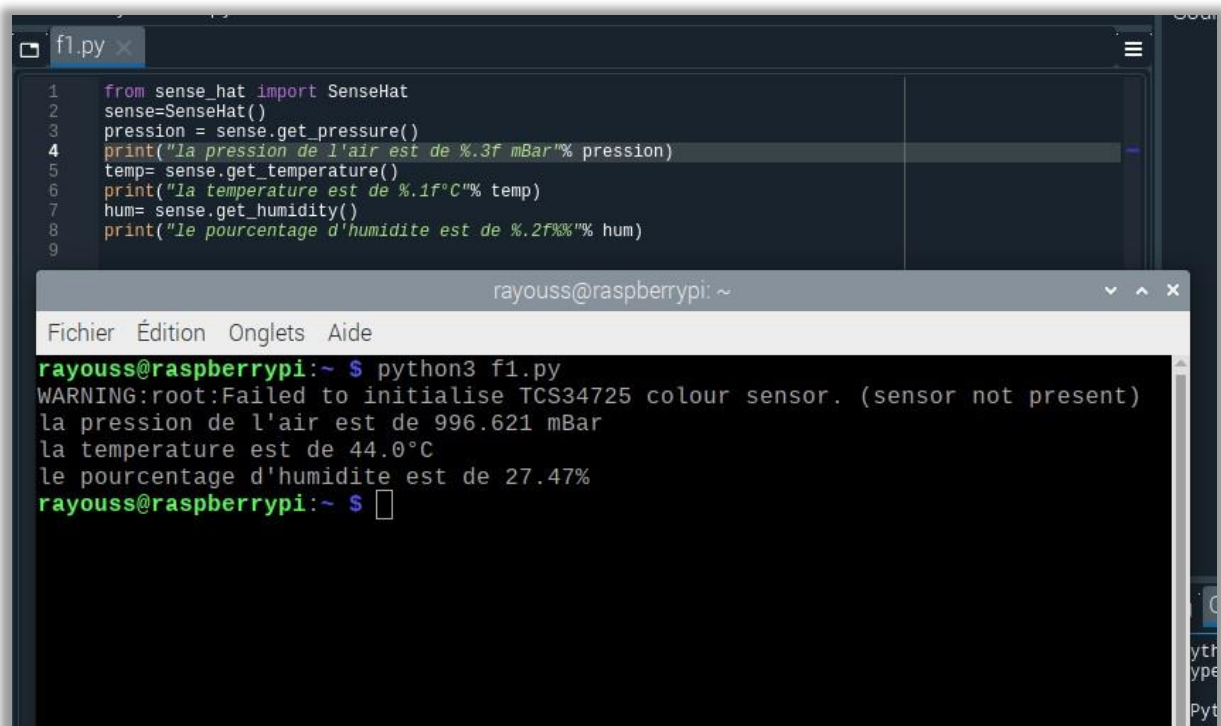
```
1 var1='it'
2 var2='works'
3 print(var1+" "+var2)
4
```

Below the code editor, a terminal window titled 'rayouss@raspberrypi: ~' shows the command 'python3 f1.py' being executed. The output of the script is 'it works'.

```
rayouss@raspberrypi:~ $ python3 f1.py
it works
rayouss@raspberrypi:~ $
```

2. Exercice 2

2.1.Question 1



The screenshot shows a Raspberry Pi desktop environment. A code editor window titled 'f1.py' is open, displaying the following Python code:

```
1 from sense_hat import SenseHat
2 sense=SenseHat()
3 pression = sense.get_pressure()
4 print("la pression de l'air est de %.3f mBar"% pression)
5 temp= sense.get_temperature()
6 print("la temperature est de %.1f°C"% temp)
7 hum= sense.get_humidity()
8 print("le pourcentage d'humidite est de %.2f%%"% hum)
9
```

Below the code editor, a terminal window titled 'rayouss@raspberrypi: ~' shows the command 'python3 f1.py' being executed. The output of the script is:

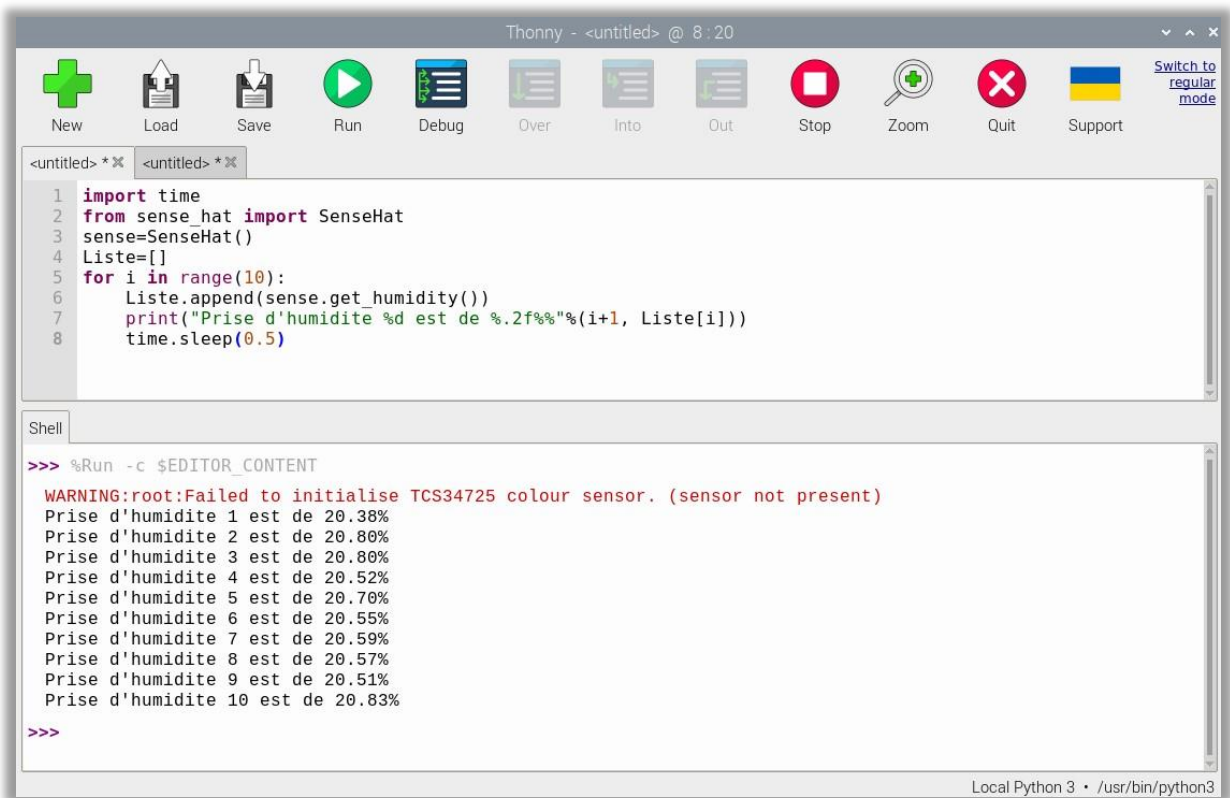
```
rayouss@raspberrypi:~ $ python3 f1.py
WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)
la pression de l'air est de 996.621 mBar
la temperature est de 44.0°C
le pourcentage d'humidite est de 27.47%
rayouss@raspberrypi:~ $
```

2.2.Question 2

L'erreur de mesure de température est due à la **température élevée** du processeur à proximité du capteur.

La valeur récupérer peut-être plus précise si l'on **améliore le refroidissement** du processeur à proximité. Ou si **l'on éloigne physiquement le capteur de température** du processeur.

2.3.Question 3



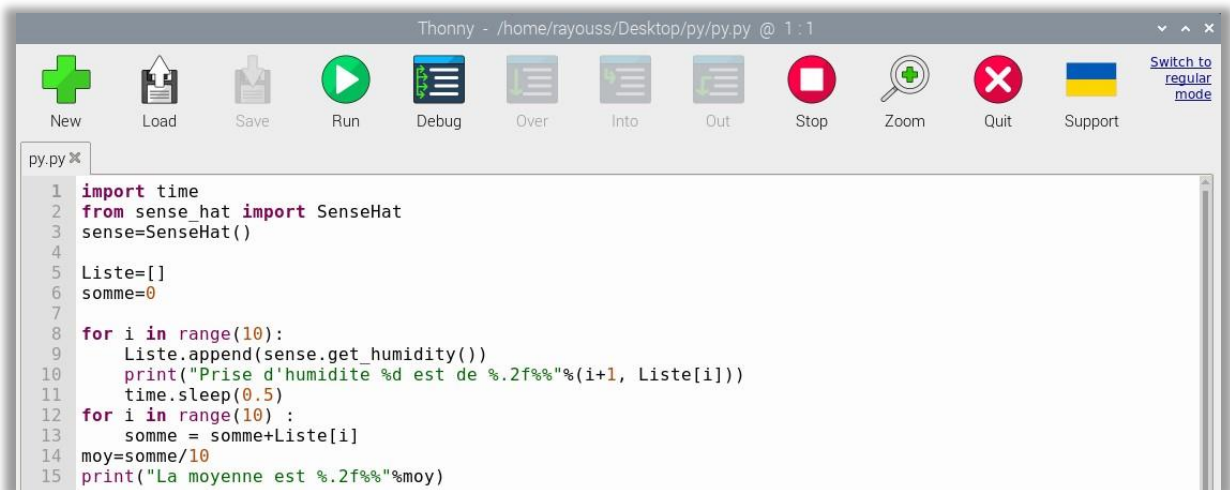
The screenshot shows the Thonny IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The main editor window displays a Python script that imports the SenseHat module and measures humidity 10 times, printing the results. The Shell window at the bottom shows the execution output, including a warning about the TCS34725 colour sensor and the humidity readings.

```
<untitled> *  
1 import time  
2 from sense_hat import SenseHat  
3 sense=SenseHat()  
4 Liste=[]  
5 for i in range(10):  
6     Liste.append(sense.get_humidity())  
7     print("Prise d'humidite %d est de %.2f%%"%(i+1, Liste[i]))  
8     time.sleep(0.5)
```

```
>>> %Run -c $EDITOR_CONTENT  
WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)  
Prise d'humidite 1 est de 20.38%  
Prise d'humidite 2 est de 20.80%  
Prise d'humidite 3 est de 20.80%  
Prise d'humidite 4 est de 20.52%  
Prise d'humidite 5 est de 20.70%  
Prise d'humidite 6 est de 20.55%  
Prise d'humidite 7 est de 20.59%  
Prise d'humidite 8 est de 20.57%  
Prise d'humidite 9 est de 20.51%  
Prise d'humidite 10 est de 20.83%  
>>>
```

Local Python 3 • /usr/bin/python3

2.4.Question 4



The screenshot shows the Thonny IDE interface. The top toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The main editor window displays a Python script that imports the SenseHat module, measures humidity 10 times, and calculates the average humidity. The Shell window at the bottom shows the execution output, including the warning about the TCS34725 colour sensor and the humidity readings.

```
py.py *  
1 import time  
2 from sense_hat import SenseHat  
3 sense=SenseHat()  
4  
5 Liste=[]  
6 somme=0  
7  
8 for i in range(10):  
9     Liste.append(sense.get_humidity())  
10    print("Prise d'humidite %d est de %.2f%%"%(i+1, Liste[i]))  
11    time.sleep(0.5)  
12 for i in range(10):  
13     somme = somme+Liste[i]  
14 moy=somme/10  
15 print("La moyenne est %.2f%%"%moy)
```

```
Shell
>>> %Run py.py
WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)
Prise d'humidite 1 est de 20.15%
Prise d'humidite 2 est de 20.35%
Prise d'humidite 3 est de 20.07%
Prise d'humidite 4 est de 19.92%
Prise d'humidite 5 est de 20.39%
Prise d'humidite 6 est de 20.27%
Prise d'humidite 7 est de 19.98%
Prise d'humidite 8 est de 19.81%
Prise d'humidite 9 est de 20.21%
Prise d'humidite 10 est de 20.10%
la moyenne est 20.12%
>>>
```

Local Python 3 • /usr/bin/python3

2.5.Question 5

Thonny - /home/rayouss/Desktop/py/py.py @ 22:35

New Load Save Run Debug Over Into Out Stop Zoom Quit Support [Switch to regular mode](#)

```
py.py
2 from sense_hat import SenseHat
3 sense=SenseHat()
4
5 Liste=[]
6 somme=0
7 sommeEcartype=0
8
9 for i in range(10):
10     Liste.append(sense.get_humidity())
11     print("Prise d'humidite %d est de %.2f%%"%(i+1, Liste[i]))
12     time.sleep(0.5)
13
14 for i in range(10) :
15     somme = somme+Liste[i]
16 moy=somme/10
17 print("La moyenne est %.2f%%"%moy)
18
19 for i in range(10) :
20     sommeEcartype=pow((Liste[i]-moy),2)+sommeEcartype
21
22 Ecartype=pow(0.1*sommeEcartype,0.5)
23 print("L'ecart type est %.2f%%"%Ecartype)
24
```

```
Shell
>>> %Run py.py
WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)
Prise d'humidite 1 est de 20.24%
Prise d'humidite 2 est de 20.51%
Prise d'humidite 3 est de 20.58%
Prise d'humidite 4 est de 20.72%
Prise d'humidite 5 est de 20.71%
Prise d'humidite 6 est de 20.72%
Prise d'humidite 7 est de 20.58%
Prise d'humidite 8 est de 20.90%
Prise d'humidite 9 est de 20.85%
Prise d'humidite 10 est de 20.86%
La moyenne est 20.67%
L'ecart type est 0.19%
>>>
```

Local Python 3 • /usr/bin/python3

2.6.Question 6

```
test csv.py* x
1 import time
2 from datetime import datetime
3 import csv
4 from sense_hat import SenseHat
5 sense=SenseHat()
6 now = datetime.now()
7 k=0
8 moyenneHumidite=[]
9 moyenneTemperature=[]
10 ecartypehum=[]
11 date=[]
12 heure=[]
13 while k<10 :
14     k+=1
15     Hum=[]
16     Temp=[]
17     sommeH=0
18     sommeT=0
19     sommeEcartypeH=0
20
21     for i in range(10):
22         Hum.append(sense.get_humidity())
23         Temp.append(sense.get_temperature())
24         print("Prise %d"%(i+1))
25         print("L'humidité est de %.2f%%"% Hum[i])
26         print("La température est de %.1f°C"%Temp[i])
27         print("")
28         time.sleep(5)
29
30     for i in range(10) :
31         sommeH = sommeH+Hum[i]
32         sommeT = sommeT+Temp[i]
33
34     moyH=sommeH/10
35     moyT=sommeT/10
36     moyenneHumidite.append(moyH)
37     moyenneTemperature.append(moyT)
38     date.append(now.strftime("%Y-%m-%d"))
39     heure.append(now.strftime("%H:%M:%S"))
40     print("La moyenne de l'humidité est %.2f%%"%moyH)
41     print("La moyenne de la température est %.1f°C"%moyT)
42
43
44     for i in range(10) :
45         sommeEcartypeH=pow((Hum[i]-moyH),2)+sommeEcartypeH
46         EcartypeH=pow(0.1*sommeEcartypeH,0.5)
47         print("L'ecart type est %.2f%%"%EcartypeH)
48         ecartypehum.append(EcartypeH)
49 with open('data.csv', mode='w', newline='') as fichier_csv :
50     writer = csv.writer(fichier_csv, delimiter=';')
51     writer.writerow(['Date', 'Heure', 'temperature moyenne', 'humidité moyenne', 'ecart type de l'humid.
52
53     for i in range(10):
54         writer.writerow([date[i], heure[i],moyenneTemperature[i],moyenneHumidite[i],ecartypehum[i]])
55         time.sleep(000)
56 print('fini')
57
```

Après exécution on obtient un fichier excel :

Date	Heure	temperature moyenne	humidité moyenne	ecart type de l'humidité
2023-10-23	09:09:40	25.726167488098145	25.726167488098145	0.1677530439144321
2023-10-23	09:09:45	25.588300895690917	25.588300895690917	0.3181433447253341
2023-10-23	09:09:50	25.667644500732422	25.667644500732422	0.1814029462577579
2023-10-23	09:09:55	25.587738227844238	25.587738227844238	0.11618806779451002
2023-10-23	09:10:00	25.656390380859374	25.656390380859374	0.09947758520440705
2023-10-23	09:10:05	25.67721061706543	25.67721061706543	0.0976401678090252
2023-10-23	09:10:10	25.67355308532715	25.67355308532715	0.09203669629678236
2023-10-23	09:10:15	25.650762939453124	25.650762939453124	0.12531887226078428
2023-10-23	09:10:20	25.66764430999756	25.66764430999756	0.13027459084155954
2023-10-23	09:10:25	25.640914916992188	25.640914916992188	0.1206899516721288

3. Exercice 3

3.1.Question 1

```
EX03.py x
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4
5 R=[255,0,0]
6 G=[0,255,0]
7 B=[0,0,255]
8 Y=[255,255,0]
9 M=[255,0,255]
10 C=[0,255,255]
11
12 co=[R,G,B,Y,M,C]
13 while 1:
14     for i in co :
15         print(i)
16         sense.clear(i)
17         time.sleep(0.25)
18
```

```
Shell
[255, 255, 0]
[255, 0, 255]
[0, 255, 255]
[255, 0, 0]
[0, 255, 0]
[0, 0, 255]
[255, 255, 0]
[255, 0, 255]
[0, 255, 255]
[255, 0, 0]
[0, 255, 0]
[0, 0, 255]
[255, 255, 0]
[255, 0, 255]
[0, 255, 255]
[255, 0, 0]
[0, 255, 0]
[0, 0, 255]
[255, 255, 0]
[255, 0, 255]
[0, 255, 255]
[255, 0, 0]
[0, 255, 0]
```

3.2.Question 2

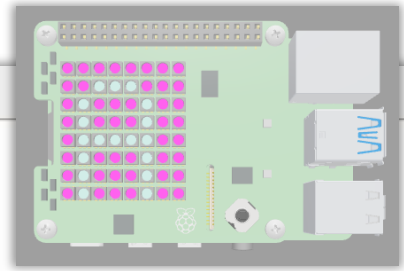
```
EX03.py x
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4 delay=0.0005
5 couleur=[0,0,255]
6 while 1:
7     for i in range (255) :
8         couleur=[i,0,255-i]
9         i=i+1
10        sense.clear(couleur)
11        print(i)
12        time.sleep(delay)
13    for i in range (255) :
14        couleur=[255-i,0,i]
15        i=i+1
16        sense.clear(couleur)
17        print(i)
18        time.sleep(delay)|
```

```
Shell
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

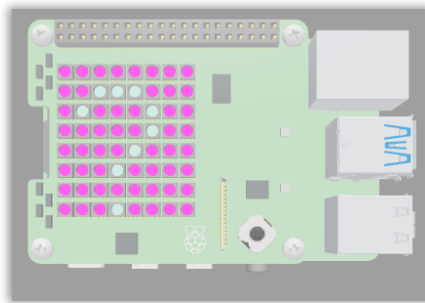
3.3.Question 3 : On affiche le Caractère « A »

EX03.py ✕

```
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4 turquoise=[64,254,208]
5 violet=[255,0,255]
6 sense.show_letter("A", text_colour=turquoise,back_colour=violet)
```



Lorsque l'on essaye d'afficher « ° » ça nous affiche un point d'interrogation « ? »



3.4.Question 4

EX03.py ✕

```
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4 turquoise=[64,254,208]
5 temp=round(sense.get_temperature(),2)
6 sense.show_message(str(temp), text_colour=turquoise,back_colour=[0,0,0])
```

La commande qui permet de trouver un fichier est la suivante **sudo find . -name "sense_hat_text.txt"**

Liste des caractères disponibles à l'écriture :

```
+~*/!"#$%<0123456789.=)(ABCDEFGHIJKLMNQRSTUvwXYZabcdefg hijklmnopqrstuvwxyz?,:|@%[&'"]\~
```


3.5. Question 5

```
EX03.py py.py
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4 rouge=[255,0,0]
5 vert=[0,255,0]
6 orange=[255,255,0]
7 while 1:
8     with open ("/sys/class/thermal/thermal_zone0/temp") as File :
9         line=File.readline()
10        Temp_mC=int(line)
11        temp=round(Temp_mC/1000,2)
12        if temp >40 and temp <60:
13            sense.show_message('CPU : '+str(temp)+'C', text_colour=vert,back_colour=[0,0,0])
14        elif temp >60 and temp <80:
15            sense.show_message('CPU : '+str(temp)+'C', text_colour=orange,back_colour=[0,0,0])
16        else :
17            sense.show_message('CPU : '+str(temp)+'C', text_colour=rouge,back_colour=[0,0,0])
18
```

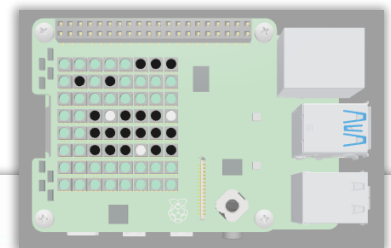
3.6. Question 6

```
EX03.py py.py
1 import time
2 from sense_hat import SenseHat
3 sense=SenseHat()
4 rouge=[255,0,0]
5 vert=[0,255,0]
6 orange=[255,255,0]
7 with open ("/sys/class/thermal/thermal_zone0/temp") as File :
8     line=File.readline()
9     Temp_mC=int(line)
10    temp=round(Temp_mC/1000,2)
11    oldtemp=0
12
13    while 1:
14        print (temp)
15        with open ("/sys/class/thermal/thermal_zone0/temp") as File :
16            line=File.readline()
17            Temp_mC=int(line)
18            temp=round(Temp_mC/1000,2)
19            if (oldtemp-temp) >=2 or (oldtemp-temp) <=-2 :
20                oldtemp=temp
21                if temp >40 and temp <60:
22                    sense.show_message('CPU : '+str(temp)+'C', text_colour=vert,back_colour=[0,0,0])
23
24                elif temp >60 and temp <80:
25                    sense.show_message('CPU : '+str(temp)+'C', text_colour=orange,back_colour=[0,0,0])
26                else :
27                    sense.show_message('CPU : '+str(temp)+'C', text_colour=rouge,back_colour=[0,0,0])
28
```

4. Exercice 4

4.1. Question 1

```
1 from sense_hat import SenseHat
2 sense=SenseHat()
3 sense.load_image('/home/rayouss/Desktop/dragon.png')
4
```



4.2.Question 2

```
EX03.py x py.py x dragon.py * x
1 from sense_hat import SenseHat
2 sense=SenseHat()
3 sense.load_image('/home/rayouss/Desktop/dragon.png')
4 liste_pixels=sense.get_pixels()
5 print(liste_pixels)
6
7 #depuis l'execution du programme
8 couleur1=[32, 152, 80]
9 couleur2=[0, 0, 0]
10 couleur3=[248, 252, 248]
```

Shell

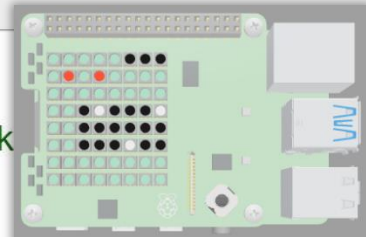
```
>>> %Run dragon.py
```

WARNING:root:Failed to initialise TCS34725 colour sensor. (sensor not present)

4.3.Question 3

```
EX03.py x py.py x dragon.py x <untitled> * x
```


```
1 from sense_hat import SenseHat  
2 sense=SenseHat()  
3 sense.load_image('/home/rayouss/Desktop/Python/Sense_Hat/1.png')  
4 sense.set_pixel(1,1,255,0,0)  
5 sense.set_pixel(3,1,255,0,0)
```

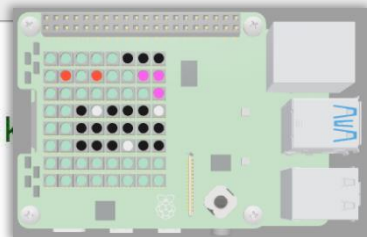


4.4.Question 4

```
EX03.py x py.py x dragon.py x <untitled> * x
```

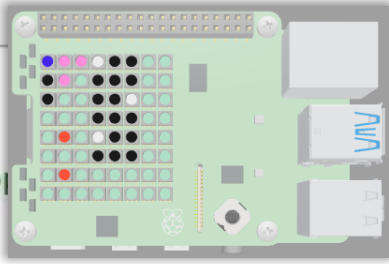
```
1 from sense_hat import SenseHat
2 sense=SenseHat()
3 sense.load_image('/home/rayouss/Desktop/1.png')
4 sense.set_pixel(1,1,255,0,0)
5 sense.set_pixel(3,1,255,0,0)
6 sense.set_pixel(6,1,84,93,1)
7 sense.set_pixel(7,1,84,93,1)
8 sense.set_pixel(7,2,84,93,1)
```





4.5.Question 5 :

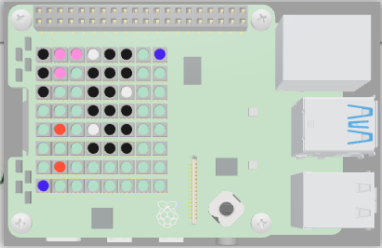
```
dragon.py *X <untitled> *X
1 from sense_hat import SenseHat
2 import time
3 sense=SenseHat()
4 sense.load_image('/home/rayouss/Desktop')
5 sense.set_pixel(1,1,255,0,0)
6 sense.set_pixel(3,1,255,0,0)
7 sense.set_pixel(6,1,255,20,147)
8 sense.set_pixel(7,1,255,20,147)
9 sense.set_pixel(7,2,255,20,147)
10 sense.set_rotation(90)
11 sense.set_pixel(0,0,0,0,255) # pour traquer le pixel (0,0)
12 time.sleep(5)
13 sense.set_rotation(0)
14
```



Le point (0,0) se trouve désormais à la position (7,0) par rapport à la position que l'on avait au départ.

4.6.Question 6 :

```
dragon.py X <untitled> *X
1 from sense_hat import SenseHat
2 import time
3 sense=SenseHat()
4 sense.load_image('/home/rayouss/Desktop')
5 sense.set_pixel(1,1,255,0,0)
6 sense.set_pixel(3,1,255,0,0)
7 sense.set_pixel(6,1,255,20,147)
8 sense.set_pixel(7,1,255,20,147)
9 sense.set_pixel(7,2,255,20,147)
10 sense.set_pixel(0,0,0,0,255) # pour traquer le pixel (0,0)
11 sense.set_rotation(90)
12 time.sleep(5)
13 sense.set_rotation(0)
14 sense.set_rotation(90,False)
15 sense.set_pixel(0,0,0,0,255)# pour traquer le pixel (0,0)
```



4.7.Question 7 :

```
<untitled> *  
1 L1=["a","b","e","t","z"]  
2 L2=L1  
3 L2.pop(0)  
4 L2.insert(0,"c")  
5 print(L1)  
6 print(L2)  
7
```

```
Shell  
>>> %Run -c $EDITOR_CONTENT  
['c', 'b', 'e', 't', 'z']  
['c', 'b', 'e', 't', 'z']  
>>>
```

On peut voir ici que L2 et L1 sont liés. Ainsi, même si $L2 = L1$, L2 n'est pas une copie de L1 mais agit comme un pointeur vers L1. Donc Si l'on modifie L2 alors L1 sera modifié car les deux sont désormais intrinsèquement liés.

4.8.Question 8:

```
<untitled> *  
1 L1=["a","b","e","t","z"]  
2 L3=L1.copy()  
3 L2=L1  
4 L2.pop(0)  
5 L2.insert(0,"c")  
6 print(L1)  
7 print(L2)  
8 print(L3)  
9
```

```
Shell  
>>> %Run -c $EDITOR_CONTENT  
['c', 'b', 'e', 't', 'z']  
['c', 'b', 'e', 't', 'z']  
['a', 'b', 'e', 't', 'z']
```

On remarque ici que l'ajout de la fonction `.copy()` nous permet de copier un 1e élément dans un second sans que les 2 ne soit liés.

4.9.Question 9: Animation du Dragon

Code de départ :

```
from sense_hat import SenseHat
import time

sense=SenseHat()
v=[34, 153, 82] #vert
b=[255, 255, 255] #blanc
n=[0, 0, 0] #noir
delai=0.05

dragonOriginal=[ #on réécrit la matrice du dragon pour faciliter son utilisation
    v,v,v,v,v,n,n,n,
    v,n,v,n,v,v,v,v,
    v,v,v,v,v,v,v,v,
    v,v,n,b,n,n,n,b,
    v,v,n,n,n,n,n,n,
    v,v,n,n,n,b,n,n,
    v,v,v,v,v,v,v,v,
    v,v,v,v,v,v,v,v,
    ]

Afficher=[]
for i in range(64) : #on crée la matrice a afficher qui est au depart vide
    Afficher.append(n)

sense.set_pixels(dragonOriginal) #affichage dragon
time.sleep(1)
```

On crée une fonction qui récupère une des 8 colonnes du dragon original :

```
def recupColonne(copie): #Fonction recupere une colonne depuis dragon original
    colonne=[]
    for i in range(64):
        if i%8==0:#met les elements de la colonne à copier dans le tableau colonne
            colonne.append(dragonOriginal[i+copie])
    return colonne
```

On crée une autre fonction qui récupère une des 8 colonnes du dragon original et la remplace dans une des 8 colonnes de la matrice à afficher

```
#Fonction qui remplace une colonne de la matrice a afficher avec une colonne du dragon original
def changeColonne(coller,copie):
    j=0
    for i in range(64):
        if i%8==0:
            if copie ==8: # met des pixels noirs dans la colonne a remplacer
                Afficher[i+coller]=n
            if copie!=8: # met les elements du tableau colonne dans la colonne a remplacer
                Afficher[i+coller]=recupColonne(copie)[j]
            j=j+1
    return Afficher
```

On crée une fonction qui réinitialise la matrice à afficher

[illegible]

Code apparition du dragon :

```
for i in range(8) : # fonction apparition Dragon
    if i==0:
        sense.set_pixels(changeColonne(i,7)) #Afficher
        colonne 7 dans colonne 0, et eteins le reste
        time.sleep(delai)
        Afficher=reset()
    if i==1:
        changeColonne(i-1,6)
        sense.set_pixels(changeColonne(i,7)) #Afficher
        colonne 7 dans colonne 1, colonne 6 dans 0 et eteins le
        reste
        time.sleep(delai)
        Afficher=reset()
    if i==2:
        changeColonne(i-2,5) # meme logique que avant
        changeColonne(i-1,6)
        sense.set_pixels(changeColonne(i,7))
        time.sleep(delai)
        Afficher=reset()
    if i==3:
        changeColonne(i-3,4)
        changeColonne(i-2,5)
        changeColonne(i-1,6)
        sense.set_pixels(changeColonne(i,7))
        time.sleep(delai)
        Afficher=reset()
    if i==4:
        changeColonne(i-4,3)
        changeColonne(i-3,4)
        changeColonne(i-2,5)
        changeColonne(i-1,6)
```

```

sense.set_pixels(changeColonne(i,7))
time.sleep(delai)
Afficher=reset()
if i==5:
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==6:
    changeColonne(i-6,1)
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==7:
    changeColonne(i-7,0)
    changeColonne(i-6,1)
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()

```

Code disparition du dragon :

```

inv=[7,6,5,4,3,2,1,0] # tableau que i va parcourir pour
la disparition du dragon
for i in inv :
    if i==7:
        changeColonne(7,6)
        changeColonne(6,5)
        changeColonne(5,4)

```

```

changeColonne(4,3)
changeColonne(3,2)
changeColonne(2,1)
changeColonne(1,0)
sense.set_pixels(changeColonne(0,8)) # decale
toute les colonne vers la droite, et eteins colonne 0
time.sleep(delai)
Afficher=reset()
if i==6:
    changeColonne(7,5)
    changeColonne(6,4)
    changeColonne(5,3)
    changeColonne(4,2)
    changeColonne(3,1)
    changeColonne(2,0)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8)) # decale
toute les colonne vers la droite, et eteins colonne 0
et 1
    time.sleep(delai)
    Afficher=reset()
if i==5:
    changeColonne(7,i-1)
    changeColonne(6,i-2)
    changeColonne(5,i-3)
    changeColonne(4,i-4)
    changeColonne(3,i-5)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8)) # meme
logique que avant
    time.sleep(delai)
    Afficher=reset()
if i==4:
    changeColonne(7,i-1)
    changeColonne(6,i-2)
    changeColonne(5,i-3)
    changeColonne(4,i-4)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==3:

```

```
changeColonne(7,i-1)
changeColonne(6,i-2)
changeColonne(5,i-3)
changeColonne(4,8)
changeColonne(3,8)
changeColonne(2,8)
changeColonne(1,8)
sense.set_pixels(changeColonne(0,8))
time.sleep(delai)
Afficher=reset()
if i==2:
    changeColonne(7,i-1)
    changeColonne(6,i-2)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==1:
    changeColonne(7,i-1)
    changeColonne(6,8)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==0:
    changeColonne(7,8)
    changeColonne(6,8)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
```

Code complet :

```
from sense_hat import SenseHat
import time

sense=SenseHat()

v=[34, 153, 82] #vert
b=[255, 255, 255] #blanc
n=[0, 0, 0] #noir

delai=0.05

#on réécrit la matrice du dragon pour faciliter son
utilisation
dragonOriginal=[
    v,v,v,v,v,n,n,n,
    v,n,v,n,v,v,v,v,
    v,v,v,v,v,v,v,v,
    v,v,n,b,n,n,n,b,
    v,v,n,n,n,n,n,n,
    v,v,n,n,n,b,n,n,
    v,v,v,v,v,v,v,v,
    v,v,v,v,v,v,v,v,
]

Afficher=[n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,
,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,
,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,]

sense.set_pixels(dragonOriginal) #affichage dragon
time.sleep(1)

def recupColonne(copie): #Fonction recupere une colonne
depuis dragon original
    colonne=[]
    for i in range(64):
        if i%8==0:
            colonne.append(dragonOriginal[i+copie]) #met les
elements de la colonne à copier dans le tableau colonne
    return colonne
```



```

    Afficher=reset()
if i==3:
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==4:
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==5:
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==6:
    changeColonne(i-6,1)
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)
    Afficher=reset()
if i==7:
    changeColonne(i-7,0)
    changeColonne(i-6,1)
    changeColonne(i-5,2)
    changeColonne(i-4,3)
    changeColonne(i-3,4)
    changeColonne(i-2,5)
    changeColonne(i-1,6)
    sense.set_pixels(changeColonne(i,7))
    time.sleep(delai)

```

```

Afficher=reset()

for i in inv :
    if i==7:
        changeColonne(7,6)
        changeColonne(6,5)
        changeColonne(5,4)
        changeColonne(4,3)
        changeColonne(3,2)
        changeColonne(2,1)
        changeColonne(1,0)
        sense.set_pixels(changeColonne(0,8)) # decale
toute les colonne vers la droite, et eteins colonne 0
        time.sleep(delai)
        Afficher=reset()
    if i==6:
        changeColonne(7,5)
        changeColonne(6,4)
        changeColonne(5,3)
        changeColonne(4,2)
        changeColonne(3,1)
        changeColonne(2,0)
        changeColonne(1,8)
        sense.set_pixels(changeColonne(0,8)) # decale
toute les colonne vers la droite, et eteins colonne 0
et 1
        time.sleep(delai)
        Afficher=reset()
    if i==5:
        changeColonne(7,i-1)
        changeColonne(6,i-2)
        changeColonne(5,i-3)
        changeColonne(4,i-4)
        changeColonne(3,i-5)
        changeColonne(2,8)
        changeColonne(1,8)
        sense.set_pixels(changeColonne(0,8)) # meme
logique que avant
        time.sleep(delai)
        Afficher=reset()
    if i==4:
        changeColonne(7,i-1)
        changeColonne(6,i-2)
        changeColonne(5,i-3)
        changeColonne(4,i-4)

```

```

changeColonne(3,8)
changeColonne(2,8)
changeColonne(1,8)
sense.set_pixels(changeColonne(0,8))
time.sleep(delai)
Afficher=reset()
if i==3:
    changeColonne(7,i-1)
    changeColonne(6,i-2)
    changeColonne(5,i-3)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==2:
    changeColonne(7,i-1)
    changeColonne(6,i-2)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==1:
    changeColonne(7,i-1)
    changeColonne(6,8)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)
    changeColonne(2,8)
    changeColonne(1,8)
    sense.set_pixels(changeColonne(0,8))
    time.sleep(delai)
    Afficher=reset()
if i==0:
    changeColonne(7,8)
    changeColonne(6,8)
    changeColonne(5,8)
    changeColonne(4,8)
    changeColonne(3,8)

```

```
changeColonne(2,8)
changeColonne(1,8)
sense.set_pixels(changeColonne(0,8))
time.sleep(delai)
Afficher=reset()
```

..... FIN