Programmation d'une entrée

définir le mode de fonctionnement de la broche

```
GPIO.setup(23,GPIO.IN) # on déclare la pin 23 comme une entrée

GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) # on affecte une resisitace de pull-down , résistence entre l'entrée du signal et la masse)

#Une méthode d'acquisition de l'état de l'entrée :
```

Créer une variable

```
etat_entre_23 == 0  # variable crée et mise a l'état 0
```

Acquisition de la valeur de l'entée

```
GPIO.input(23) == etat_entre_23  # lecture de l'état de l'entrée et enregistrement dans la variable
```

Le Raspberry Pi

Les opérateurs pour les conditions :

== égal à
!= différent de
< plus petit que
> plus grand que
<= inférieur ou égal à
>= supérieur ou égal à

Conditions multiples :

and (ET)
or (OU)
in / not in (DANS / PAS DANS)

Mots clé des conditions :

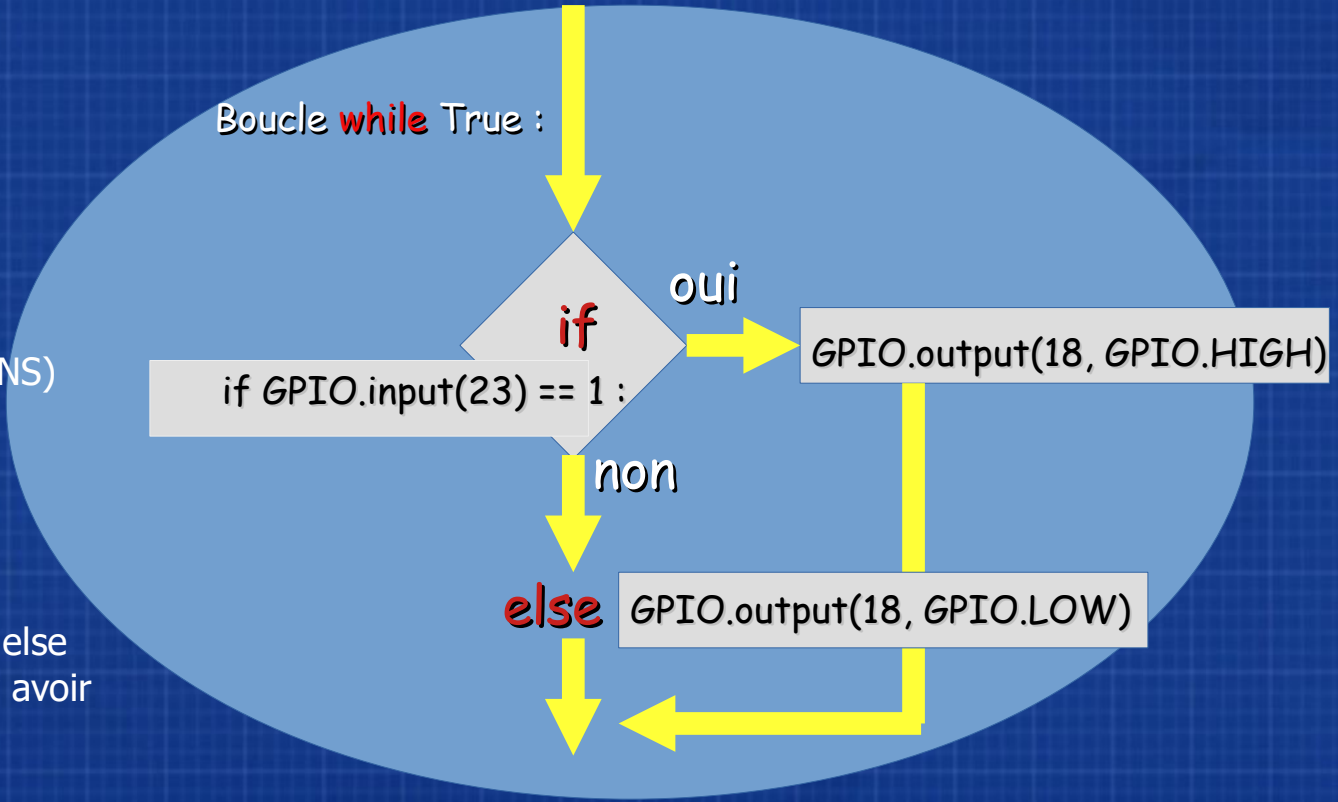
if SI
If not SI N'EST PAS
else SINON un seul else
elif SINON SI il peut y avoir plusieurs elif

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(23, GPIO.IN)
GPIO.output(18, GPIO.LOW)
```

Programmation

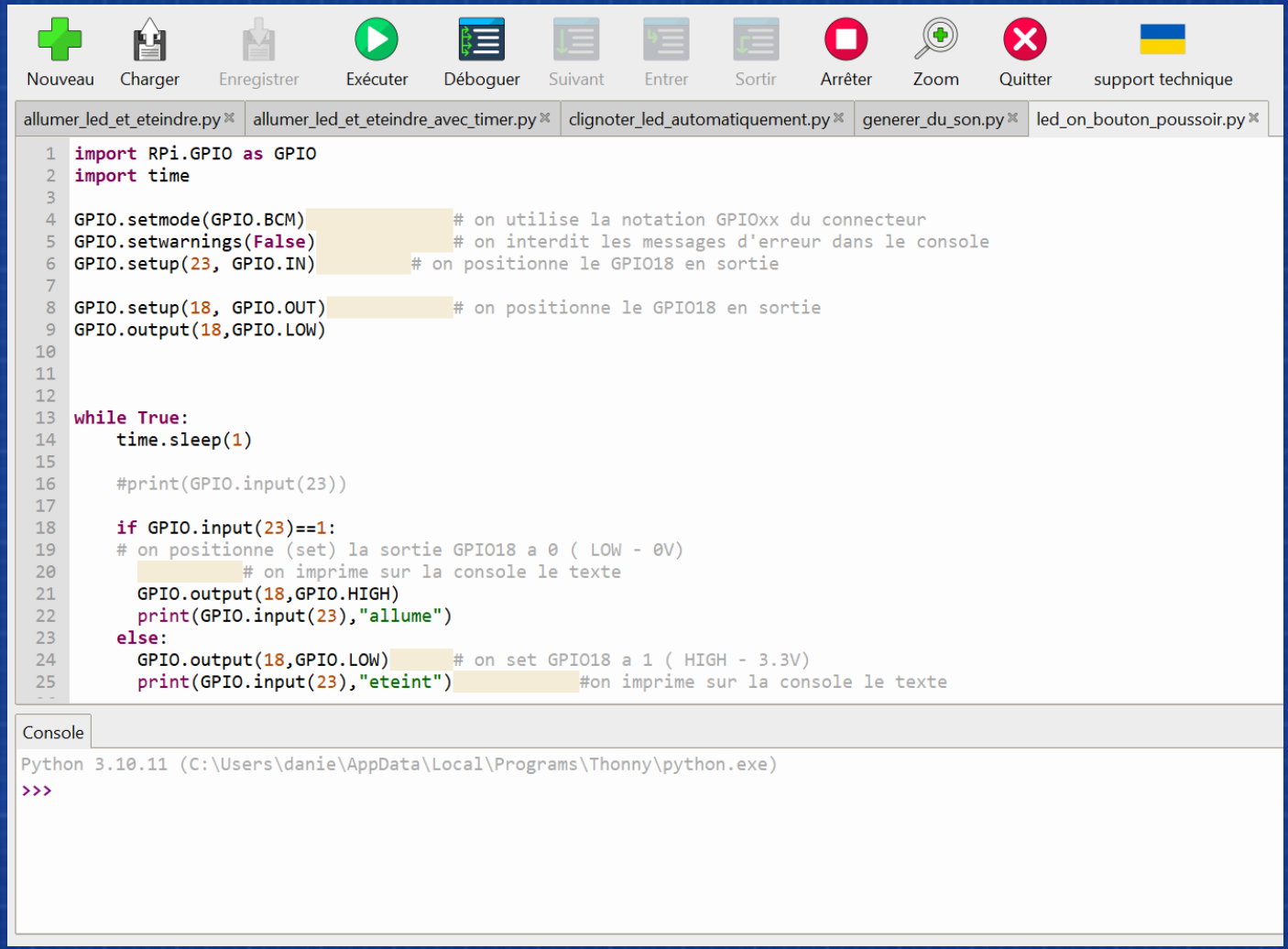
Combinaisons des conditions :

if 0 < age >= 100 :



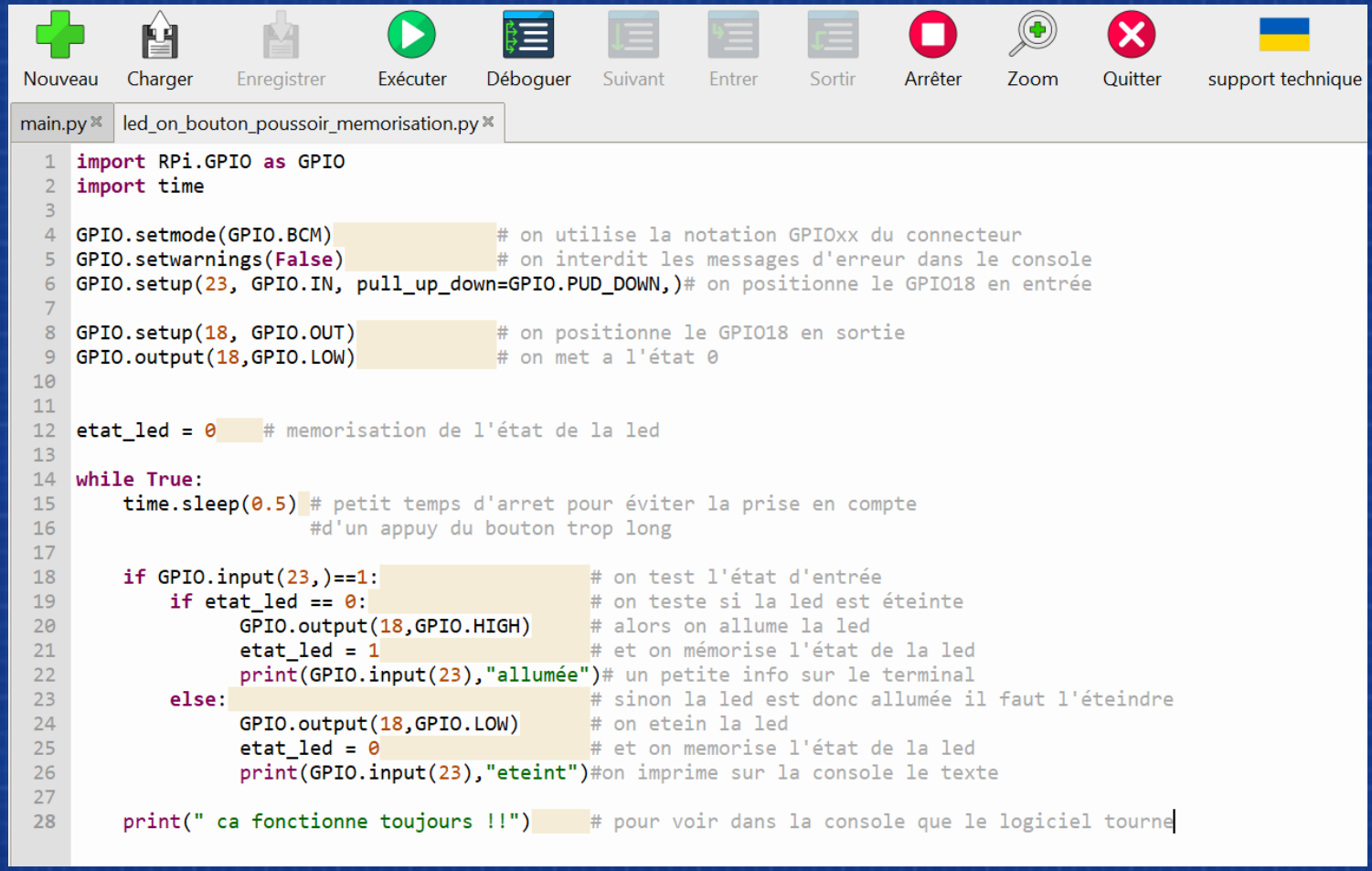
Le Raspberry Pi

ALLUMER LED AVEC BOUTON POUSSOIR APPUYE



Le Raspberry Pi

ALLUMER/ETEINDRE LED PAR IMPULSION SUR BP




```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BCM)              # on utilise la notation GPIOxx du connecteur
5  GPIO.setwarnings(False)             # on interdit les messages d'erreur dans le console
6  GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN,)# on positionne le GPIO18 en entrée
7
8  GPIO.setup(18, GPIO.OUT)            # on positionne le GPIO18 en sortie
9  GPIO.output(18,GPIO.LOW)            # on met a l'état 0
10
11
12 etat_led = 0     # memorisation de l'état de la led
13
14 while True:
15     time.sleep(0.5) # petit temps d'arret pour éviter la prise en compte
16                     #d'un appuy du bouton trop long
17
18     if GPIO.input(23,)==1:                  # on test l'état d'entrée
19         if etat_led == 0:                   # on teste si la led est éteinte
20               GPIO.output(18,GPIO.HIGH)     # alors on allume la led
21               etat_led = 1                  # et on mémorise l'état de la led
22               print(GPIO.input(23),"allumée")# un petite info sur le terminal
23         else:                               # sinon la led est donc allumée il faut l'éteindre
24               GPIO.output(18,GPIO.LOW)      # on etein la led
25               etat_led = 0                  # et on memorise l'état de la led
26               print(GPIO.input(23),"eteint")#on imprime sur la console le texte
27
28     print(" ca fonctionne toujours !!")     # pour voir dans la console que le logiciel tourne
```