Université IBN TOFAIL – Kénitra
ECOLE NATIONALE DES SCIENCES APPLIQUEES

Cours Systèmes d'Exploitation Embarqués

CH4: Python sur Raspberry Pi

Par:

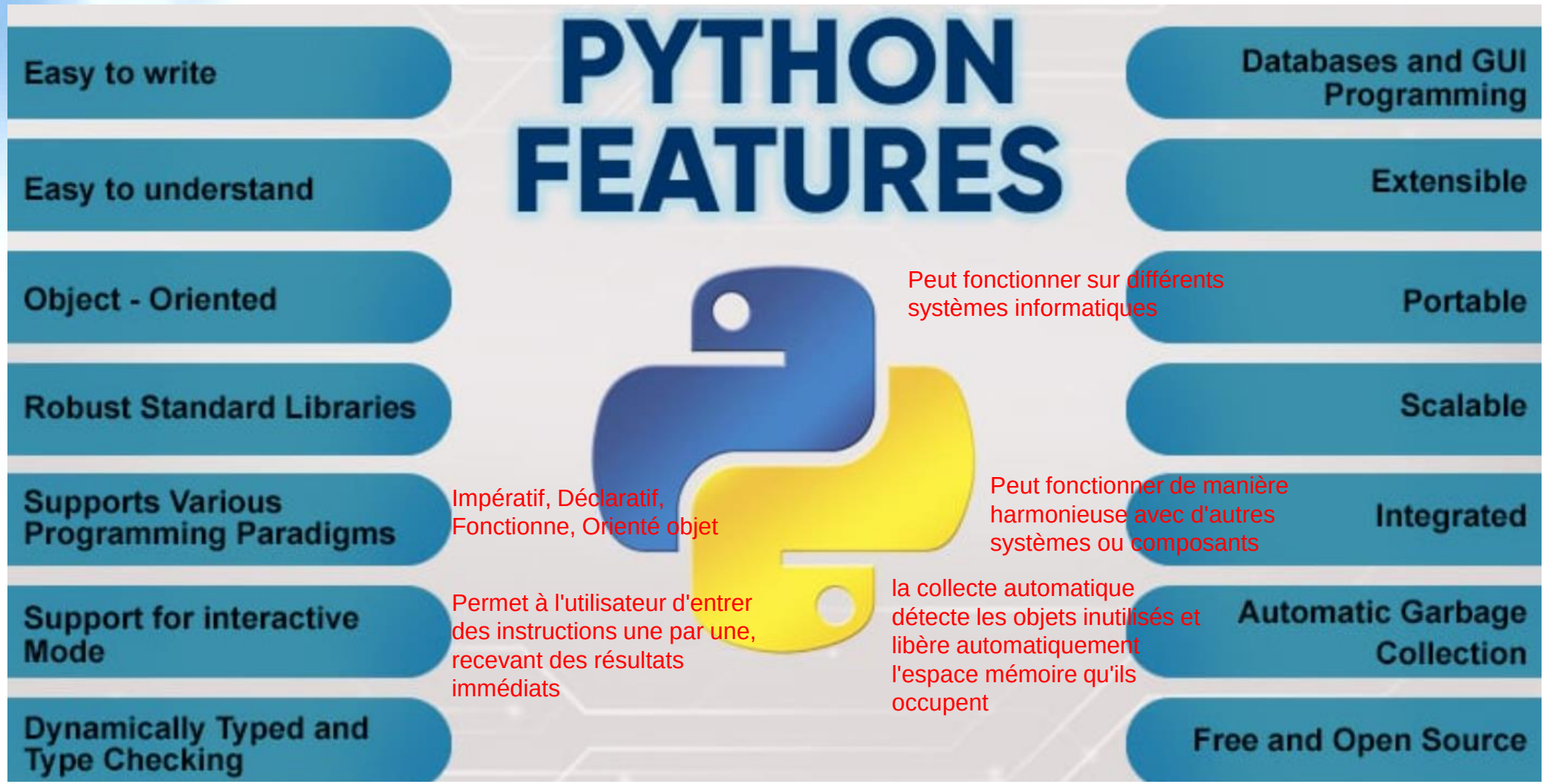
Hassan EL FADIL

elfadil.h@gmail.com

1. Présentation de Python

- **Création de Python :** En février 1991 Guido van Rossum a publié la première version de Python, la version 0.9.0.
- **Le nom "Python" :** inspiré par l'amour de Guido van Rossum pour les Monty Python, un groupe comique britannique. Il voulait un nom court, unique et légèrement mystérieux pour son nouveau langage
- **Open Source et Communauté :** En 2000, Python 2.0 a été publié, et à partir de là, le développement de Python est devenu de plus en plus ouvert et basé sur la communauté. Python Software Foundation (PSF) a été créée en 2001 pour promouvoir, protéger et développer Python.
- **Versions ultérieures :** Plusieurs versions se sont succédées. Actuellement : Python 3

2. Caractéristiques de Python



…Caractéristiques de Python

Python est un langage de programmation polyvalent et puissant qui est largement utilisé dans divers domaines. Ses caractéristiques principales comprennent :

1. **Lisibilité du Code :** La syntaxe de Python est conçue pour être claire et lisible. L'utilisation de **l'indentation** pour délimiter les blocs de code encourage une programmation structurée.
2. **Facilité d'Apprentissage :** Python est souvent recommandé comme langage d'initiation à la programmation en raison de sa syntaxe simple et de son apprentissage facile.
3. **Orientation Objet :** Python prend en charge la programmation orientée objet (POO), permettant aux développeurs de créer et d'utiliser des classes et des objets pour organiser le code de manière modulaire et réutilisable.

…Caractéristiques de Python

4. **Vaste Bibliothèque Standard :** Python est livré avec une bibliothèque standard riche, fournissant des modules et des packages pour effectuer une variété de tâches, de l'interaction réseau à la manipulation de fichiers, en passant par la gestion des threads.
5. **Utilisation de différentes paradigmes de programmation:** Programmation Impérative, Programmation Déclarative, POO, Programmation Fonctionnelle, …
6. **Langages de Programmation Supportant l'Interactivité:** les utilisateurs peuvent entrer des instructions ou des commandes directement et les voir immédiatement exécutées (utilisation d'un CLI)
7. **Interprété et Dynamiquement Typé :** Python est un langage interprété, ce qui signifie que le code peut être exécuté directement sans nécessiter de compilation préalable. Il est également dynamiquement typé, ce qui permet une flexibilité élevée lors de l'assignation des types de données aux variables.

…Caractéristiques de Python

8. **GUI programming" (Graphical User Interface programming):** fait référence à la programmation d'interfaces utilisateur graphiques.
9. **Extensible**: Capacité à être étendu, modifié, ou enrichi facilement sans nécessiter de modifications majeures de sa structure fondamentale.
10. **Portable :** Peut être exécuté sur différentes plates-formes informatiques sans nécessiter de modifications majeures
11. **Evolutif** (Scalable): Peut être étendu ou adapté pour répondre à une demande croissante de manière efficace.
12. **Support pour Intégration avec d'autres Langages :** Python offre des mécanismes pour intégrer du code écrit dans d'autres langages tels que C et C++, ce qui permet d'utiliser des bibliothèques existantes et d'optimiser les performances dans des cas spécifiques.

…Caractéristiques de Python

13. **Gestion Automatique de la Mémoire** (automatic garbage collection): Python dispose d'un gestionnaire de mémoire automatique qui simplifie la gestion des ressources.

14. **Open Source :** Python est distribué sous une licence open source, ce qui signifie que le code source est accessible, modifiable et distribuable gratuitement.

15. **Communauté Active :** Python bénéficie d'une communauté mondiale active qui contribue au développement du langage, partage des bibliothèques open source et fournit un support via des forums, des listes de diffusion et des événements communautaires.

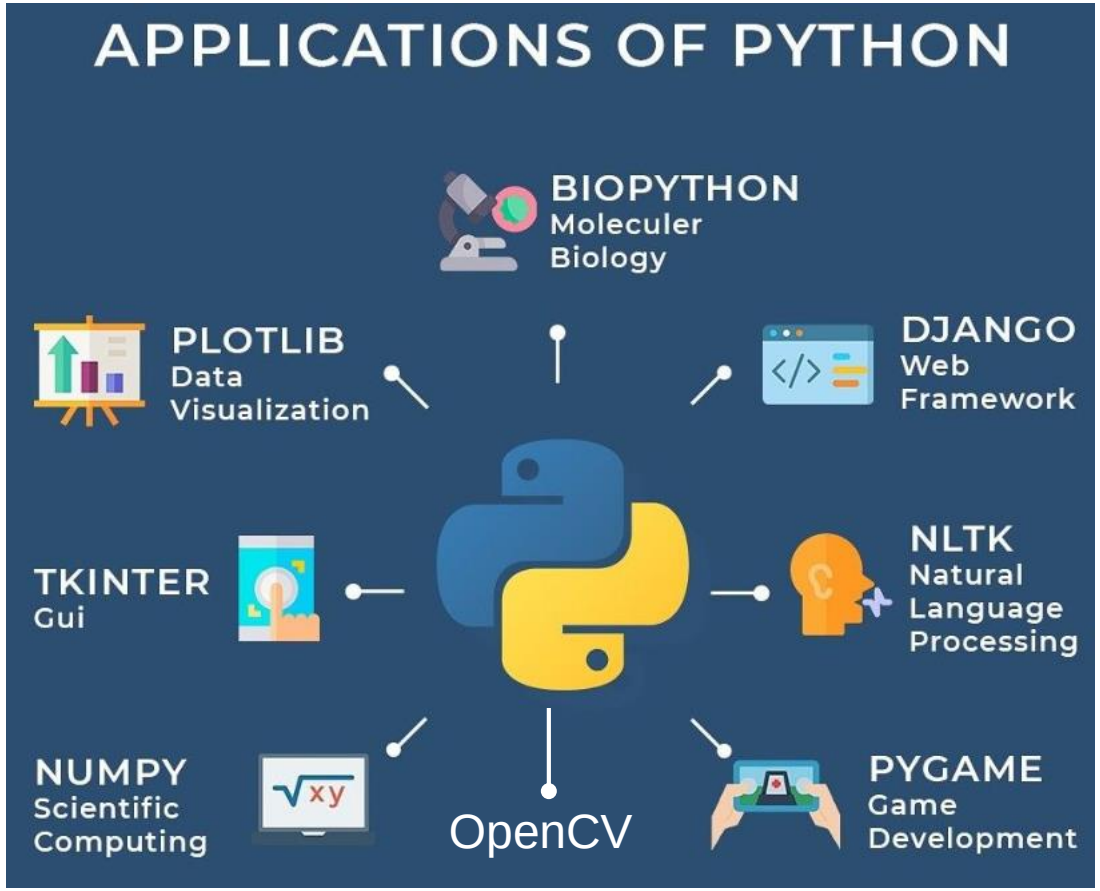
…Caractéristiques de Python

16. **Polyvalence :** Python est un langage polyvalent utilisé dans une variété de domaines tels que le développement web, l'automatisation, la science des données, l'intelligence artificielle, le développement de jeux, etc. Sa polyvalence en fait un choix populaire pour les développeurs.

17. **Support Math et IA**: Python est doté de fonctionnalités, de bibliothèques ou de frameworks qui facilitent le travail dans les domaines des mathématiques et de l'intelligence artificielle (IA)

3. Domaines d'applications

Les caractéristiques précédentes font de Python un langage puissant et attrayant pour les développeurs, que ce soit pour les projets débutants ou pour des applications complexes à grande échelle.



4. Installation de Python sur le Raspberry Pi

- L'installation de Python sur un Raspberry Pi est généralement un processus simple.
- Le Raspberry Pi est souvent livré avec une version de Python préinstallée, mais vous pouvez également mettre à jour vers une version plus récente si nécessaire

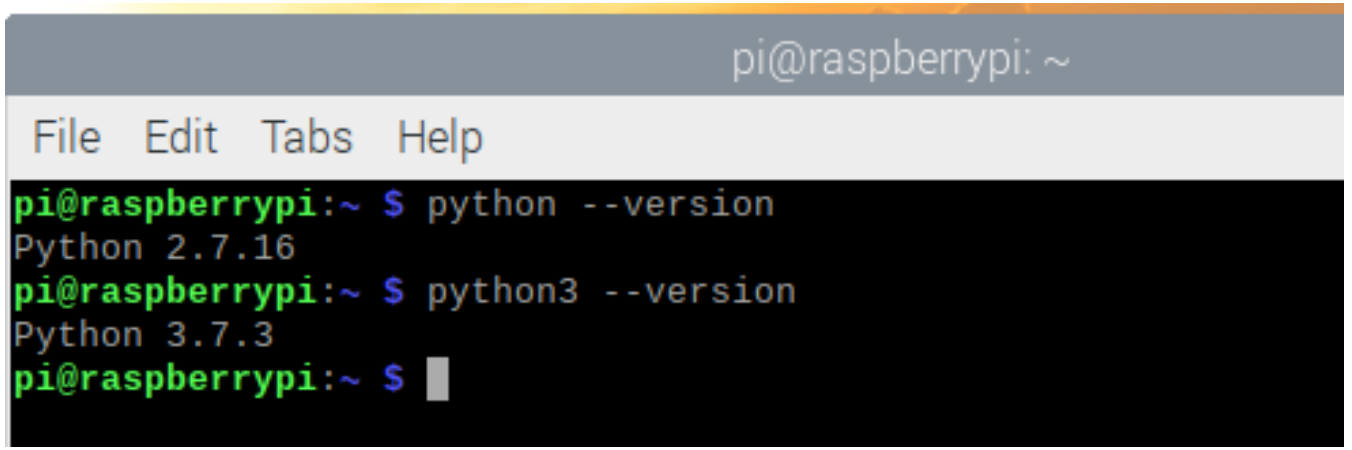
… Installation de Python sur le Raspberry Pi

1 Vérification de la Version Python Installée:

Pour vérifier si Python est déjà installé sur votre Raspberry Pi et connaître sa version, ouvrez le terminal et tapez la commande suivante :

`python --version` OU `python3 --version`

Cette commande renvoie la version de Python 2 ou la version par défaut de Python sur votre système.

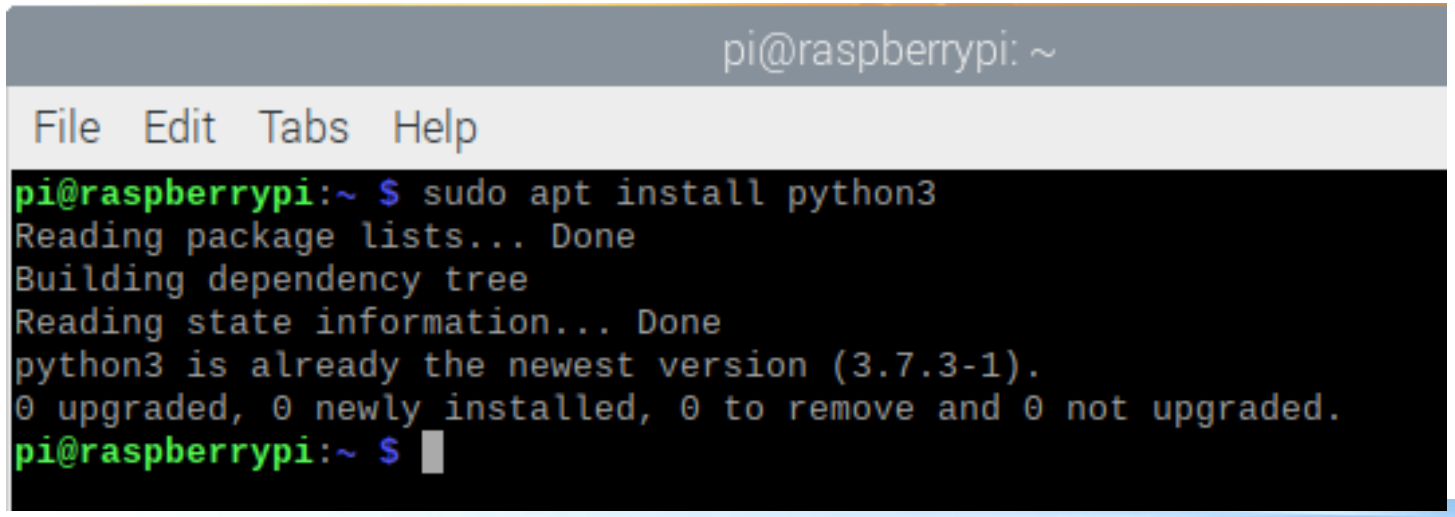


… Installation de Python sur le Raspberry Pi

2 Installation de Python 3 :

Si vous souhaitez mettre à jour vers la dernière version, utilisez la commande suivante pour installer Python 3:

```
sudo apt update
sudo apt install python3
```



… Installation de Python sur le Raspberry Pi

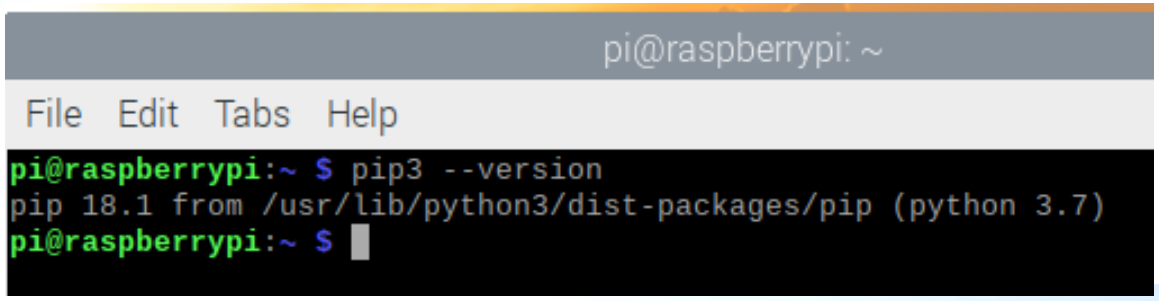
3 Installation de pip (Gestionnaire de Paquets Python) :

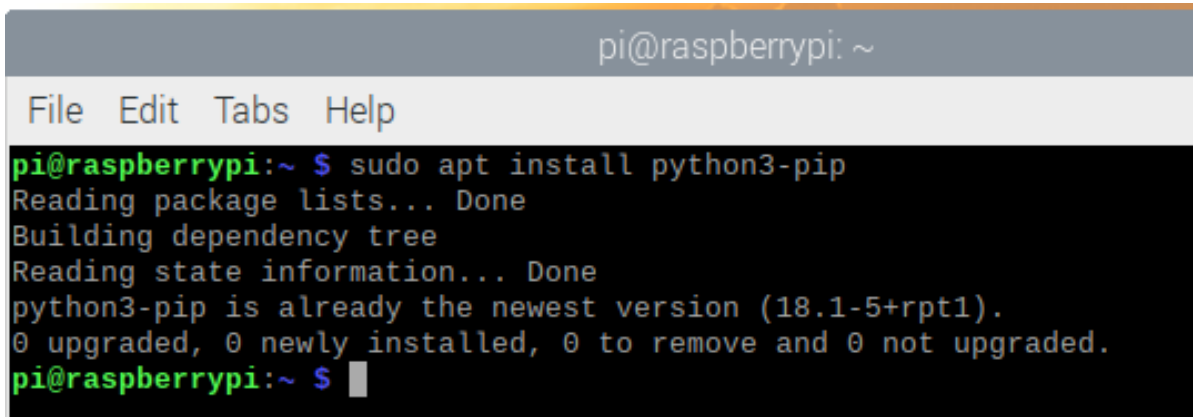
Le gestionnaire de paquets Python, **pip**, est souvent utile pour installer des bibliothèques et des modules. Installez-le avec la commande :

```
sudo apt install python3-pip
```

4 Vérification de l'Installation de pip

```
pip3 --version
```





5. Elaboration d'un programme Python

Si vous souhaitez écrire et exécuter un programme Python sur un Raspberry Pi, suivez ces étapes :

1 Choisissez un éditeur de texte ou un IDE:

- Utiliser un IDE (Environnement de développement intégré) comme **Thonny**, **IDLE**,
- ou même éditer directement dans un éditeur de texte (comme **nano** ou **vim**) pour créer vos scripts Python.

2 Écrivez votre programme :

- Utilisez votre éditeur de texte ou IDE pour créer un nouveau fichier avec une extension "**.py**" (par exemple, "**mon_programme.py**").
- Écrivez votre code Python dans ce fichier.

…Elaboration d'un programme Python

3 Enregistrez le fichier:

- Sauvegardez le fichier après avoir écrit votre code.

4 Exécutez le programme :

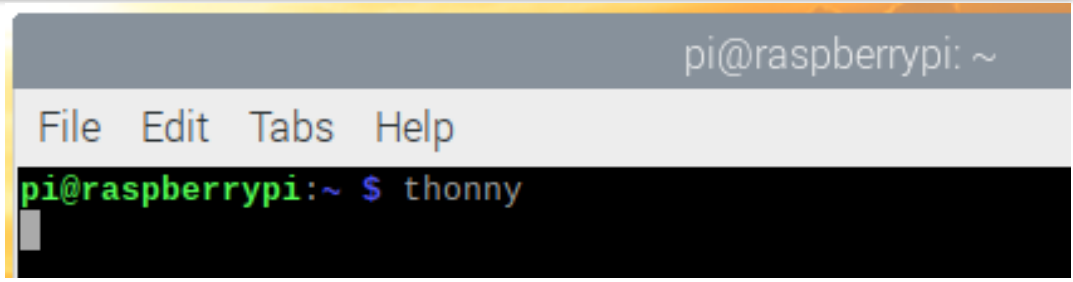
- Ouvrez un terminal et utilisez la commande suivante pour exécuter votre script Python:

```
python mon_programme.py     # Pour Python 2
python3 mon_programme.py    # Pour Python 3
```

***Remarque**: Assurez-vous d'être dans le répertoire où se trouve votre fichier Python, ou spécifiez le chemin complet du fichier.*

…Elaboration d'un programme Python

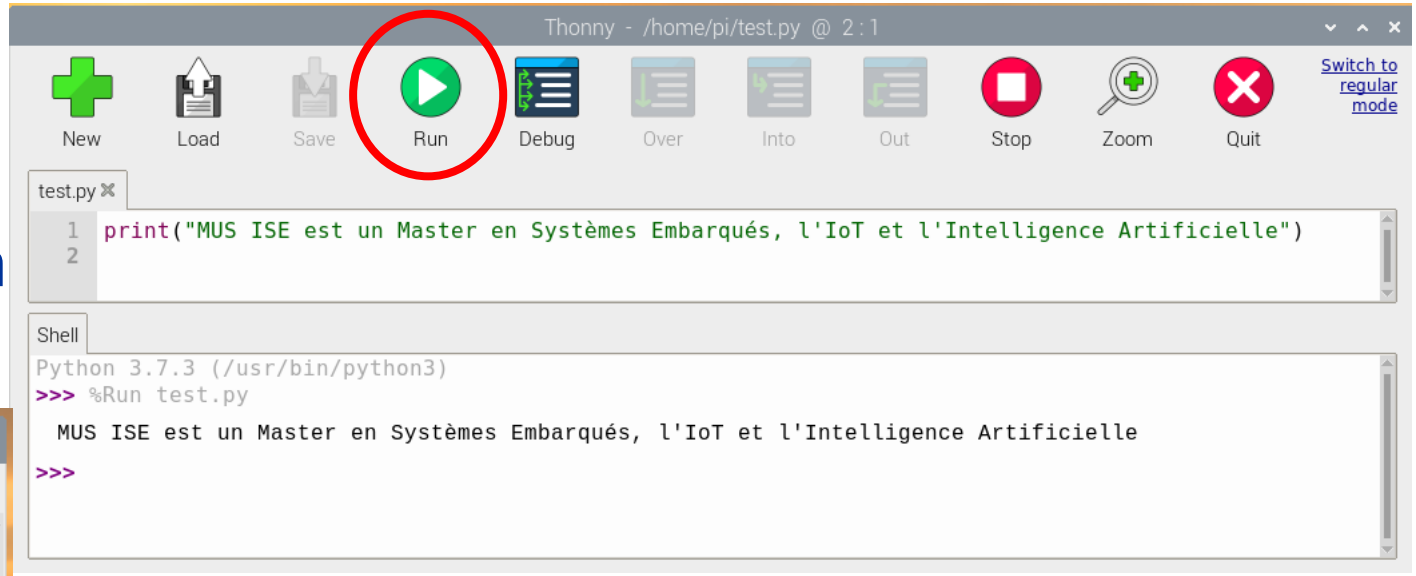
Exemple d'un programme:

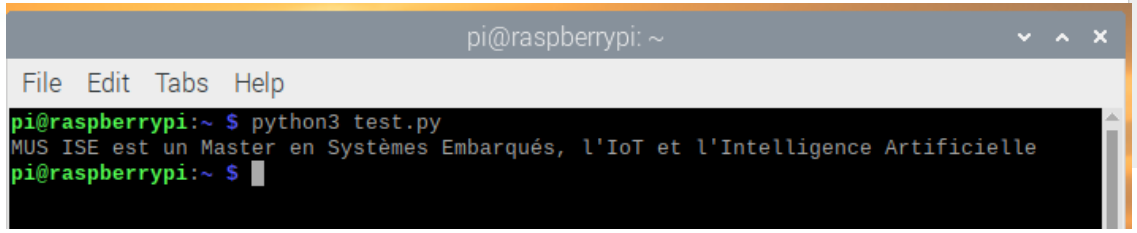


- Lancer thonny

Assurez-vous que Thonny est installé sur votre Raspberry Pi. Si ce n'est pas le cas, vous pouvez l'installer en utilisant la commande suivante dans le terminal : **sudo apt-get install thonny**

- Ecrire le petit programme ci-contre
- Enregistrer le fichier: **test.py**
- Exécuter le programme (appui sur **Run**) ou taper: **python3 test.py** en ligne de commande



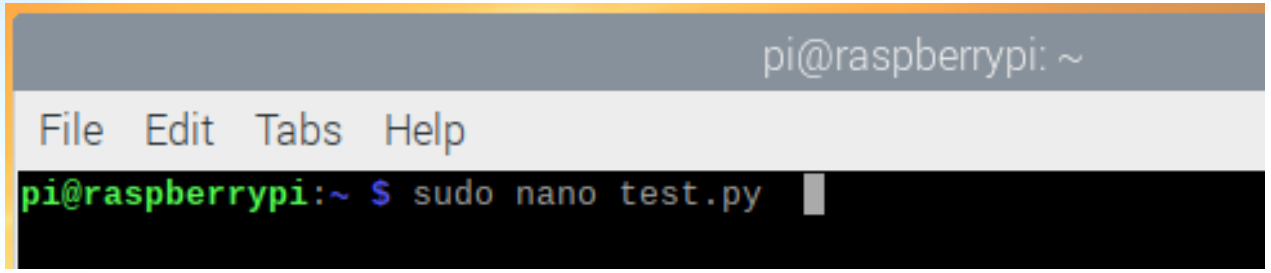


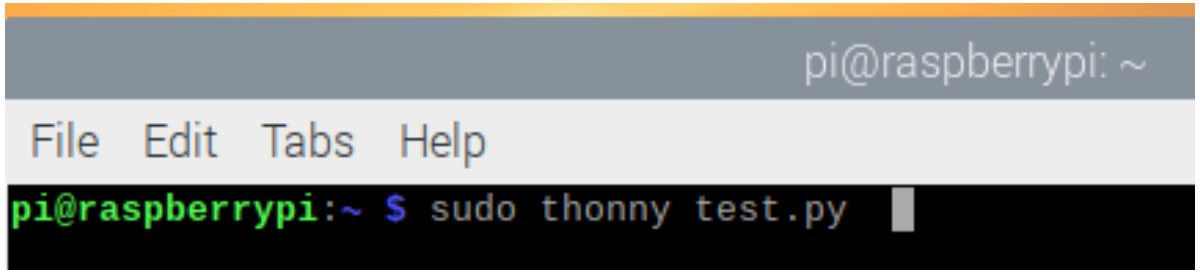
…Elaboration d'un programme Python

Exemple d'un programme:

- Pour ouvrir le fichier **test.py**, vous pouvez utiliser l'une des commandes suivantes:

```
sudo nano test.py
sudo thonny test.py
```





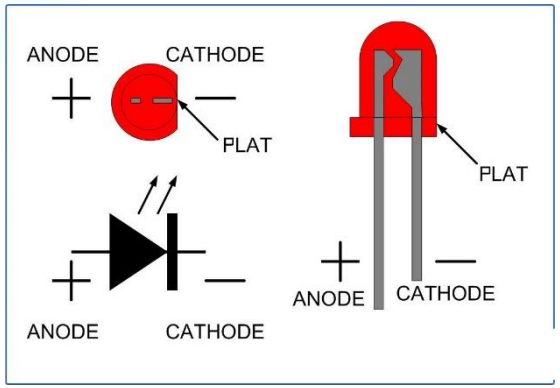
- Pour arrêter : **ctrl+c**

6. Les GPIOs

Programme du clignotement d'une LED

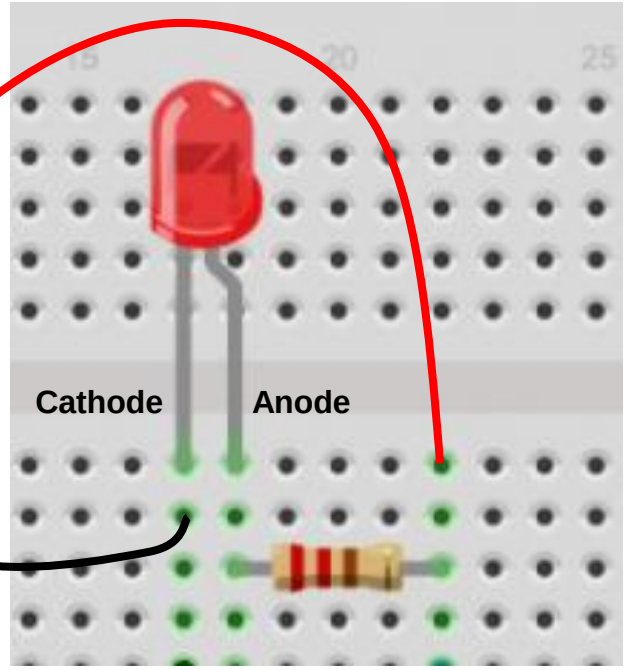
Il s'agit de programmer le clignotement de la LED à un fréquence désirée.

On utilisera **GPIO26 (pin 37)** comme sortie.

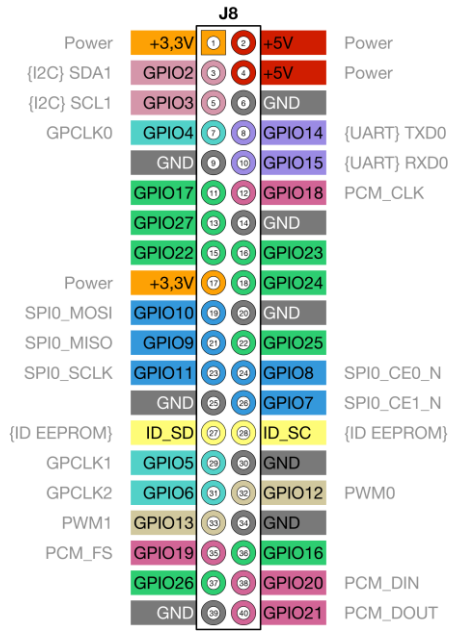


Pour la LED: $If_{max} = 20mA$

$$R > \frac{(3.3 - 0.7)}{If_{max}} = 130\Omega$$



$R = 220\Omega$



Les GPIOs: Programme de clignotement d'une LED

- Assurez-vous d'avoir la bibliothèque GPIO de Raspberry Pi installée (**RPi.GPIO**).
  - Lancez Python avec la commande : python3
  - Puis tapez dans l'interpréteur : import RPi.GPIO as GPIO
  - **Si aucune erreur** ⇒ la bibliothèque est bien installée.

- Si ce n'est pas le cas, vous pouvez l'installer en utilisant la commande suivante dans le terminal de votre Raspberry Pi

```
pip install RPi.GPIO
```

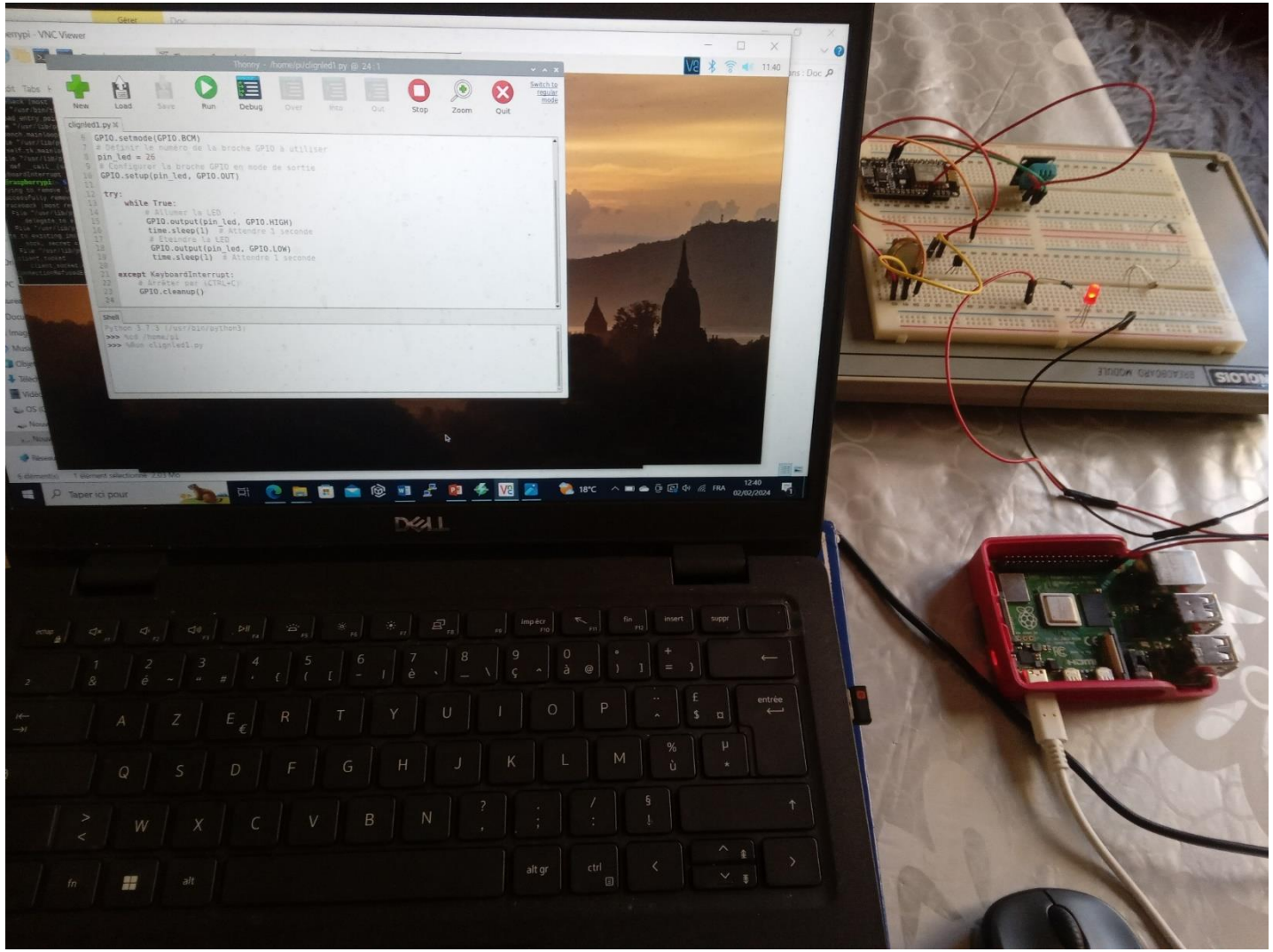
Enregistrer le fichier sous le nom: **clignled1.py**

Sert à **réinitialiser l'état des broches GPIO** que vous avez utilisées dans votre script

```
import RPi.GPIO as GPIO
import time
# Désactiver les avertissements de la bibliothèque GPIO
GPIO.setwarnings(False)
# Définir le mode de numérotation des broches
GPIO.setmode(GPIO.BCM)
# Définir le numéro de la broche GPIO à utiliser
pin_led = 26
# Configurer la broche GPIO en mode de sortie
GPIO.setup(pin_led, GPIO.OUT)

try:
    while True:
        # Allumer la LED
        GPIO.output(pin_led, GPIO.HIGH)
        time.sleep(1)  # Attendre 1 seconde
        # Éteindre la LED
        GPIO.output(pin_led, GPIO.LOW)
        time.sleep(1)  # Attendre 1 seconde

except KeyboardInterrupt:
    # Arrêter par (CTRL+C)
    GPIO.cleanup()
```

Les GPIOs: Programme de clignotement d'une LED

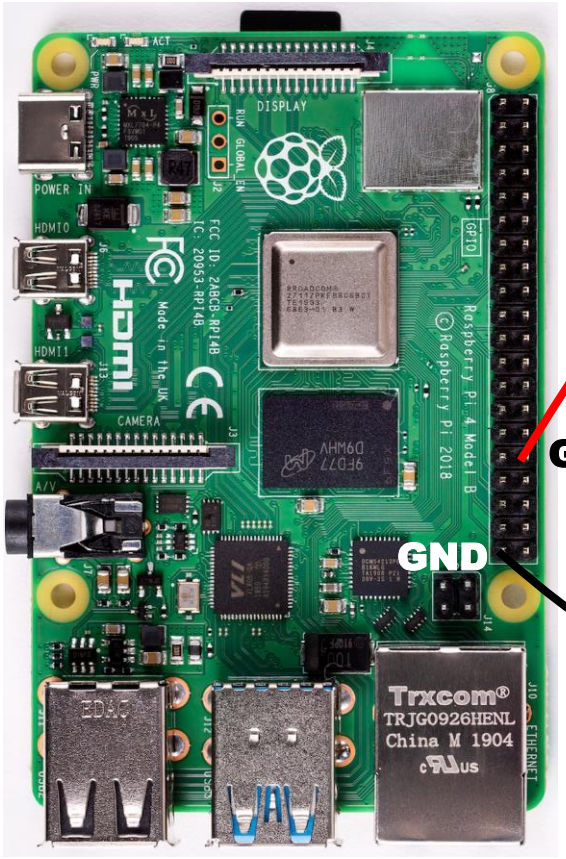
Banc Expérimental:

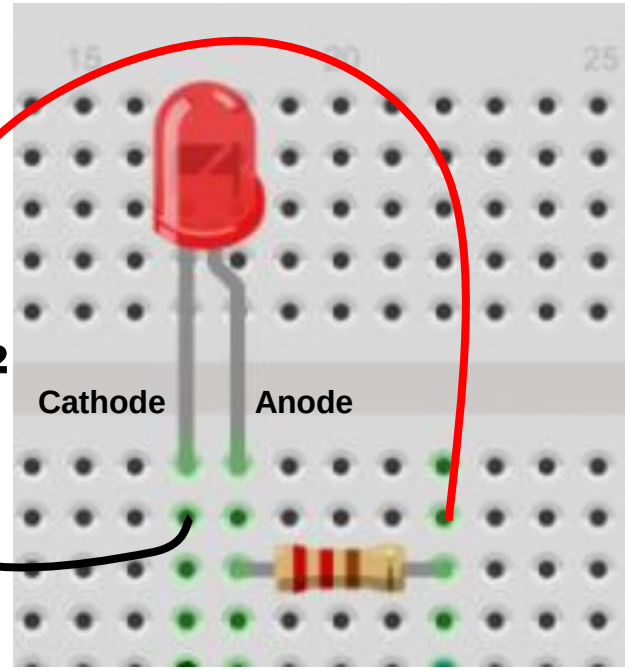


Les GPIOs : PWM: Variation de la Luminosité d'une LED

Il s'agit de faire varier la luminosité d'une LED en utilisant la PWM.

Pour cela on utilisera la broche **PWM0** (**GPIO12**)





$R = 220\Omega$

Les GPIOs : PWM: Variation de la Luminosité d'une LED

Utilisation de la bibliothèque gpiozero:

gpiozero est une bibliothèque Python qui facilite l'utilisation des GPIO sur des cartes Raspberry Pi. Elle permet de contrôler des composants électroniques tels que des LED, des boutons, des capteurs, des moteurs, etc.

Si elle n'est pas installé, taper:

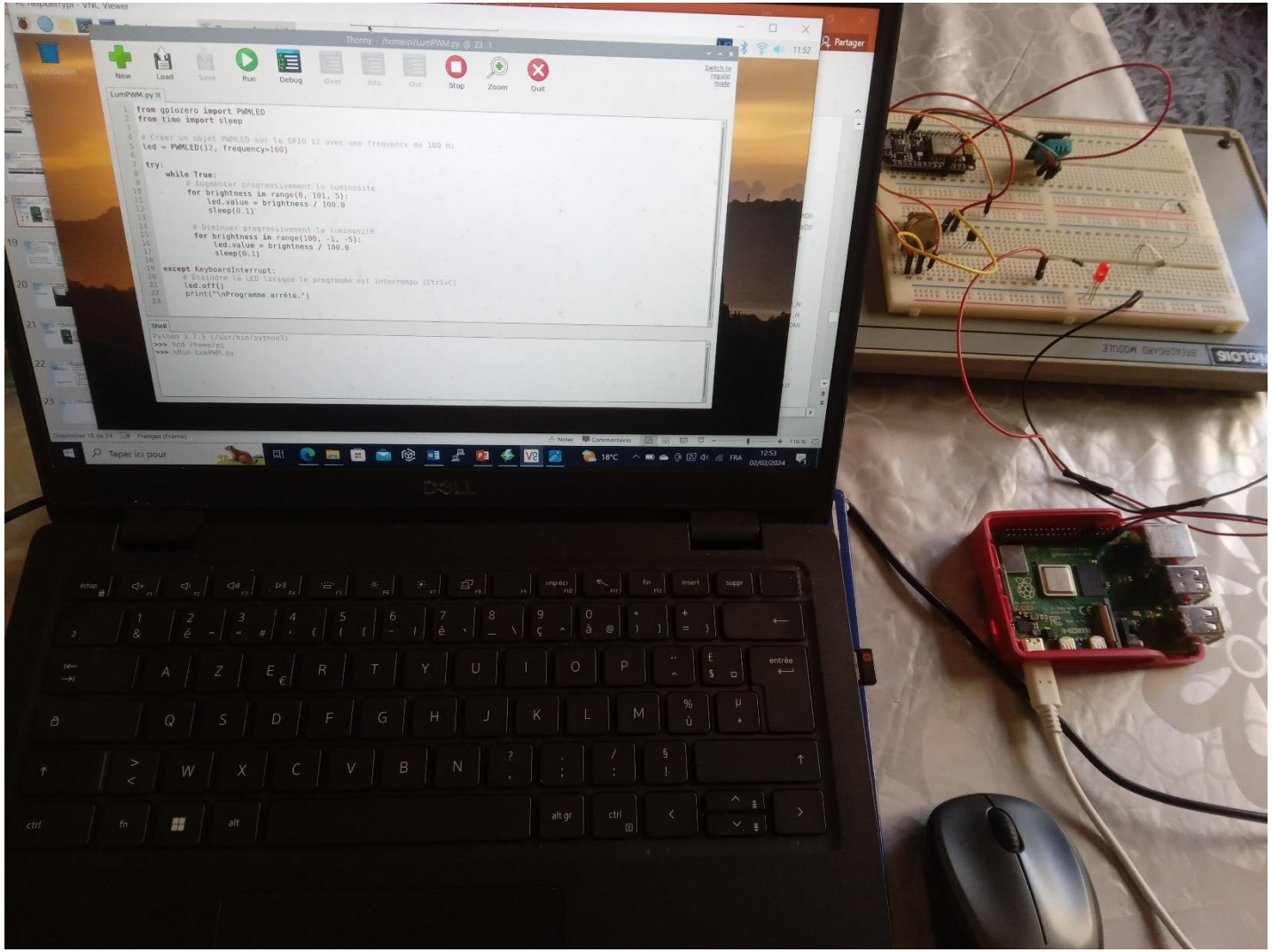
```
pip install gpiozero
```

Enregistrer le fichier sous le nom:
LumPWM.py

```
from gpiozero import PWMLED
from time import sleep

# Créer un objet PWMLED sur le GPIO 12 avec une fréquence de 100 Hz
led = PWMLED(12, frequency=100)

try:
    while True:
        # Augmenter progressivement la luminosité
        for brightness in range(0, 101, 5):
            led.value = brightness / 100.0
            sleep(0.1)

        # Diminuer progressivement la luminosité
        for brightness in range(100, -1, -5):
            led.value = brightness / 100.0
            sleep(0.1)

except KeyboardInterrupt:
    # Éteindre la LED lorsque le programme est interrompu (Ctrl+C)
    led.off()
```

Les GPIOs : PWM: Variation de la Luminosité d'une LED

Banc Expérimental:



7. Exécuter plus d'un programme de contrôle au même temps

Code 1: led1.py

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
while True:
        GPIO.output(11, True)
        time.sleep(2)
        GPIO.output(11, False)
        time.sleep(2)
```

Code 2: led2.py

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(13, GPIO.OUT)
while True:
        GPIO.output(13, True)
        time.sleep(1)
        GPIO.output(13, False)
        time.sleep(1)
```

Exécuter les deux programme en ajoutant le symbole **&** à la fin de chaque ligne:

```
sudo python led1.py &
sudo python led2.py &
```

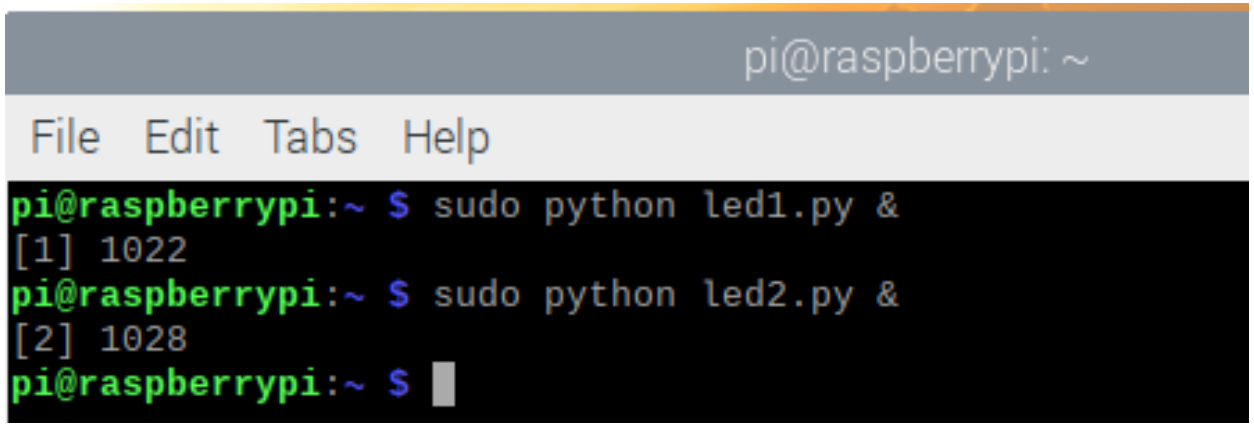
… Exécuter plus d'un programme de contrôle au même temps

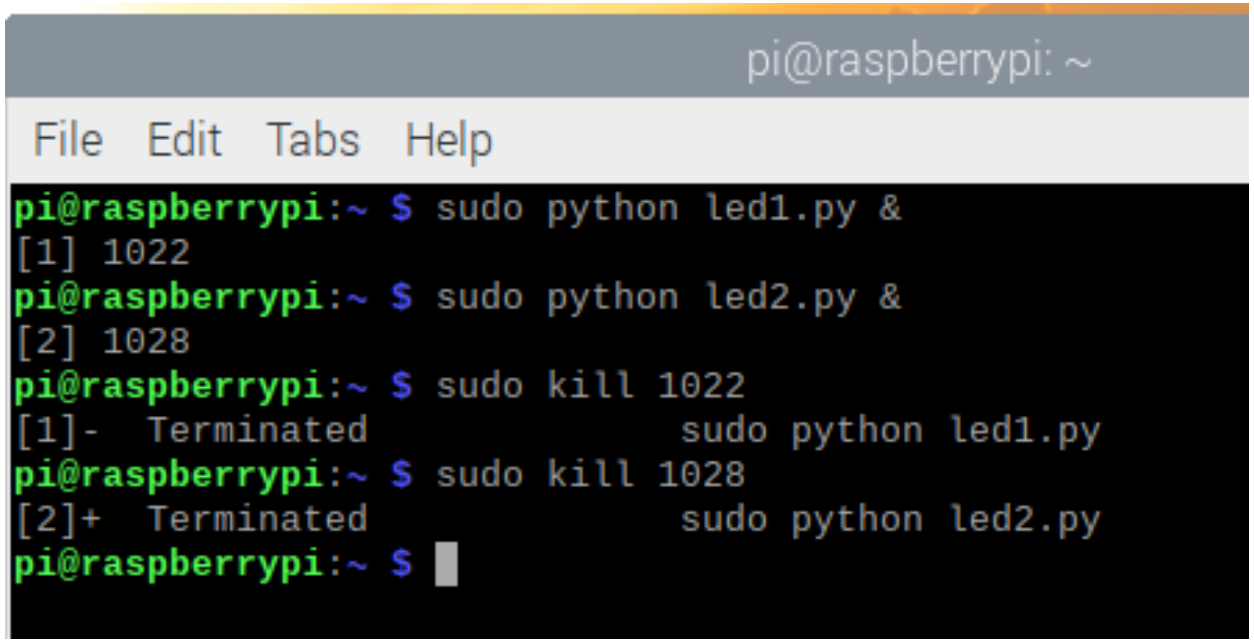
On vous renvoie les informations:

- **[1]** : c'est le numéro du processus en arrière-plan.
- **1022** : c'est le numéro d'identification général du processus.

Cette information vous permet de tuer le processus avec **kill** si nécessaire

```
sudo kill 1022
sudo kill 1028
```



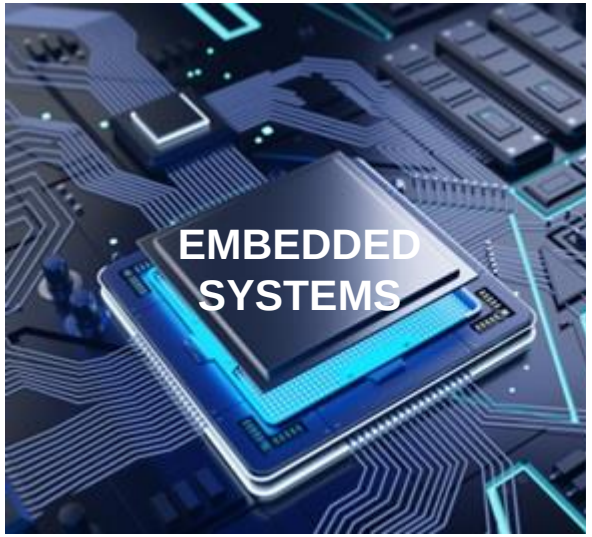


8. Simulateur Raspberry Pi Pico

Pour ceux qui ne disposent pas de cartes Raspberry Pi, ils peuvent utiliser un simulateur Wokwi: https://wokwi.com/pi-pico

Wokwi est une plateforme en ligne pour la simulation de circuits électroniques et de microcontrôleurs. Elle permet aux utilisateurs de concevoir, tester et déboguer leurs projets électroniques directement dans un navigateur web. Voici quelques-unes de ses caractéristiques principales :

1. **Support des Microcontrôleurs** : Wokwi prend en charge des microcontrôleurs populaires comme Arduino, ESP32 et Raspberry Pi Pico, entre autres.
2. **Bibliothèque de Composants**: La plateforme offre une large gamme de composants, y compris des capteurs, des actionneurs, des afficheurs et d'autres périphériques, qui peuvent être facilement ajoutés à vos projets.
3. **Éditeur de Code** : Wokwi inclut un éditeur de code intégré où vous pouvez écrire et téléverser du code vers vos microcontrôleurs simulés.
4. **Simulation en Temps Réel** : Vous pouvez voir des simulations en temps réel de vos circuits, vous aidant à visualiser comment votre projet se comportera dans le monde réel.
5. **Partage et Collaboration** : Les projets peuvent être partagés avec d'autres, permettant la collaboration et les retours d'expérience.



FIN !
Questions ?