

**Table des matières**

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Première Séance</b>	<b>2</b>
2.1	Le Raspberry Pi 5 . . . . .	2
2.2	Montage électronique . . . . .	2
2.3	Analogie électrique . . . . .	4
2.3.1	Le courant électrique . . . . .	4
2.3.2	La tension électrique . . . . .	4
2.4	Programmation Python . . . . .	4
2.4.1	Démarrage . . . . .	4
2.4.2	Commandes python . . . . .	5
2.4.3	Script python . . . . .	5
2.4.4	Script avec boucle . . . . .	6
2.4.5	Montage et contrôle de deux LEDs . . . . .	6
2.4.6	Eteindre le RPi 5 . . . . .	7
<b>3</b>	<b>Deuxième Séance</b>	<b>8</b>
3.1	Montage électronique . . . . .	8
3.2	Programmation Python . . . . .	8
3.2.1	Démarrage . . . . .	8
3.2.2	Commandes python . . . . .	8
3.2.3	Fonctions élémentaires . . . . .	10
3.2.4	Script python . . . . .	10
3.2.5	Eteindre le RPi 5 . . . . .	11
<b>4</b>	<b>Troisième Partie</b>	<b>11</b>
<b>5</b>	<b>Annexe à destination des encadrants</b>	<b>12</b>
5.1	Matériel . . . . .	12
5.2	Configuration du Raspberry Pi . . . . .	12
5.3	Références et ressources . . . . .	13

## 1 Introduction

Durant la première séance, nous allons découvrir comment il est possible d'allumer et d'éteindre une LED en réalisant un petit montage électronique et en écrivant un script python simple. Cette séance est nécessaire pour comprendre quelques bases pour la suite !

Dans la deuxième séance, nous commencerons enfin à utiliser le ruban LED et à le contrôler avec des scripts python toujours assez simples. Enfin, dans la dernière séance, nous continuerais notre exploration sur le ruban LED, chacun à son rythme, on pourra apprendre à créer des effets lumineux plus complexes, et même essayer de les activer via une application téléphone !

## 2 Première Séance

### 2.1 Le Raspberry Pi 5

Au cours de nos trois séances, nous allons utiliser un nano-ordinateur (un ordinateur miniature “simplifié”) appelé **Raspberry Pi 5** (RPi 5), qui nous permettra de contrôler une LED simple, ou plusieurs LEDs sur un ruban. Il est d’abord vendu seul, sous forme d’une carte unique de la taille d’une carte de crédit, appelée carte mère, mais en fonction du projet à réaliser, il est possible d’ajouter des cartes et des composants électroniques complémentaires comme une caméra si on veut filmer, des roues pour faire une voiture télécommandée, ou comme dans notre cas un ruban LED !

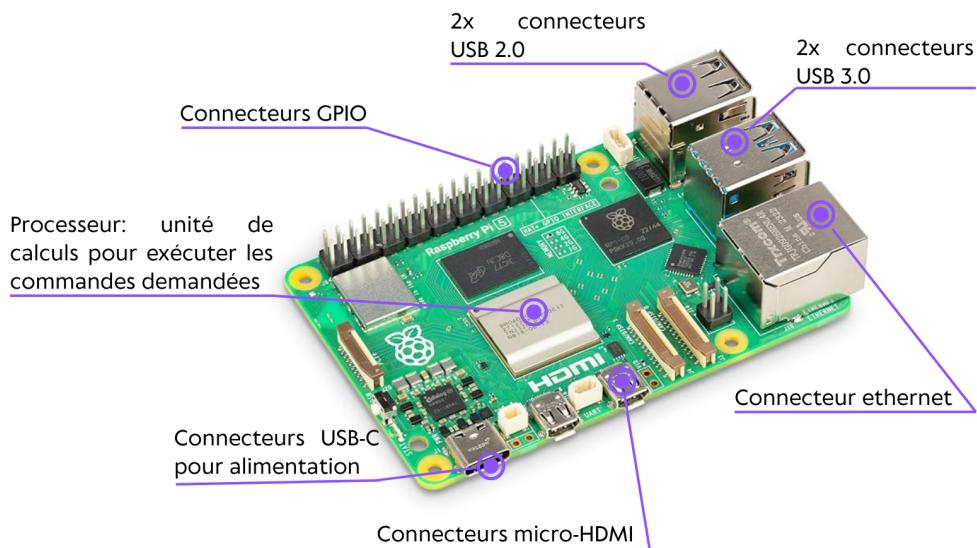


Figure 1: Le Raspberry Pi 5

En général, ces composants supplémentaires se connectent sur les pins (broches) **GPIO = General Purpose Input/Output**. C’est sur ces pins que l’on va connecter nos LEDs; elles permettent au RPi 5 de communiquer aux LEDs les commandes que nous allons envoyer.

### 2.2 Montage électronique

Pour certains montages électroniques, souvent lorsque l’on est au début d’un projet et que l’on fait ce qu’on appelle du “prototypage”, on a besoin d’une carte intermédiaire pour connecter les différents

composants électroniques, appelée carte à trous.

Une carte à trous est une plaque en plastique avec plein de petits trous dans lesquels on peut insérer des fils et des composants électroniques (sans avoir besoin de souder). À l'intérieur de la carte à trous, les trous sont connectés entre eux par des bandes métalliques, ce qui permet de faire des connexions électriques facilement.

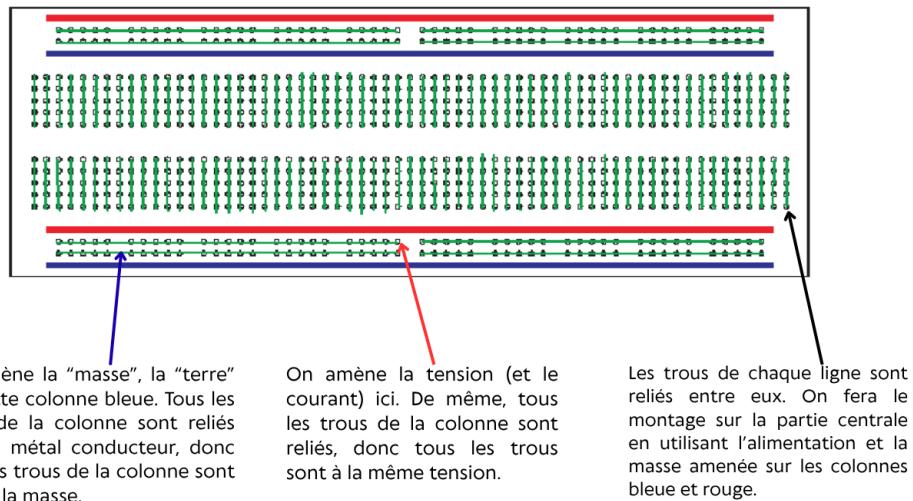


Figure 2: Présentation d'une breadboard (carte à trous)

On va connecter la LED à la pin GPIO numéro 16 en suivant le schéma de montage de la figure 3.

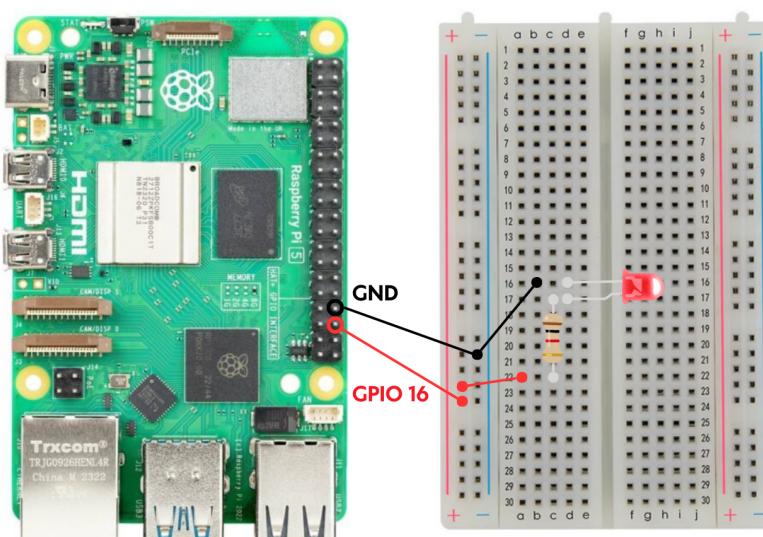


Figure 3: Montage électronique

On remarque qu'en plus de la LED (en rouge), il y a une résistance (en beige). La résistance sert à limiter le courant électrique qui traverse la LED, pour éviter de l'endommager.

## 2.3 Analogie électrique

### 2.3.1 Le courant électrique

Imagine un tuyau rempli d'eau. Chaque goutte d'eau représente un électron. Quand tu ouvres le robinet, les gouttes se déplacent d'un bout à l'autre du tuyau. Dans un circuit électrique, c'est pareil : les électrons (les gouttes d'eau) arrivent par la source d'alimentation et doivent ressortir en rejoignant la terre (ou la masse). Augmenter le courant, C'est comme augmenter le débit d'eau dans le tuyau : plus il y a de gouttes (d'électrons) qui circulent par seconde, plus le courant est fort.

### 2.3.2 La tension électrique

La tension électrique est comme la pression de l'eau dans le tuyau. Plus la pression est élevée, plus l'eau (ou les électrons) est poussée avec force à travers le tuyau (ou le circuit).

Augmenter la tension, C'est comme monter le tuyau plus haut, au 5ème étage d'un immeuble par exemple : l'eau (les électrons) aura plus d'énergie pour circuler, même si le tuyau est un peu plus étroit ou plus long.

## 2.4 Programmation Python

### 2.4.1 Démarrage

Une fois le montage électronique réalisé, on peut brancher le clavier, la souris, et le câble micro-hdmi au Raspberry Pi. Puis, on peut brancher le câble d'alimentation pour allumer le Raspberry Pi 5.

Il va s'allumer, ou "booter", et afficher l'écran de sélection de l'utilisateur. On va sélectionner:

- Utilisateur : **illumine**
- Mot de passe : **illumine**

Après quelques secondes, le bureau du Raspberry Pi 5 va s'afficher. On va ouvrir le terminal (icône noire avec un > blanc) pour pouvoir taper des commandes.

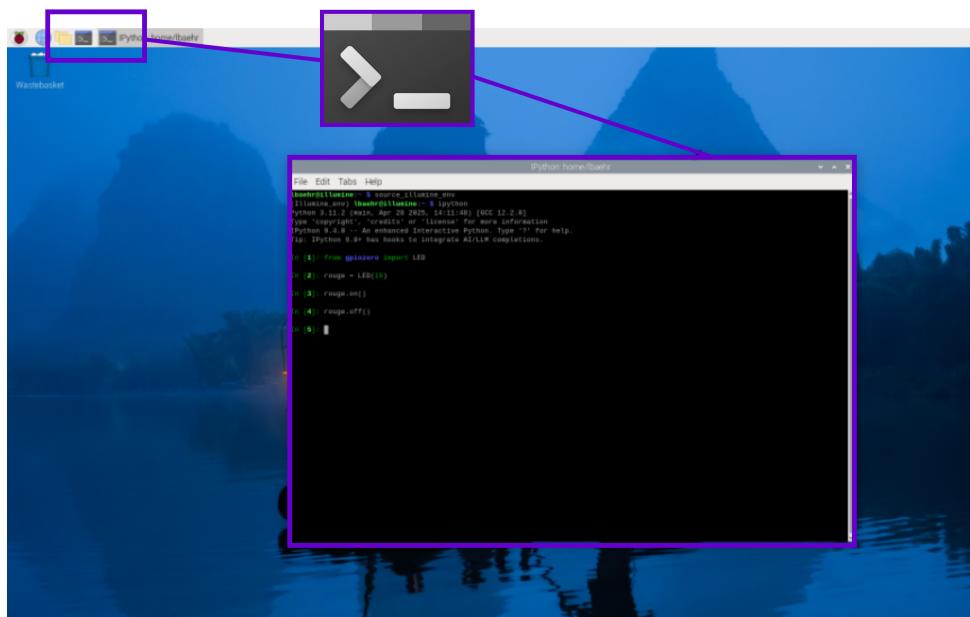


Figure 4: Bureau du Raspberry Pi 5

### 2.4.2 Commandes python

Pour contrôler la LED, on va utiliser le langage de programmation **Python**. Pour comprendre ce qu'est un langage de programmation, on peut faire une analogie avec le jeu "Jacques a dit". Dans ce jeu, un joueur donne des instructions aux autres joueurs, "debout", "saute", "tourne-toi", etc. Les autres joueurs doivent suivre ces instructions. De la même manière, on va donner des instructions à l'ordinateur, sauf qu'il ne parle pas la même langue que nous, c'est pour ça qu'on utilise un langage de programmation comme Python.

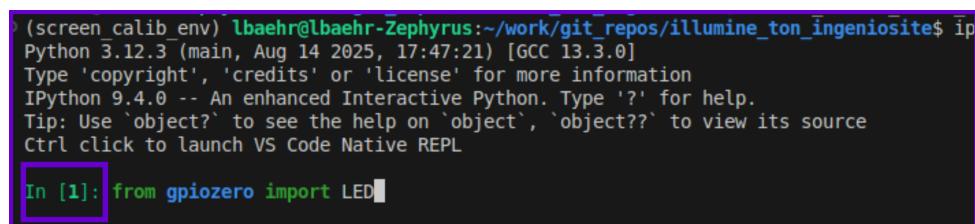
Les premières commandes qu'on va lui donner sont, en français :

- "Place toi dans un environnement spécifique"
- "Rassemble tes outils"
- "Connecte toi à la LED"
- "Allume la LED"
- "Eteins la LED"

En Python, ça donne:

```
cd rpi5_projects/git_repos/illumine/python/1ere_seance/
source_illumine_env
ipython
```

Après ces deux premières commandes, on est dans un environnement Python interactif qui ressemble à ça:



```
(screen_calib_env) lbaehr@lbaehr-Zephyrus:~/work/git_repos/illumine_ton_ingeniosite$ ip
Python 3.12.3 (main, Aug 14 2025, 17:47:21) [GCC 13.3.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.4.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: Use `object?` to see the help on `object`, `object??` to view its source
Ctrl click to launch VS Code Native REPL
In [1]: from gpiozero import LED
```

Figure 5: Environnement Python interactif

On peut maintenant taper les commandes suivantes pour contrôler la LED:

```
from gpiozero import LED
rouge = LED(16)
rouge.on()
rouge.off()
```

Lance ces commandes une par une dans le terminal en appuyant sur la touche "Entrée" et observe ce qu'il se passe !

### 2.4.3 Script python

On a réussi à allumer et éteindre la LED en tapant les commandes une par une, mais ce n'est pas très pratique. Si on doit faire ça 10, ou 100 fois, ça va vite devenir lassant. On peut donc automatiser ça en écrivant un script Python, c'est-à-dire un fichier texte qui contient toutes les commandes à exécuter, et qui est donné en entier à l'ordinateur. C'est comme une recette de cuisine ! Toutes les étapes sont listées dans l'ordre et l'ordinateur les suit, comme si c'était toi qui faisait un gâteau.

On va d'abord sortir de l'environnement python interactif en faisant `exit`

Toujours dans le terminal, on va ouvrir un éditeur de texte et créer un fichier avec la commande:

```
gedit led_control.py
```

Une fenêtre va s'ouvrir (patience, ça prend parfois du temps!), c'est l'éditeur de texte. On va écrire les commandes suivantes dans cette fenêtre:

```
from gpiozero import LED  
from time import sleep
```

Le reste à toi de deviner ! Le but est que la LED s'allume au moins 5 fois. On va utiliser ce qu'on a fait avant, ainsi que 3 commandes données ici dans le désordre:

```
sleep(1)  
print("LED ON")  
print("LED OFF")
```

Une fois le script terminé, on l'enregistre et on le ferme. Pour le donner à l'ordinateur, c'est-à-dire "l'exécuter", on tape dans le terminal:

```
python3 led_control.py
```

Essaye d'exécuter ton script plusieurs fois !

#### 2.4.4 Script avec boucle

On a vu que le script permet de donner toutes les commandes d'un coup à l'ordinateur, mais on a quand même dû écrire toutes les commandes pour allumer et éteindre la LED 5 fois. Si on voulait le faire 100 fois, ça ferait beaucoup de lignes à écrire ! Heureusement, en programmation, il existe des structures qui permettent de répéter des actions sans avoir à les écrire plusieurs fois, ça s'appelle des **boucles**.

On va modifier le script précédent pour utiliser une boucle **for** qui va répéter l'allumage et l'extinction de la LED 5 fois. La structure à utiliser est la suivante:

```
n = #nombre de répétitions  
for i in range(n):  
    # instructions à répéter
```

Essaye de modifier ton script en utilisant cette structure pour allumer et éteindre la LED autant de fois que tu veux !

Une fois le script modifié, enregistre-le et exécute-le à nouveau avec la commande:

```
python3 led_control.py
```

Amuse-toi à changer le nombre de répétitions et la durée d'allumage/extinction de la LED pour voir ce que ça donne !

#### 2.4.5 Montage et contrôle de deux LEDs

Si tu es arrivée jusque là et que tu as envie d'aller plus loin, tu peux refaire le même montage avec une deuxième LED. Comme tu t'en doutes, on va connecter la deuxième LED à deux autres pins, les pins 37 et 39 (voir Figure 6), qui correspondent à la GPIO 26 et à une autre masse (Ground). Attention aussi à la façon dont tu connectes tes fils à la breadboard !

Fais vérifier ton montage par un encadrant avant de passer à la suite.

Ensuite, il va falloir modifier ton script Python pour contrôler les deux LEDs. Aide toi de ce que tu as fait avant. Tu sais maintenant que la pin n'est plus la numéro 16 mais la 26, tu dois donc changer le numéro quand tu te connectes à la LED. Attention aussi au nom que tu lui donnes, si tu nommes les deux avec le même nom, l'ordinateur ne saura pas de qui tu parles !

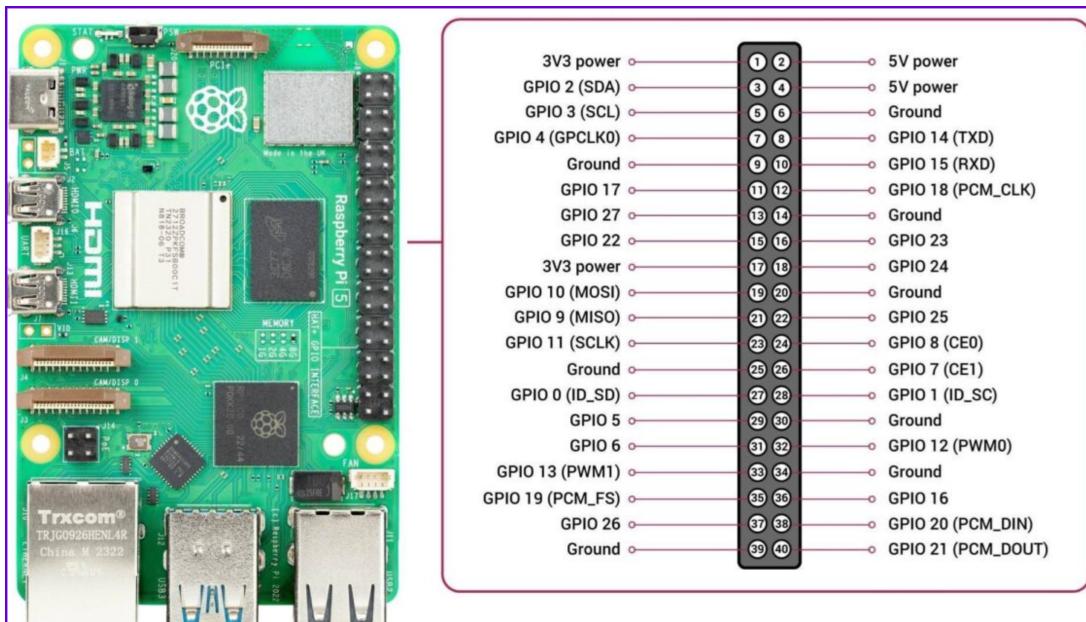


Figure 6: Liste des GPIOs

#### 2.4.6 Eteindre le RPi 5

On peut éteindre le RPi 5 comme un ordinateur classique, mais on peut aussi l'éteindre avec une commande dans le terminal:

```
sudo shutdown -h now
```

Il faut taper le mot de passe: rien ne s'affiche quand tu tapes mais c'est normal ! Tape le bon mot de passe et le RPi 5 va s'éteindre.

Bravo !

### 3 Deuxième Séance

#### 3.1 Montage électronique

On va connecter le ruban LED au raspberry Pi, en utilisant une carte de prototypage, en suivant le schéma ci-dessous:

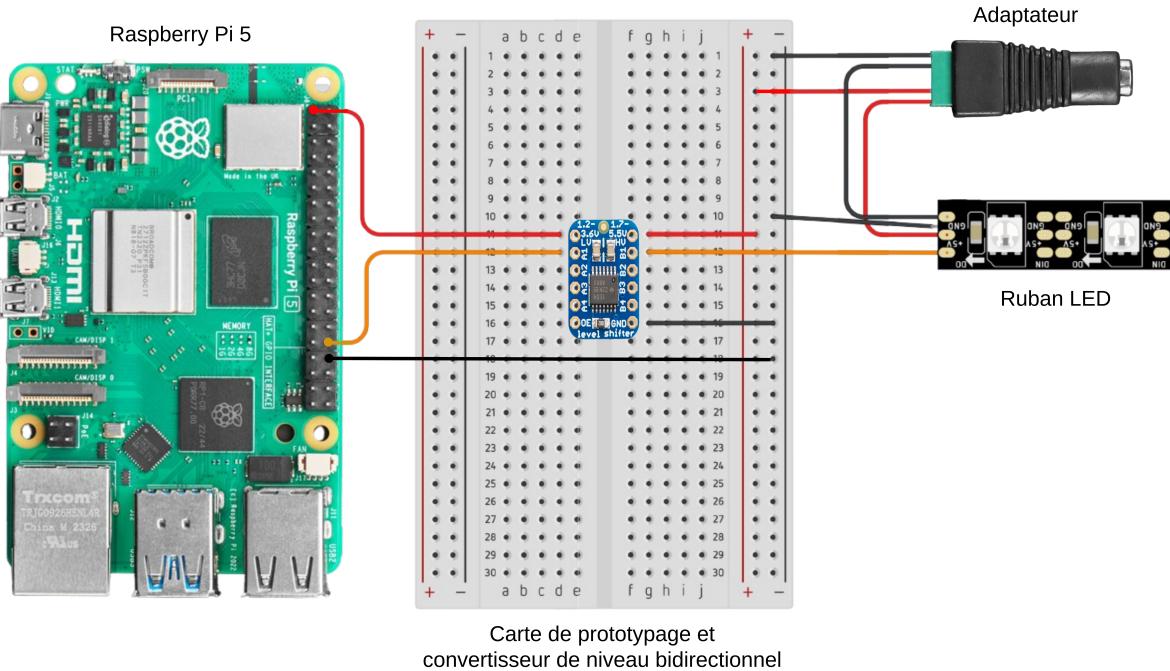


Figure 7: Schéma de connexion du ruban LED au Raspberry Pi

#### 3.2 Programmation Python

##### 3.2.1 Démarrage

Une fois le montage électronique réalisé, on peut brancher le clavier, la souris, et le câble micro-hdmi au Raspberry Pi. Puis, on peut brancher le câble d'alimentation pour allumer le Raspberry Pi 5.

Il va s'allumer, ou "booter", et afficher l'écran de sélection de l'utilisateur. On va sélectionner:

- Utilisateur : **illumine**
- Mot de passe : **illumine**

Après quelques secondes, le bureau du Raspberry Pi 5 va s'afficher. On va ouvrir le terminal (icône noire avec un > blanc) pour pouvoir taper des commandes, comme montré sur la figure 8.

##### 3.2.2 Commandes python

On va se placer dans un environnement python interactif en tapant les commandes suivantes dans le terminal:

```
cd rpi5_projects/git_repos/illumine/python/2eme_seance/
source_illumine_env
ipython
```

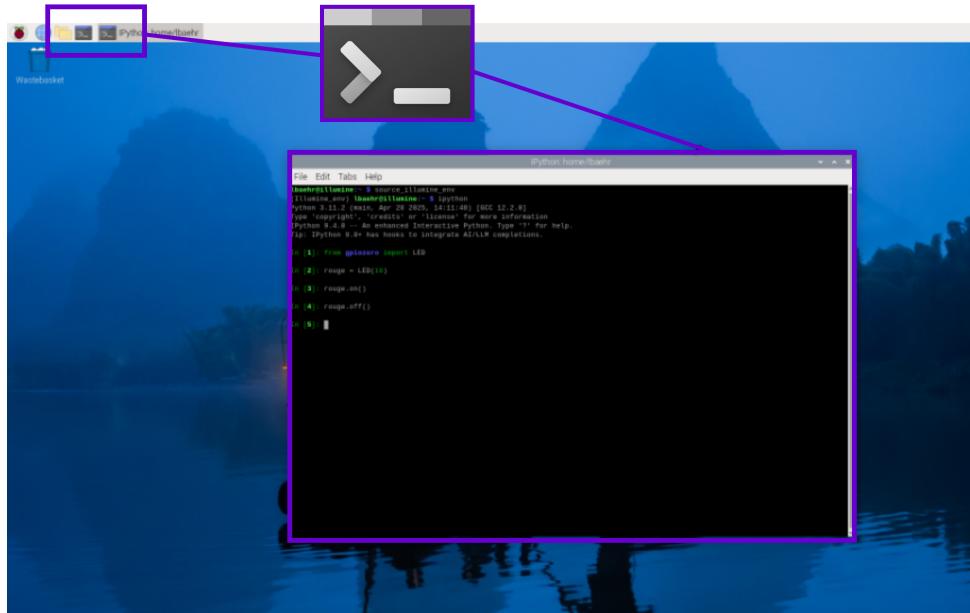


Figure 8: Bureau du Raspberry Pi

```

lbaehr@lbaehr-Zephyrus:~/work/git_repos/illumine_ton_ingeniosite$ ip
Python 3.12.3 (main, Aug 14 2025, 17:47:21) [GCC 13.3.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.4.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: Use `object?` to see the help on `object`, `object??` to view its source
Ctrl click to launch VS Code Native REPL
In [1]: from gpionano import LED

```

Figure 9: Environnement Python interactif

On est maintenant dans un environnement Python interactif qui ressemble à celui sur la figure 9. On peut donc taper des commandes python pour contrôler le ruban LED:

```

import RPi5_Neopixel as np
ruban = np.Neopixel_Tape()
ruban.on_all_led('orchid')

```

On remarque qu'il faut donner un nom de couleur en anglais, alors voici la liste des couleurs disponibles:

'red'	'dark_orange'	'green'
'dark_red'	'golden_orange'	'dark_green'
'red_orange'	'yellow_orange'	'forest_green'
'crimson'	'yellow'	'teal_green'
'rust'	'bright_yellow'	'teal'
'blood_red'	'lemon_yellow'	'bright_turquoise'
'tomato'	'pale_yellow'	'medium_turquoise'
'reddish_orange'	'lime_yellow'	'dark_turquoise'
'burnt_orange'	'chartreuse'	'cyan'
'pumpkin'	'spring_green'	'light_cyan'
'orange'	'light_green'	'aqua'

'light_sky_blue'	'indigo'	'fuchsia'
'cornflower_blue'	'blue_violet'	'hot_pink'
'steel_blue'	'purple'	'light_pink'
'dodger_blue'	'medium_purple'	'pink'
'light_blue'	'orchid'	'light_coral'
'powder_blue'	'violet'	'salmon_pink'
'cadet_blue'	'thistle_purple'	'coral'
'lavender_blue'	'pink_purple'	'tomato_pink'
'light_indigo'	'magenta'	'rosy_brown'

N'oublie pas les guillements autour du nom de la couleur !

### 3.2.3 Fonctions élémentaires

Plusieurs fonctions élémentaires sont disponibles pour piloter une ou plusieurs LED du ruban.

*Exercice:* Essaye les fonctions suivantes en remplaçant **num** par un numéro de LED entre 0 et 59, **color\_name** par un nom de couleur de la liste précédente, et **times** par le nombre de fois que tu veux faire l'effet. Décris sur les pointillés ce que fait chaque fonction.

```
ruban.on_led(num, color_name): .....
ruban.off_led(num): .....
ruban.on_all_led(color_name): .....
ruban.off_all_led(): .....
ruban.avance(num, color_name, times): .....
ruban.recule(num, color_name, times): .....
ruban.clignote(num, color_name, times): .....
ruban.clignote_all(color_name, times): .....
ruban.random(color_name, times): .....
ruban.fill_rainbow(): .....
```

### 3.2.4 Script python

Nous allons maintenant écrire des fonctions plus complexes, mais pour cela nous devons rapidement expliquer ce que fait une boucle *while*. Une boucle *while* permet de répéter une série d'instructions tant qu'une condition est vraie. Une analogie peut être "tant que je ne suis pas arrivée au bout du chemin, je continue d'avancer". Voici un exemple de boucle *while* en python (à ne pas taper, c'est juste un exemple):

```
compteur = 0
while compteur < 5:
    compteur = compteur + 1
```

On va maintenant quitter l'environnement python interactif en tapant la commande `exit()`.

*Exercice:* Tape la commande suivante pour ouvrir et compléter le script python `charge.py`:

```
gedit charge.py
```

Une fenêtre comme celle de la figure 10 va s'ouvrir. La description de la fonction est donnée, et c'est à toi de compléter le script en utilisant les fonctions élémentaires vues précédemment, ainsi qu'une ou plusieurs boucles *while* si nécessaire (et/ou *for*).

```
import RPi5_Neopixel as np

ruban = np.Neopixel_Tape()

"""Remplit et vide la bande un certain nombre de fois avec le nom de couleur donné

    color_name: nom de la couleur à utiliser

"""

num_pixels = 60
ruban.on_led(0, ) # n'oublie pas le nom de la couleur que tu veux utiliser
```

Description de ce que doit faire la fonction

Zone où continuer le script

Figure 10: Fonction `charge.py` à compléter

Une fois terminé, sauvegarde le fichier (Ctrl+S), et tu peux soit fermer la fenêtre et revenir au terminal, soit garder la fenêtre ouverte, ouvrir un autre terminal et retaper:

```
cd rpi5_projects/git_repos/illumine_ton_ingeniosite/2nd_seance/
source_illumine_env
```

Dans les deux cas, tape:

```
python3 charge.py
```

pour exécuter ton script python et voir si le résultat est celui attendu !

Refait cela en remplaçant la fonction `charge.py` par les fonctions `chariot.py` et `croise.py`.

### 3.2.5 Eteindre le RPi 5

On peut éteindre le RPi 5 comme un ordinateur classique, mais on peut aussi l'éteindre avec une commande dans le terminal:

```
sudo shutdown -h now
```

Il faut taper le mot de passe: rien ne s'affiche quand tu tapes mais c'est normal ! Tape le bon mot de passe et le RPi 5 va s'éteindre.

Bravo !

## 4 Troisième Partie

A venir...

## 5 Annexe à destination des encadrants

### 5.1 Matériel

Voici la liste du matériel utilisé pour ce TP. Des produits similaires peuvent être utilisés, vous pouvez vous référer à la section 5.3 pour comprendre les spécificités du matériel.

- **Raspberry Pi 5 4 Go BCM2712 2,4 GHz** <https://fr.rs-online.com/web/p/raspberry-pi/0219253>
- **Carte mémoire A2 class micro-SD cards for DDR50 and SDR104 32GB** <https://mou.sr/3Km0bGc>
- **Alimentation pour Raspberry Pi** 1.2m USB type C avec Fiche femelle européenne <https://fr.rs-online.com/web/p/alimentations-raspberry-pi/0219261>
- **Adafruit NeoPixel Digital RGBW LED Strip - White PCB** 60 LED/m 1m <https://mou.sr/40tlNWa>
- **Female DC Power Adapter** <https://mou.sr/3NrRk6Y>
- **Adaptateur AC/DC 5V c.c., 4A, 24W, C14** <https://fr.rs-online.com/web/p/adaptateurs-ac-dc/9048474>
- **Cordon d'alimentation RS PRO** Connecteur CEI C13 CEE 7/7, 1m, 10 A / 250 V <https://fr.rs-online.com/web/p/cordons-d-alimentation/6266751>
- **TXB0104 Bi-Direction Level Shifter** <https://mou.sr/4heobXq>
- **HDMI vers micro HDMI 1 m** <https://fr.rs-online.com/web/p/cables-raspberry-pi/2012171>
- **Platine d'essai** <https://fr.rs-online.com/web/p/platinas-d-essais/2153175>
- **Strap pour platine d'essai mâle-mâle** <https://fr.rs-online.com/web/p/straps-pour-platinas-d-essais/2048241>
- **Strap pour platine d'essai mâle-femelle** <https://fr.rs-online.com/web/p/straps-pour-platinas-d-essais/2048243>

Si vous souhaitez réaliser la première séance avec une LED unique, il en faudra également une, ainsi qu'une résistance.

**Note:** Si vous utilisez le convertisseur de niveau (Level-Shifter) proposé ici, il faudra souder les broches fournies avec vous-même.

### 5.2 Configuration du Raspberry Pi

Dans un premier temps, il faut installer Raspberry Pi OS sur la carte mémoire et configurer le Raspberry Pi 5. Pour simplifier cette étape, vous trouverez dans ce Proton Drive une image de la carte mémoire déjà configurée:

L'environnement virtuel python `illumine_env` est déjà créé et contient toutes les librairies nécessaires, mais voici la liste de celles assez spécifiques si vous souhaitez le refaire vous-même:

- `gpiozero`

- Adafruit-Blinka-Raspberry-Pi5-Neopixel
- adafruit-circuitpython-pixelbuf
- adafruit-circuitpython-led-animation
- ipython

En plus, vous aurez besoin de librairies plus classiques comme `numpy`, `time`, etc.

**Note:** Si vous modifiez les chemins des dossiers, n'oubliez pas de modifier le TP en conséquence (commande `cd` ).

### 5.3 Références et ressources

Toutes les explications claires et détaillées sont disponibles sur le merveilleux site d'Adafruit.

- Une présentation des rubans LED NeoPixel: <https://learn.adafruit.com/neopixels-on-raspberry-pi>
- Comment réaliser le montage électronique: <https://learn.adafruit.com/neopixels-on-raspberry-pi/raspberry-pi-wiring>
- Les librairies python à installer et un exemple de code: <https://learn.adafruit.com/neopixels-on-raspberry-pi/python-usage>
- Les modifications nécessaires pour le Raspberry Pi 5: <https://learn.adafruit.com/circuitpython-on-raspberry-pi-using-neopixels-on-the-pi-5>
- Présentation du Level Shifter TXB0104: <https://www.adafruit.com/product/1875>

Si vous souhaitez réaliser la première séance avec une LED unique, voici des ressources utiles:

- Le circuit électronique simple <https://raspberry-pi.fr/led-raspberry-pi/>
- La librairie gpiodzero <https://raspberrypi.stackexchange.com/questions/148686/raspberry-pi-5-gpio-zero>