

### **Comment installer une Os sur la raspberry:**

- mettre carte sd dans un pc
- chercher sur google raspberry os download
- choisir la version de la raspberry
- et le reste est intuitif

### **Comment trouver l'adresse IP de la raspberry pi:**

- chercher l'adresse ip du site du routeur
  - mot de passe par défaut admin (pas de nom d'utilisateur)
  - dans la barre aller sur état
  - tables des clients DHCP
- ou
- faire ip -c a pour s'assurer d'être connecter au réseau local
  - faire nmap -sN <adresse du réseau locale>

### **Comment se connecter à la Raspberry pi en ssh:**

- faire "sudo raspi-config" sur le terminale de la raspberry pi
- aller dans interface options
- aller dans ssh et cliquer sur "yes"

Puis sur un pc distant:

- taper la commande "ssh pi@<Adresse IP de la raspberry>"
- taper "yes"

### **Comment installer python sur la raspberry:**

- taper "sudo apt- get install python"
- taper "sudo apt- get install python3"

### **Comment faire allumer une LED avec la raspberry pi:** (code

source:<https://chat.openai.com/share/0c597419-2972-441f-8f65-d125ba52ab9b>)

```
GNU nano 7.2
ledras.py
import RPi.GPIO as GPIO
import time

# Définir le mode de numérotation des broches
GPIO.setmode(GPIO.BCM)

# Définir le numéro de la broche à utiliser
led_pin = 20
```

```

# Configurer la broche comme une sortie
GPIO.setup(led_pin, GPIO.OUT)

try:
    while True:
        # Allumer la LED
        GPIO.output(led_pin, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)  # Attendre 1 seconde

        # Éteindre la LED
        GPIO.output(led_pin, GPIO.LOW)
        print("LED éteinte")
        time.sleep(1)  # Attendre 1 seconde

except KeyboardInterrupt:
    # Arrêter le script proprement lorsque Ctrl+C est pressé
    GPIO.cleanup()

```

### **Comment faire allumer plusieurs LEDs avec la raspberry pi:**

```

import RPi.GPIO as GPIO
import time

# Définir le mode de numérotation des broches
GPIO.setmode(GPIO.BCM)

# Définir le numéro de la broche à utiliser
led_pin3 = 20
led_pin2 = 21
led_pin1 = 16
# Configurer la broche comme une sortie
GPIO.setup(led_pin1, GPIO.OUT)
GPIO.setup(led_pin2, GPIO.OUT)
GPIO.setup(led_pin3, GPIO.OUT)
try:
    while True:
        # Allumer les LEDs 1 à 1
        GPIO.output(led_pin1, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)
        GPIO.output(led_pin1, GPIO.LOW)
        print("LED éteint")
        GPIO.output(led_pin2, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)
        GPIO.output(led_pin2, GPIO.LOW)
        print("LED éteint")

```

```

        GPIO.output(led_pin3, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1)
        GPIO.output(led_pin3, GPIO.LOW)
        print("LED éteint")

except KeyboardInterrupt:
    # Arrêter le script proprement lorsque Ctrl+C est pressé
    GPIO.cleanup()

```

si le lien du code source y marche pas :

la commande suivante dans le terminal :

bash

Copy code

```
pip install RPi.GPIO
```

Ensuite, vous pouvez créer un script Python pour contrôler la LED. Voici un exemple de programme qui allume et éteint la LED à l'aide du GPIO 17 (vous pouvez ajuster le numéro de broche en fonction de votre configuration) :

python

Copy code

```

import RPi.GPIO as GPIO
import time

# Définir le mode de numérotation des broches
GPIO.setmode(GPIO.BCM)

# Définir le numéro de la broche à utiliser
led_pin = 17

# Configurer la broche comme une sortie
GPIO.setup(led_pin, GPIO.OUT)

try:
    while True:
        # Allumer la LED
        GPIO.output(led_pin, GPIO.HIGH)
        print("LED allumée")
        time.sleep(1) # Attendre 1 seconde


        # Éteindre la LED
        GPIO.output(led_pin, GPIO.LOW)
        print("LED éteinte")
        time.sleep(1) # Attendre 1 seconde

except KeyboardInterrupt:
    # Arrêter le script proprement lorsque Ctrl+C est pressé
    GPIO.cleanup()

```

Sauvegardez ce script avec une extension ".py" (par exemple, "led\_control.py") et exécutez-le sur votre Raspberry Pi avec la commande suivante dans le terminal :

bash

 Copy code

```
python led_control.py
```

Assurez-vous que le script a les permissions appropriées pour être exécuté. Vous pouvez utiliser la commande `chmod +x led_control.py` pour cela.

Ce script fait alterner l'état de la LED entre allumé et éteint toutes les secondes. Vous pouvez ajuster la durée du sommeil (`time.sleep()`) selon vos préférences.