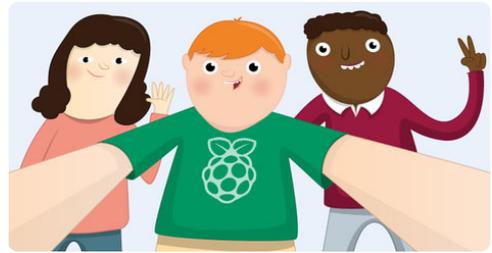


## Premiers pas avec le module caméra

Capturer des images et des vidéos avec le module caméra du Raspberry Pi et Python



### Étape 1 Introduction

---

Apprends à connecter le module caméra Raspberry Pi à ton Raspberry Pi et à capturer des images, enregistrer des vidéos et appliquer des effets d'image.



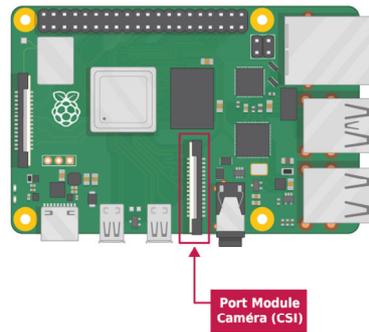
# Étape

## 2 Ce dont tu auras besoin

---

### Un ordinateur Raspberry Pi avec un port module caméra

Tous les modèles actuels du Raspberry Pi ont un port pour connecter le module caméra.



**Remarque :** Si tu veux utiliser un Raspberry Pi Zero, tu as besoin d'une nappe caméra qui s'adapte au plus petit port du module caméra du Raspberry Pi Zero.

### Module caméra du Raspberry Pi



Il existe deux versions du module caméra :

- **La version standard** (<https://www.raspberrypi.org/products/camera-module-v2/>), conçue pour prendre des photos en lumière normale
- **La version NoIR** (<https://www.raspberrypi.org/products/pi-noir-camera-v2/>), qui n'a pas de filtre infrarouge, tu peux donc l'utiliser avec une source de lumière infrarouge pour prendre des photos dans l'obscurité

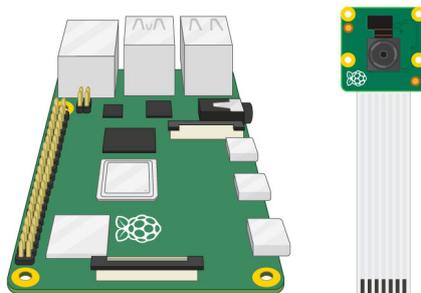
## Étape

### 3 Connecter le module caméra

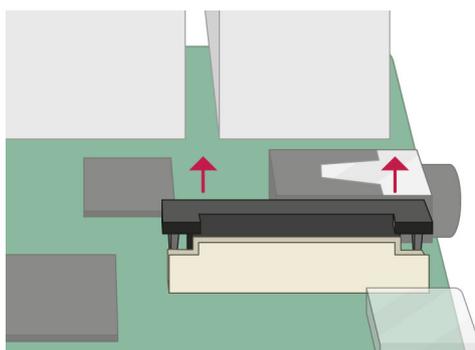
---

Assure-toi que ton Raspberry Pi est éteint.

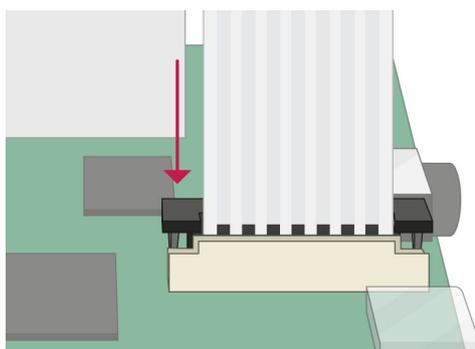
1. Trouve le port du module caméra.



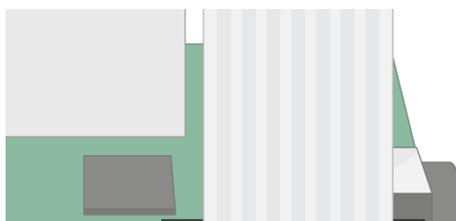
1. Tire doucement sur les bords du clip en plastique du port.

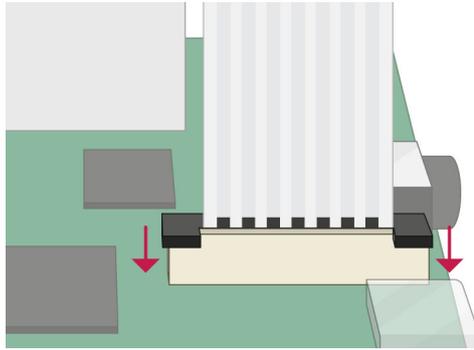
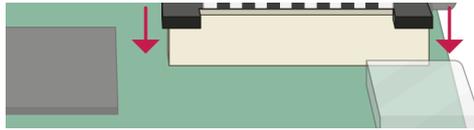


1. Insère la nappe caméra ; assure-toi que la nappe est dans le bon sens.

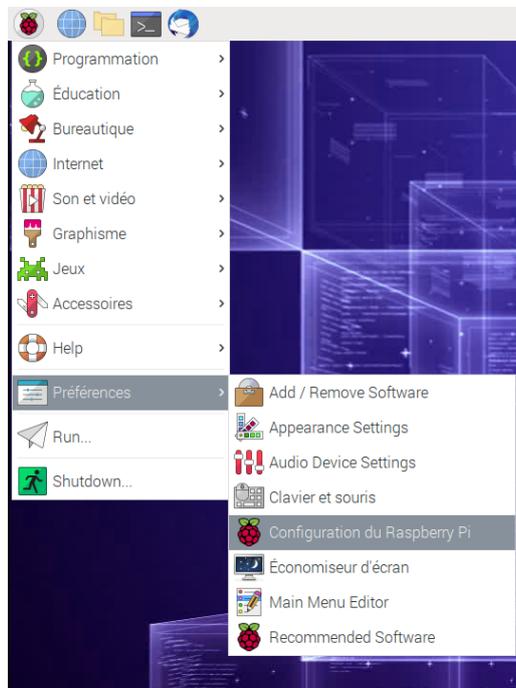


1. Remets le clip en plastique en place.





- Démarre ton Raspberry Pi.
- Va dans le menu principal et ouvre l'outil **Configuration du Raspberry Pi**.



- Sélectionne l'onglet **Interfaces** et assure-toi que la caméra est **activée** :



- Redémarre ton Raspberry Pi.

## Étape

### 4 Comment contrôler le module caméra via la ligne de commande

---

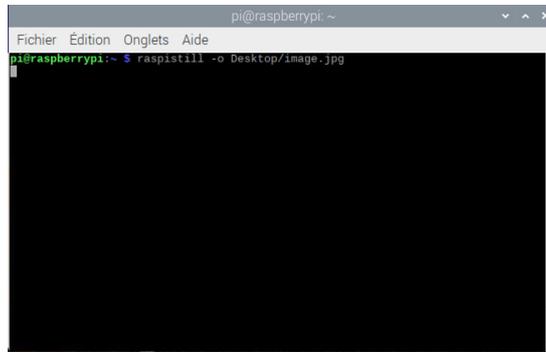
Maintenant que ton module caméra est connecté et que le logiciel est activé, essaie les outils de ligne de commande `raspistill` et `raspivid`.

- Ouvre une fenêtre de terminal en cliquant sur l'icône du moniteur noir dans la barre des tâches :



- Saisis la commande suivante pour capturer une image fixe et l'enregistrer sur le bureau :

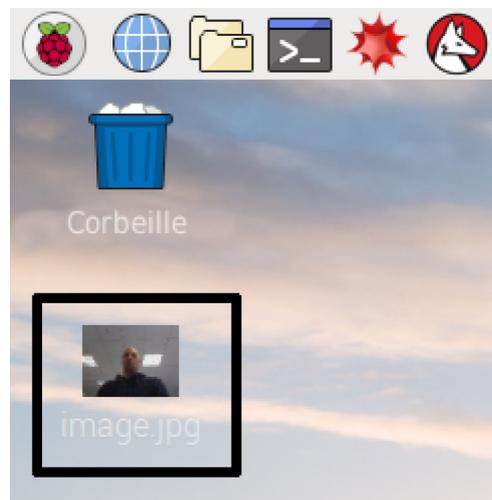
```
raspistill -o Desktop/image.jpg
```



- Appuie sur Entrée pour exécuter la commande.

Lorsque la commande s'exécute, tu peux voir l'aperçu de la caméra s'ouvrir pendant cinq secondes avant qu'une image fixe ne soit capturée.

- Recherche l'icône du fichier image sur le bureau et double-clique sur l'icône du fichier pour ouvrir l'image.



En ajoutant différentes options, tu peux définir la taille et l'apparence de l'image que prend la commande `raspistill`.

- Par exemple, ajoute `-h` et `-w` pour modifier la hauteur et la largeur de l'image :

```
raspistill -o Desktop/image-small.jpg -w 640 -h 480
```

- Enregistre maintenant une vidéo avec le module caméra en utilisant la commande `raspivid` :

```
raspivid -o Desktop/video.h264
```

- Pour lire le fichier vidéo, double-clique sur l'icône du fichier `video.h264` sur le bureau pour l'ouvrir dans VLC Media Player.

Pour plus d'informations et d'autres options que tu peux utiliser avec ces commandes, lis la documentation pour **raspistill**

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md> et la

documentation pour **raspivid**

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspivid.md>.

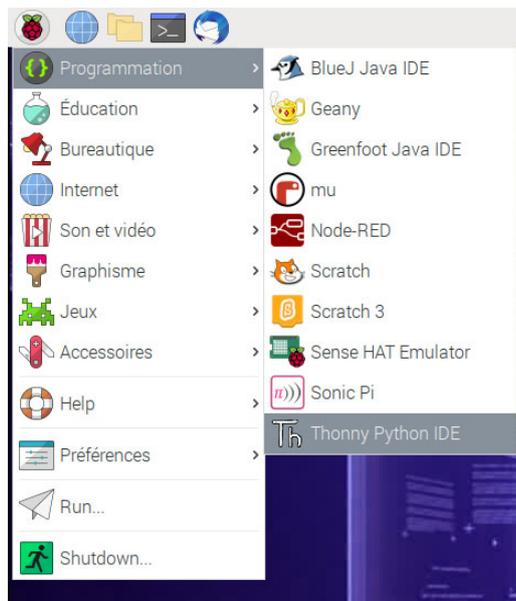
## Étape

### 5 Comment contrôler le module caméra avec du code Python

---

La bibliothèque Python `pi camera` te permet de contrôler ton module caméra et de créer des projets incroyables.

- Ouvre un éditeur Python 3, tel que **Thonny Python IDE** :



- Ouvre un nouveau fichier et enregistre-le sous `camera.py`.

**Remarque** : il est important que tu **ne sauvegardes jamais le fichier en tant que** `pi camera.py`.

- Entre le code suivant :

```
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
sleep(5)
camera.stop_preview()
```

- Enregistre et teste ton programme. L'aperçu de la caméra devrait s'afficher pendant cinq secondes, puis se refermer.





**Remarque :** l'aperçu de la caméra ne fonctionne que lorsqu'un moniteur est connecté à ton Raspberry Pi. Si tu utilises un accès à distance (tel que SSH ou VNC), tu ne verras pas l'aperçu de la caméra.

- Si ton aperçu est à l'envers, tu peux le faire pivoter de 180 degrés avec le code suivant :

```
camera = PiCamera()  
camera.rotation = 180
```

Tu peux faire pivoter l'image de **90**, **180** ou **270** degrés. Pour réinitialiser l'image, définis la **rotation** sur **0** degrés.

Il est préférable de rendre l'aperçu légèrement transparent afin de voir si des erreurs se produisent dans ton programme lorsque l'aperçu est activé.

- Rendre l'aperçu de la caméra transparent en définissant un niveau **alpha** :

```
camera.start_preview(alpha=200)
```

La valeur **alpha** peut être n'importe quel nombre compris entre **0** et **255**.

## Étape

### 6 Capturer des images fixes avec du code Python

---

Utilise maintenant le module caméra et Python pour capturer des images fixes.

- Modifie ton code pour ajouter une ligne `camera.capture()` :

```
camera.start_preview()
sleep(5)
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

**Remarque :** c'est important d'utiliser un `sleep` pendant au moins deux secondes avant de capturer une image, car cela donne au capteur de l'appareil photo le temps de détecter les niveaux de lumière.

- Exécute le code.

Tu devrais voir l'aperçu de la caméra pendant cinq secondes, et ensuite une image fixe devrait être capturée. Pendant la prise de vue, tu peux voir l'aperçu s'ajuster brièvement à une résolution différente.

Ta nouvelle image devrait être enregistrée sur le bureau.

- Maintenant, ajoute une boucle pour prendre cinq images d'affilée :

```
camera.start_preview()
for i in range(5):
    sleep(5)
    camera.capture('/home/pi/Desktop/image%s.jpg' % i)
camera.stop_preview()
```

La variable `i` compte le nombre de fois que la boucle a été exécutée, de 0 à 4. Par conséquent, les images sont enregistrées sous `image0.jpg`, `image1.jpg`, et ainsi de suite.

- Exécute à nouveau le code et maintiens le module caméra en position.

La caméra doit prendre une image toutes les cinq secondes. Une fois la cinquième image prise, l'aperçu se ferme.

- Regarde ton bureau pour trouver les cinq nouvelles images.

## Étape 7 Enregistrer des vidéos avec du code Python

---

Maintenant enregistre une vidéo !

- Modifie ton code pour supprimer `capture()` et ajoute à la place `start_recording()` et `stop_recording()`

Ton code devrait ressembler à ceci :

```
camera.start_preview()
camera.start_recording('/home/pi/Desktop/video.h264')
sleep(5)
camera.stop_recording()
camera.stop_preview()
```

- Exécute le code.

Ton Raspberry Pi devrait ouvrir un aperçu, enregistrer 5 secondes de vidéo, puis fermer l'aperçu.

# Étape

## 8

### Comment modifier les paramètres d'image et ajouter des effets d'image

---

Le logiciel Python `pi camera` fournit un certain nombre d'effets et de configurations pour changer l'apparence de tes images.

**Remarque :** certains paramètres n'affectent que l'aperçu et non l'image capturée, certains n'affectent que l'image capturée, et beaucoup d'autres affectent les deux.

#### Définir la résolution de l'image

Tu peux modifier la **résolution** de l'image prise par le module caméra.

Par défaut, la résolution de l'image est définie sur la résolution de ton moniteur. La résolution maximale est de  $2592 \times 1944$  pour les images fixes et de  $1920 \times 1080$  pour l'enregistrement vidéo.

- Utilise le code suivant pour définir la **résolution** au maximum et capture une image.

**Remarque :** tu dois également définir le taux d'images à 15 pour activer cette résolution maximale.

```
camera.resolution = (2592, 1944)
camera.framerate = 15
camera.start_preview()
sleep(5)
camera.capture('/home/pi/Desktop/max.jpg')
camera.stop_preview()
```

La résolution minimale est de  $64 \times 64$ .

- Essaie de capturer une image avec la résolution minimale.

#### Ajouter du texte à ton image

Tu peux ajouter du texte à ton image à l'aide de la commande `annotate_text`.

- Exécute ce code pour l'essayer :

```
camera.start_preview()
camera.annotate_text = "Bonjour le monde !"
sleep(5)
camera.capture('/home/pi/Desktop/text.jpg')
camera.stop_preview()
```

#### Modifier l'apparence du texte ajouté

- Définis la taille du texte avec le code suivant :

```
camera.annotate_text_size = 50
```

Tu peux définir la taille du texte sur n'importe quelle valeur comprise entre 6 et 160. La taille par

défaut est 32.

Il est également possible de modifier la couleur du texte.

- Tout d'abord, ajoute `Color` à ta ligne `import` en haut du programme :

```
from picamera import PiCamera, Color
```

- Ensuite, sous la ligne `import`, modifie le reste de ton code pour qu'il ressemble à ceci :

```
camera.start_preview()
camera.annotate_background = Color('blue')
camera.annotate_foreground = Color('yellow')
camera.annotate_text = " Bonjour le monde "
sleep(5)
camera.stop_preview()
```

### Modifier la luminosité de l'aperçu

Tu peux modifier la luminosité de l'aperçu. La luminosité par défaut est 50 et tu peux la régler sur n'importe quelle valeur comprise entre 0 et 100.

- Exécute le code suivant pour essayer ceci :

```
camera.start_preview()
camera.brightness = 70
sleep(5)
camera.capture('/home/pi/Desktop/bright.jpg')
camera.stop_preview()
```

- La boucle suivante ajuste la luminosité et ajoute également du texte pour afficher le niveau de luminosité actuelle :

```
camera.start_preview()
for i in range(100):
    camera.annotate_text = "Brightness: %s" % i
    camera.brightness = i
    sleep(0.1)
camera.stop_preview()
```

### Modifier le contraste de l'aperçu

De la même manière que la luminosité de l'aperçu, tu peux modifier le contraste de l'aperçu.

- Exécute le code suivant pour essayer ceci :

```
camera.start_preview()
for i in range(100):
    camera.annotate_text = "Contrast: %s" % i
    camera.contrast = i
    sleep(0.1)
camera.stop_preview()
```

## Ajouter des effets d'image sympas

Tu peux utiliser `camera.image_effect` pour appliquer un effet d'image particulier.

Les options d'effet d'image sont :

- `none`  
(aucun)
- `negative` (négatif)
- `solarize`  
(solariser)
- `sketch` (dessin)
- `denoise`  
(antibruit)
- `emboss` (relief)
- `oilpaint`  
(peinture à l'huile)
- `hatch`  
(hachurer)
- `gpen`
- `pastel`
- `watercolor`  
(aquarelle)
- `film`
- `blur`  
(flou)
- `saturation`
- `colorswap`  
(permutation de couleur)
- `washedout`  
(délavé)
- `posterise`  
(postériser)
- `colorpoint` (point de couleur)
- `colorbalance` (balance des couleurs)
- `cartoon` (dessin animé)
- `deinterlace1`  
(désentrelacement1)
- `deinterlace2`  
(désentrelacement2)

L'effet par défaut est `none`.

- Choisis un effet d'image et essaie-le :

```
camera.start_preview()
camera.image_effect = 'colorswap'
sleep(5)
camera.capture('/home/pi/Desktop/colorswap.jpg')
camera.stop_preview()
```

- Exécute ce code pour boucler sur **tous** les effets d'image avec `camera.IMAGE_EFFECTS` :

```
camera.start_preview()
for effect in camera.IMAGE_EFFECTS:
    camera.image_effect = effect
    camera.annotate_text = "Effect: %s" % effect
    sleep(5)
camera.stop_preview()
```



### Définir le mode d'exposition de l'image

Tu peux utiliser `camera.exposure_mode` pour définir l'exposition sur un mode particulier.

Les options de mode d'exposition sont :

- off (désactivé)
- auto  
(automatique)
- night  
(nuit)
- nightpreview (vision  
nocturne)
- backlight (éclairage de  
derrière)
- spotlight  
(projecteur)
- sports
- snow  
(neige)
- beach (plage)
- verylong (très  
long)
- fixedfps (image fixe par  
seconde)
- antishake  
(anti-tremblement)

- fireworks (feux d'artifice)

Le mode par défaut est auto.

- Choisis un mode d'exposition et essaie-le :

```
camera.start_preview()
camera.exposure_mode = 'beach'
sleep(5)
camera.capture('/home/pi/Desktop/beach.jpg')
camera.stop_preview()
```

- Tu peux boucler tous les modes d'exposition avec `camera.EXPOSURE_MODES`, comme tu l'as fait pour les effets d'image.

### Modifier la balance des blancs de l'image

Tu peux utiliser `camera.awb_mode` pour régler la balance des blancs automatique sur un mode prédéfini.

Les modes de balance des blancs disponibles sont :

- off (éteint)
- auto  
(automatique)
- sunlight (lumière du soleil)
- cloudy (nuageux)
- shade  
(ombre)
- tungsten  
(tungstène)
- fluorescent
- incandescent
- flash
- horizon

La valeur par défaut est auto.

- Choisis un mode de balance des blancs automatique et essaie-le :

```
camera.start_preview()
camera.awb_mode = 'sunlight'
sleep(5)
camera.capture('/home/pi/Desktop/sunlight.jpg')
camera.stop_preview()
```

- Tu peux boucler tous les modes de balance des blancs automatique avec `camera.AWB_MODES`, comme tu l'as fait pour les effets d'image.

## Étape

### 9 Et ensuite ?

---

Maintenant que tu sais comment utiliser ton module caméra, tu peux par exemple :

- Ajouter des boutons pour contrôler la caméra à l'aide du code Python **GPIO Zero** (<https://gpiozero.readthedocs.org/>)
- Intégrer la caméra avec Minecraft Pi
- Publier automatiquement les images de la caméra sur Twitter

Essaie ces projets de module caméra pour en savoir plus :

- Créer un film à **bouton poussoir stop-motion** (<https://projects.raspberrypi.org/fr-FR/projects/push-button-stop-motion/>)
- Créer un **photomaton Minecraft** (<https://projects.raspberrypi.org/fr-FR/projects/minecraft-photobooth/>)
- Obtenir des **ours Babbage pour tweeter des images** (<https://projects.raspberrypi.org/fr-FR/projects/tweeting-babbage/>)
- Construire un **détecteur parent** (<https://projects.raspberrypi.org/fr-FR/projects/parent-detector/>)
- Utiliser le module caméra NoIR pour créer une **niche à oiseaux infrarouge** (<https://projects.raspberrypi.org/fr-FR/projects/infrared-bird-box/>)

Pour plus d'informations sur l'écriture de code Python pour contrôler le module caméra, consulte la grande documentation **picamera** (<https://picamera.readthedocs.org/>).

---

Ce projet a été traduit par des bénévoles:

Jonathan Vannieuwerkerke

Michel Arnols

Grâce aux bénévoles, nous pouvons donner aux gens du monde entier la chance d'apprendre dans leur propre langue. Vous pouvez nous aider à atteindre plus de personnes en vous portant volontaire pour la traduction - plus d'informations sur **rpf.io/translate** (<https://rpf.io/translate>).

---

Publié par **Raspberry Pi Foundation** (<https://www.raspberrypi.org>) sous un **Creative Commons** license (<https://creativecommons.org/licenses/by-sa/4.0/>).

Voir le projet et la licence sur GitHub (<https://github.com/RaspberryPiLearning/getting-started-with-picamera>)