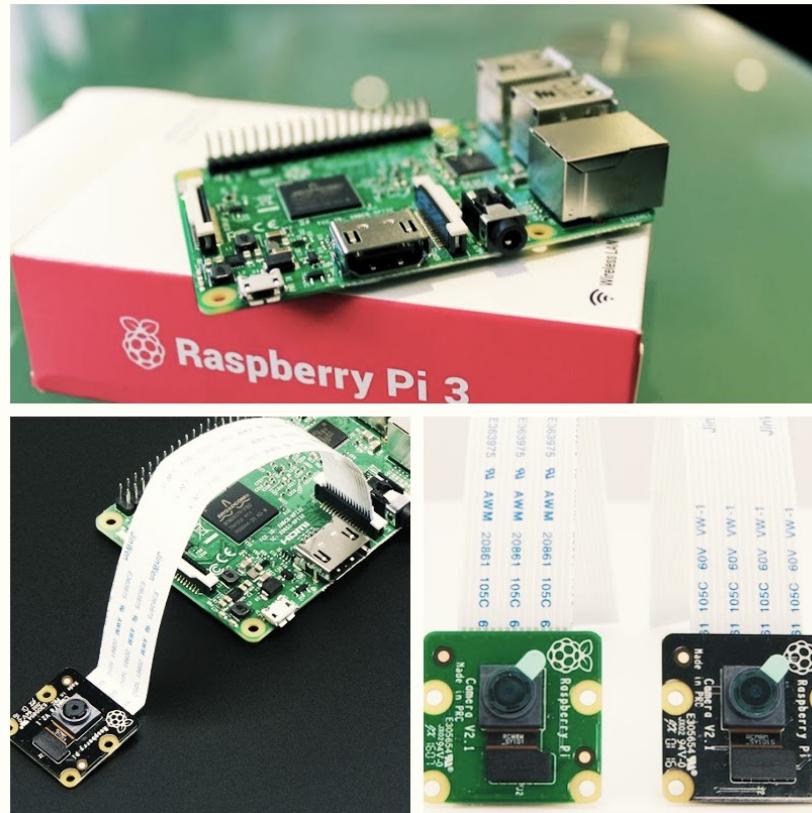


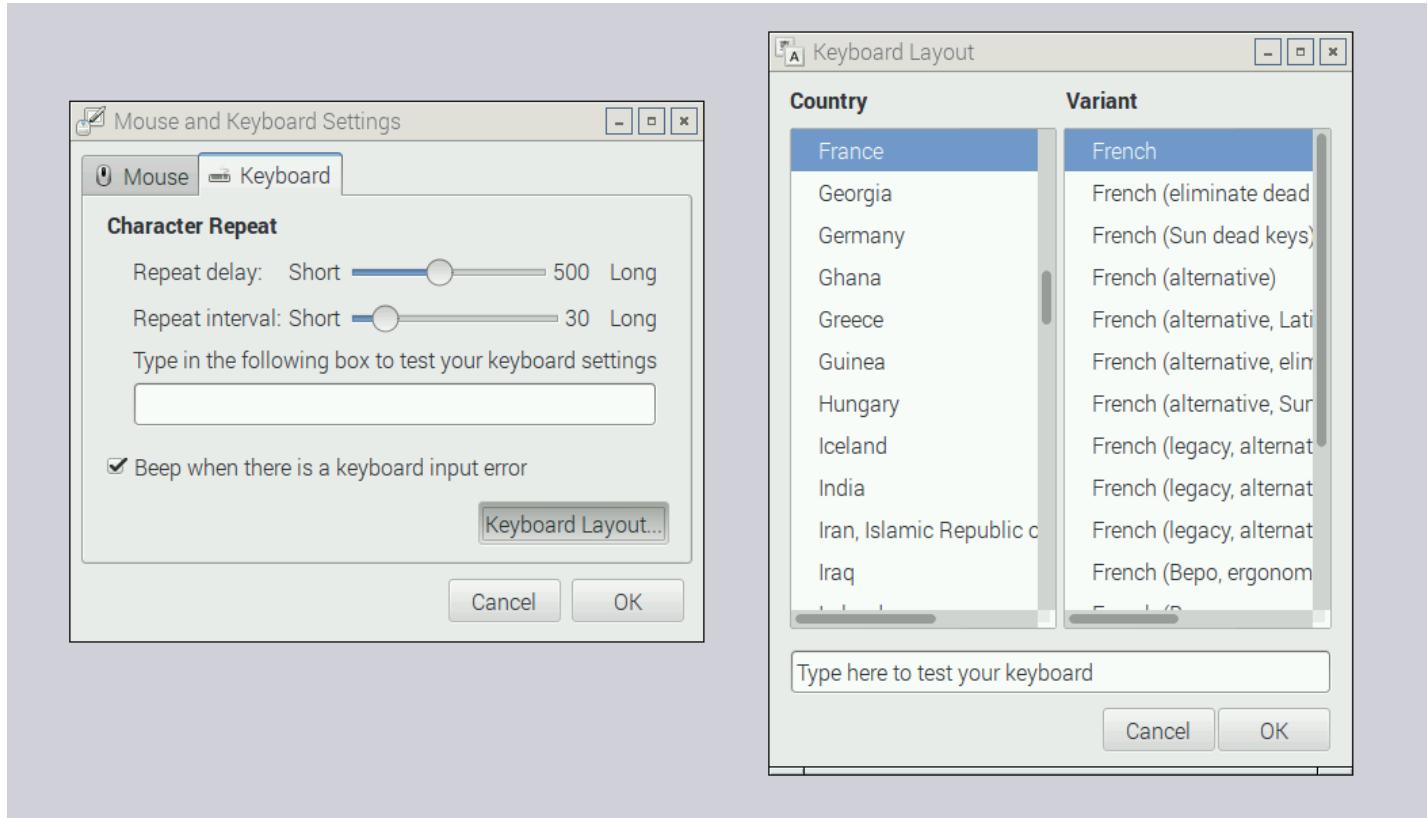
# WAJUG

## Programmer et contrôler votre caméra Raspberry Pi3



# Étape 1 – Préparation

- Dans le menu principal, « Preferences », ouvrir le « Mouse and Keyboard Settings »



Étape 1 – Préparation

Étape 2 – Contrôler la caméra

Étape 3 – Afficher les raccourcis

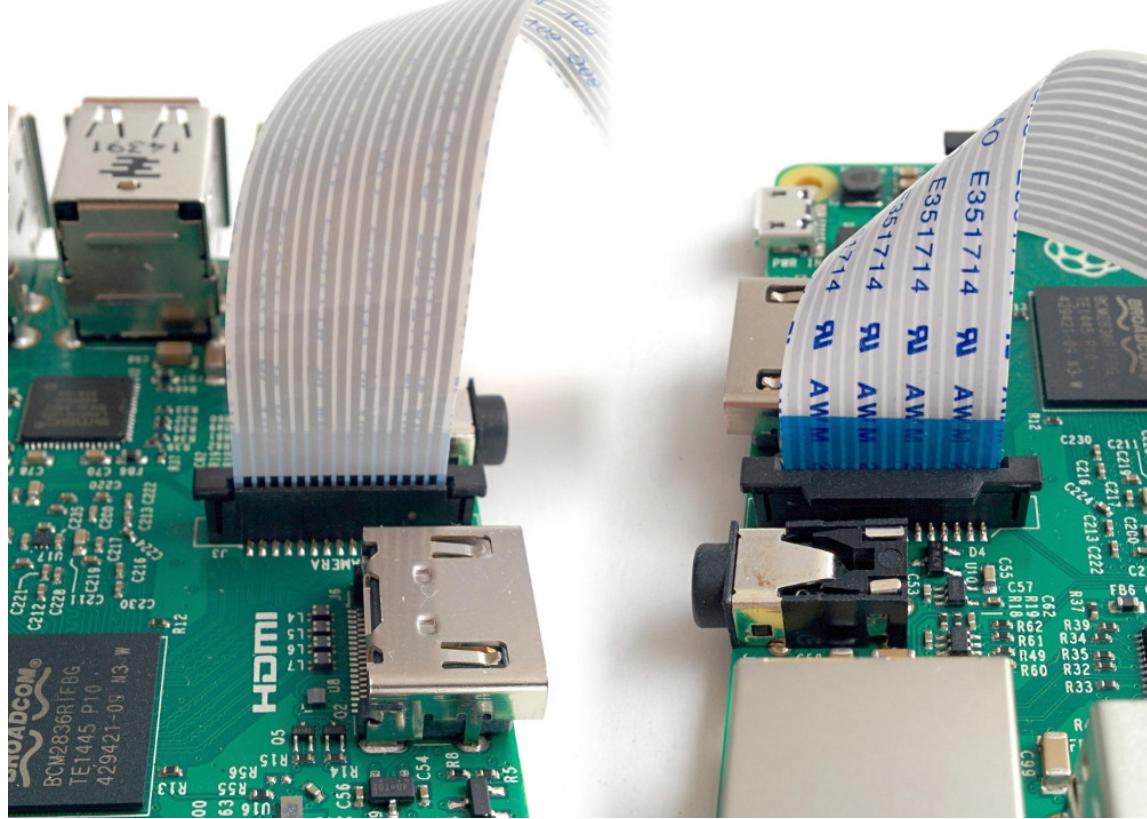
Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

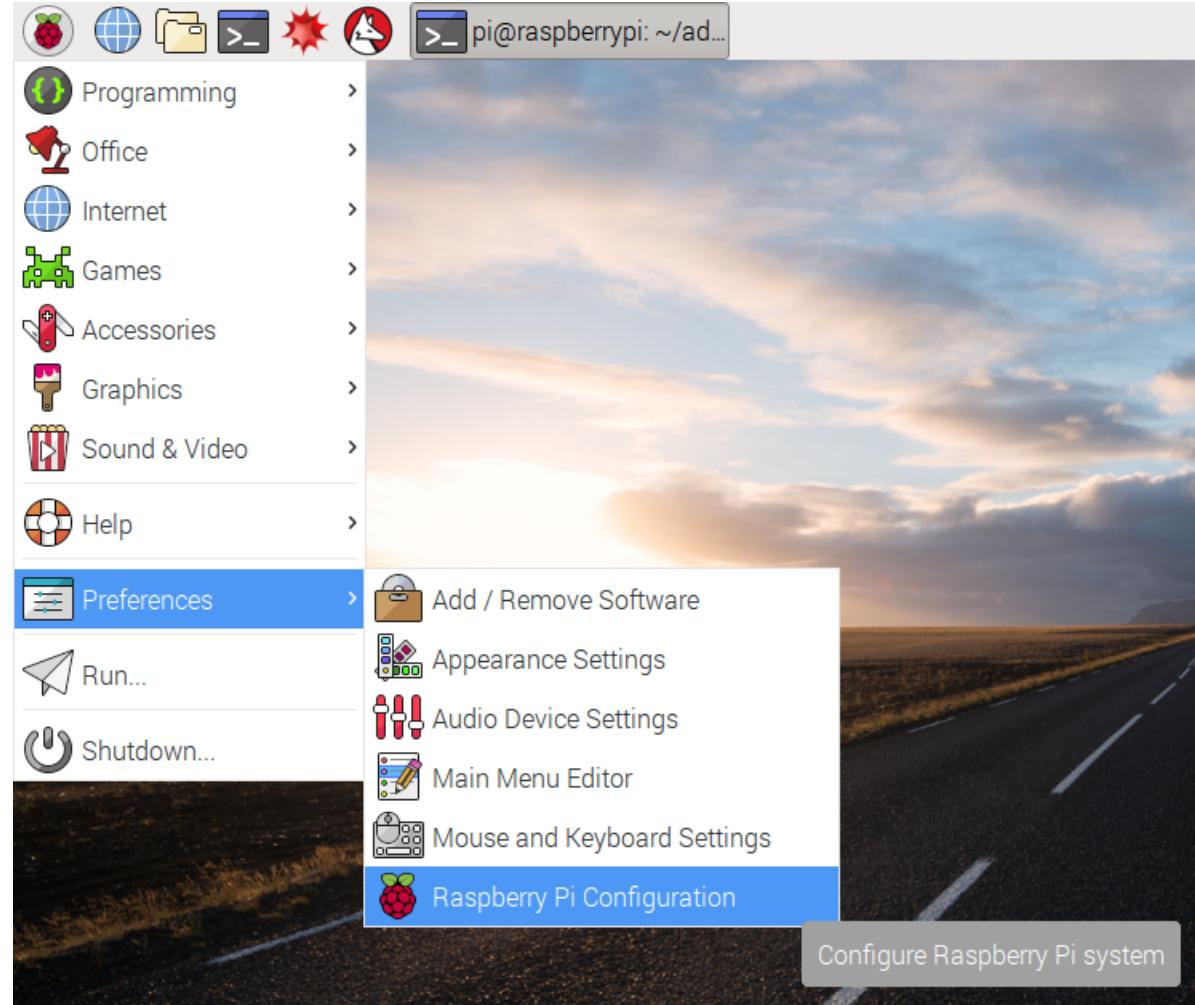
# Étape 1 – Préparation

- Connecter sa caméra



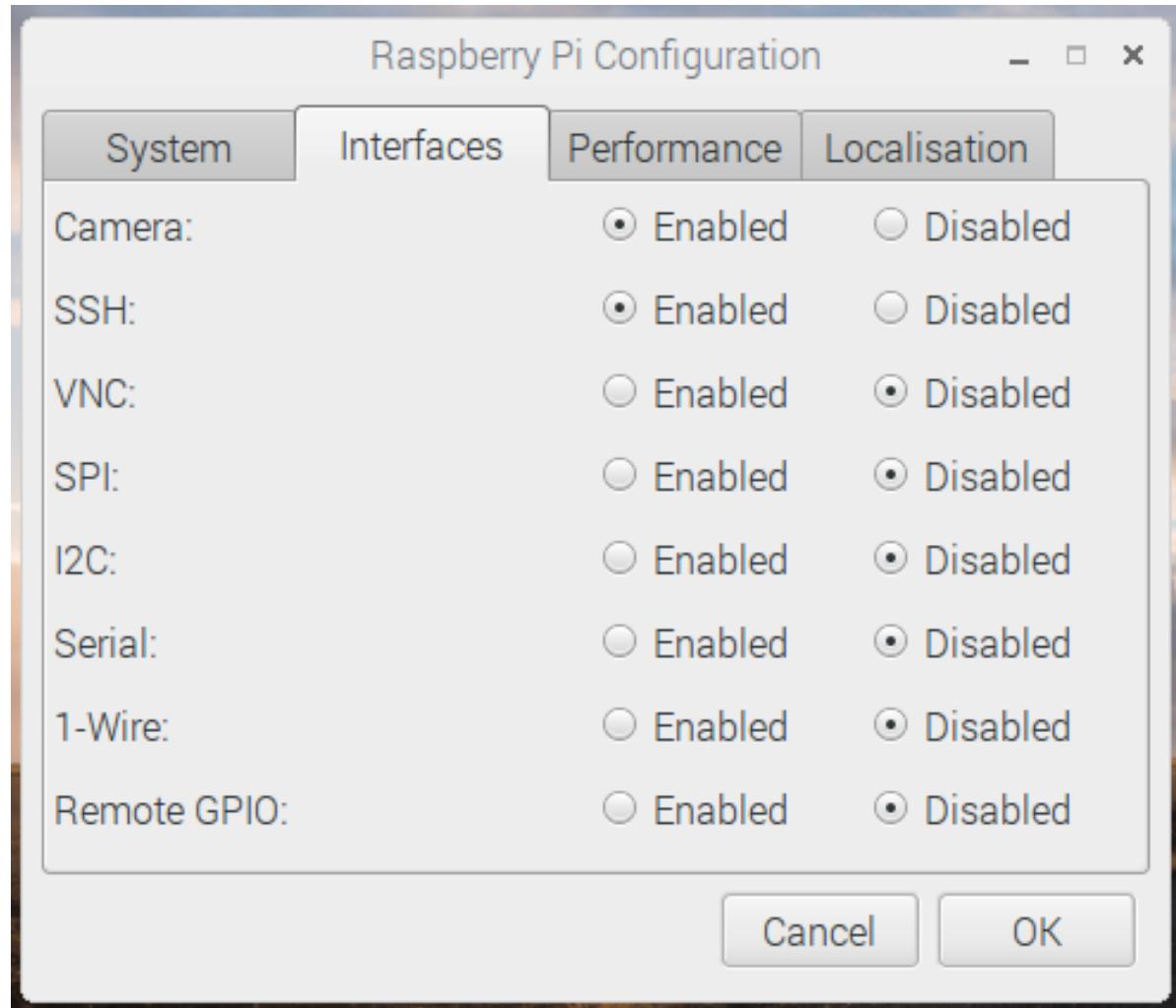
# Étape 1 – Préparation

- Dans le menu principal, « Preferences », ouvrir le « Raspberry Pi Configuration Tool »



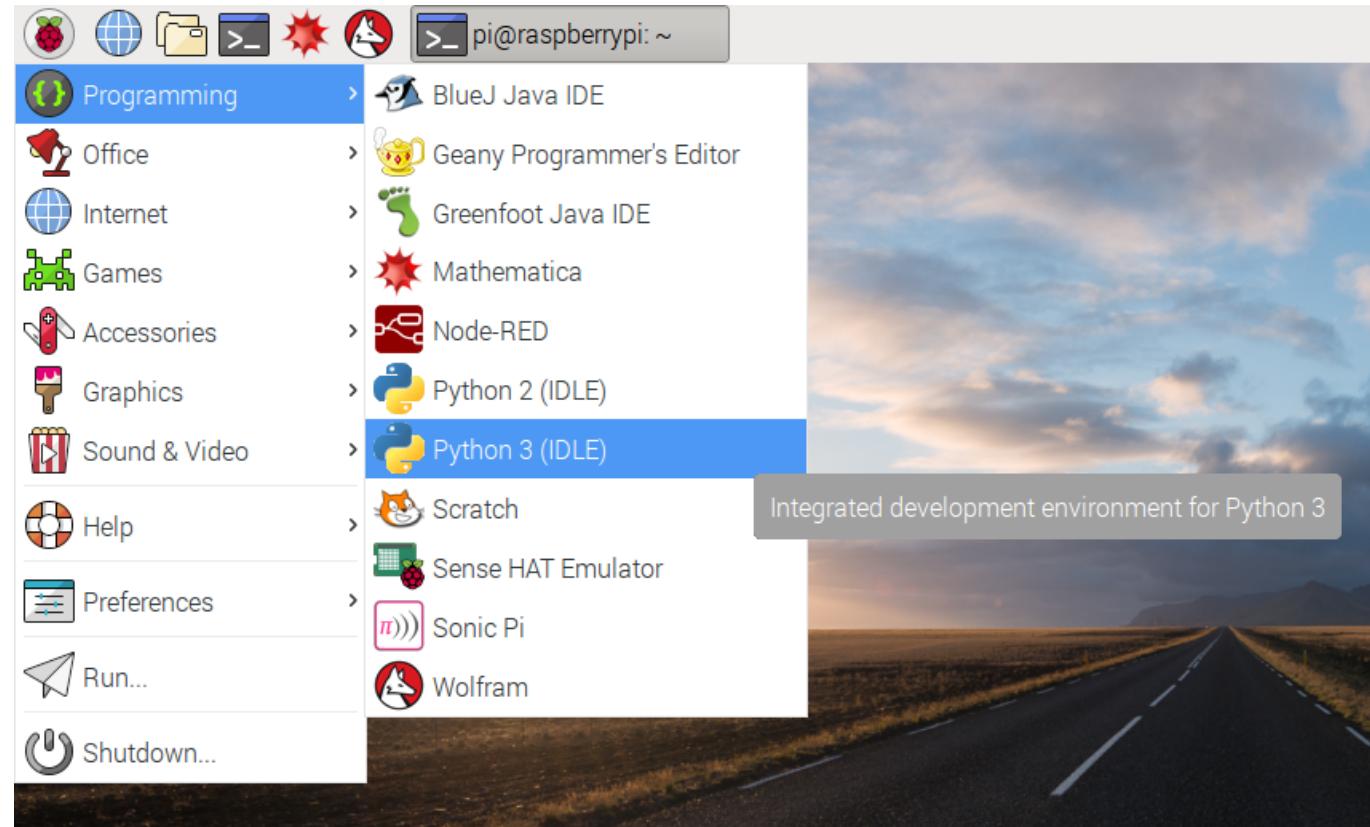
# Étape 1 – Préparation

- Activer la caméra dans l'onglet « Interfaces »



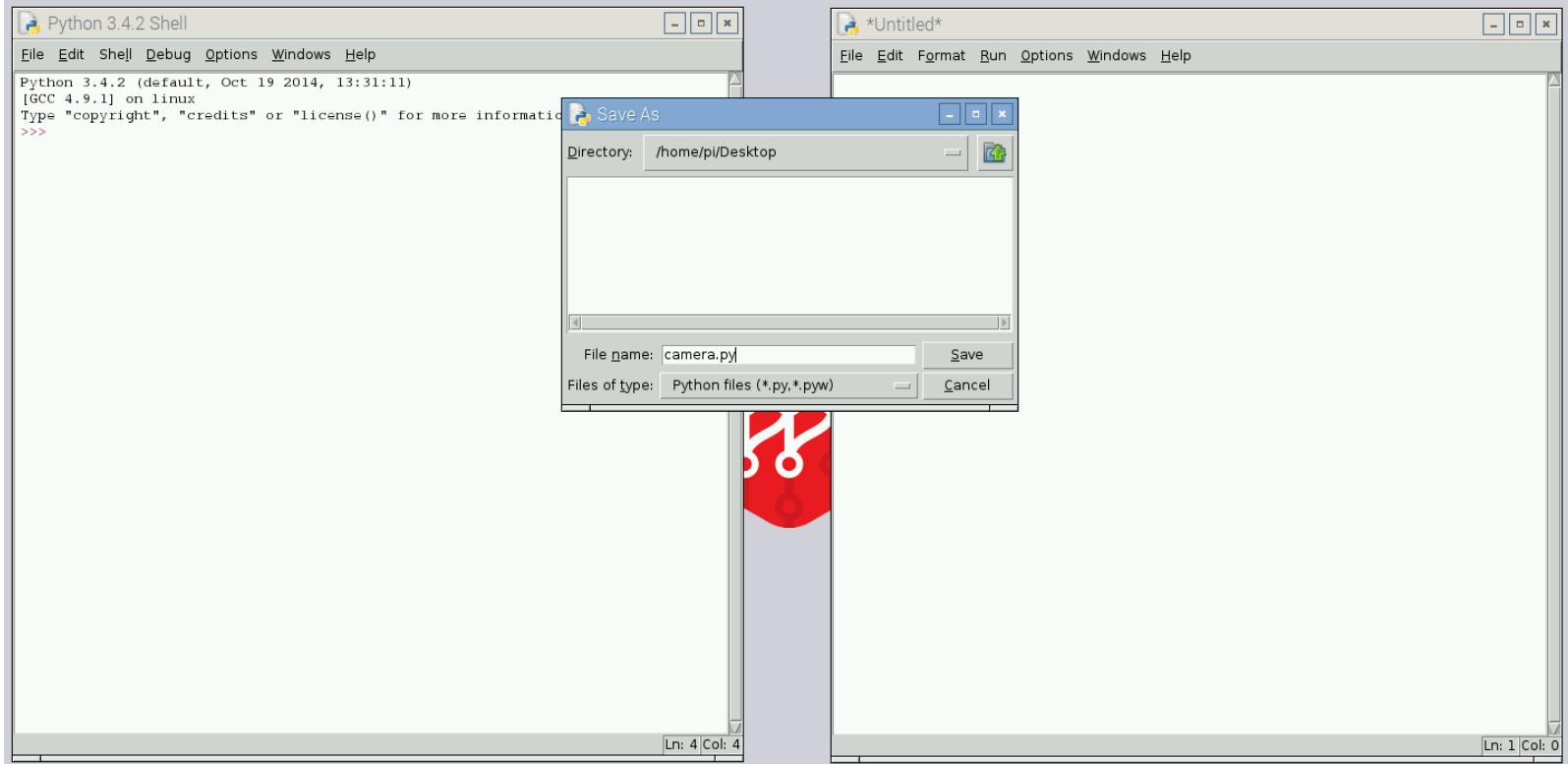
# Étape 1 – Préparation

- Dans le menu principal, «Programming », ouvrir « Python 3 »



# Étape 1 – Préparation

- Ouvrir un nouveau fichier (**CTRL + N**) et le sauvegarder (**CTRL + S**) sur le Desktop sous le nom ***camera.py***



# Étape 1 – Préparation

- Écrire le code suivant :

```
01. from picamera import PiCamera  
02. from time import sleep  
03.  
04. camera = PiCamera()  
05.  
06. camera.start_preview()  
07. sleep(5)  
08. camera.stop_preview()
```



# Étape 1 – Préparation

- Sauvegarder en appuyant sur **CTRL + S** et démarrer le programme en appuyant sur **F5**
- Hello World!

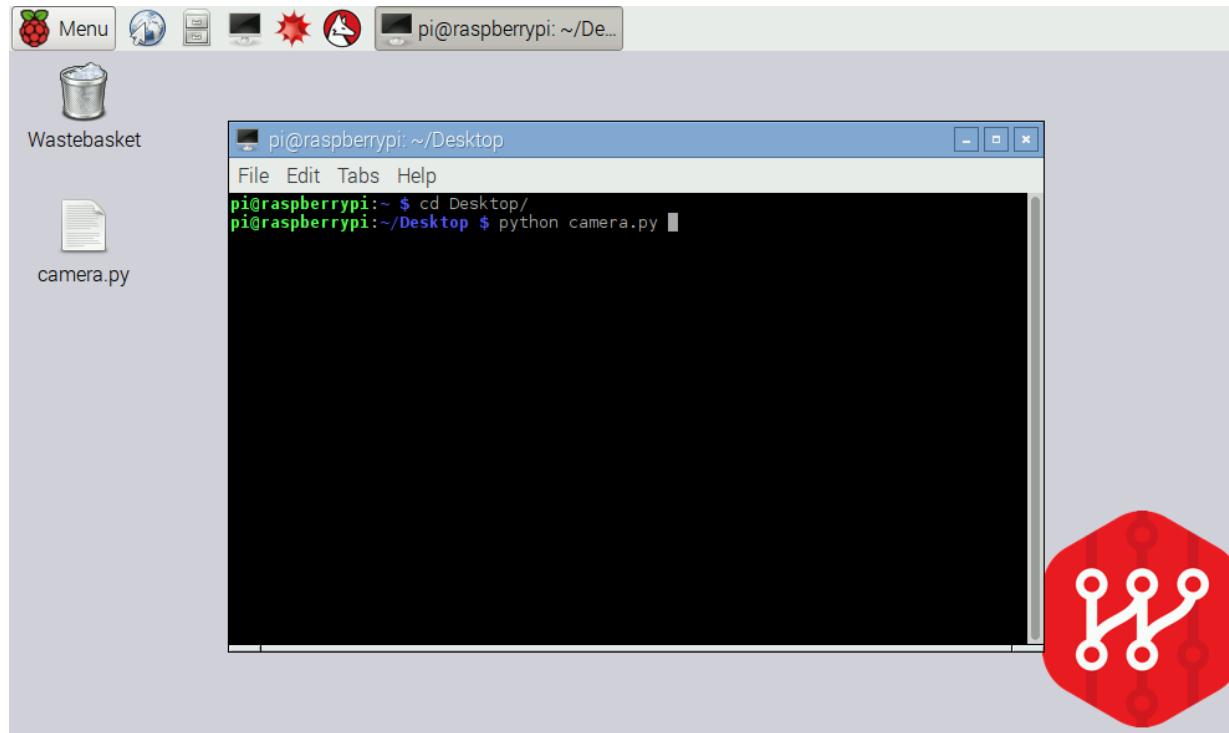
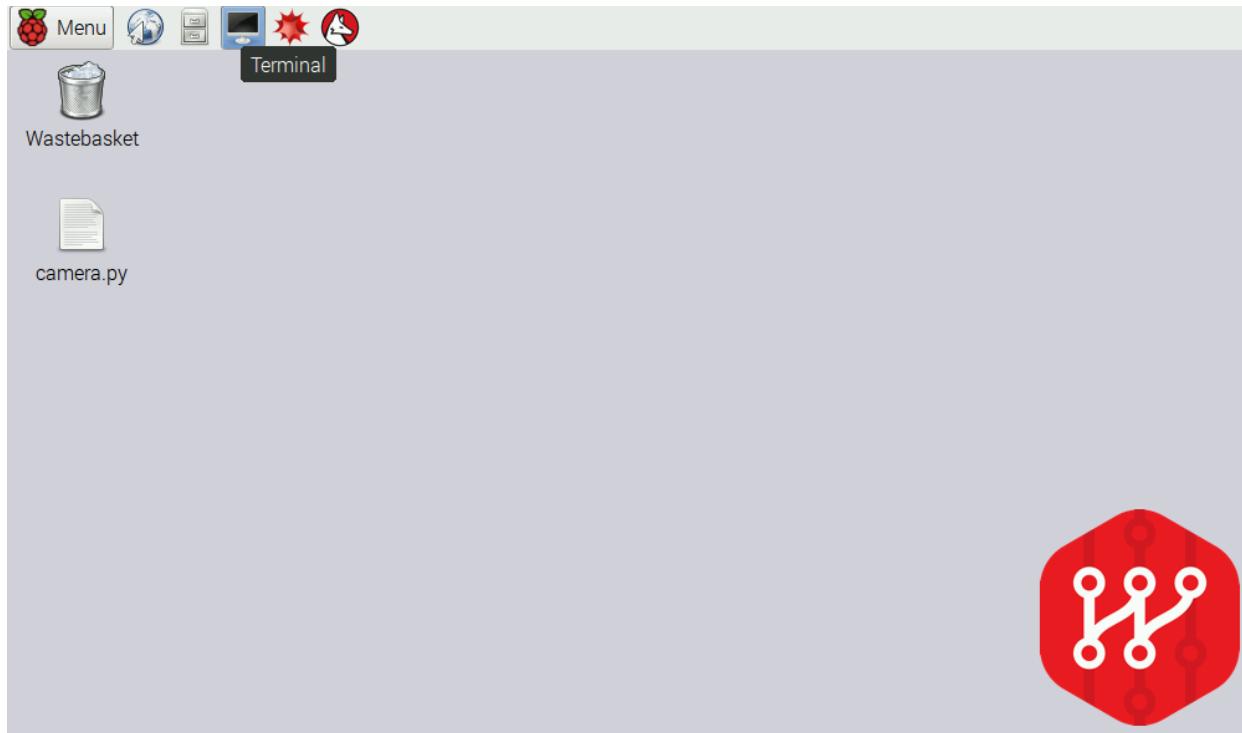


# Étape 2 – Contrôler la caméra

- Comment contrôler sa caméra pour l'éteindre manuellement par exemple?
  - Utilisation du **terminal** qui permet de recevoir des « input »
  - Utilisation du clavier comme « input »

# Étape 2 – Contrôler la caméra

- Lancer son programme en ligne de commande dans le terminal



# Étape 2 – Contrôler la caméra

- Ajoutons des **import** pour gérer les « input »

```
01. import RPi.GPIO as io
02. io.setmode(io.BCM)
03. import sys, tty, termios, time
04. from picamera import PiCamera
05. from time import sleep
```



# Étape 2 – Contrôler la caméra

- Fonction ***getch()*** permettant de récupérer la touche qui a été enfoncée

```
01. def getch():
02.     fd = sys.stdin.fileno()
03.     old_settings = termios.tcgetattr(fd)
04.     try:
05.         tty.setraw(sys.stdin.fileno())
06.         ch = sys.stdin.read(1)
07.     finally:
08.         termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
09.     return ch
```



# Étape 2 – Contrôler la caméra

- Nous commençons notre programme comme vu à l'étape 1 (démarrer la caméra)

```
01. | camera = PiCamera()  
02. | camera.start_preview()
```

# Étape 2 – Contrôler la caméra

- Ajout d'une boucle pour récupérer la touche enfoncée
  - La caméra se coupe si et seulement si la touche « X » est enfoncée

```
01. while True:  
02.     char = getch()  
03.     print(char)  
04.  
05.     if(char == "x"):  
06.         camera.stop_preview()  
07.         break  
08.  
09.     char = ""
```



Étape 1 – Préparation

Étape 2 – Contrôler la caméra

**Étape 3 – Afficher les raccourcis**

Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

# Étape 3 – Afficher les raccourcis

- Beaucoup de fonctionnalités disponibles
- Beaucoup de raccourcis clavier à retenir
  - => Affichage d'un texte sur la preview de la caméra
  - => Affichage des touches correspondantes à leur fonctionnalité

# Étape 3 – Afficher les raccourcis

- Les variables dont nous avons besoin :
  - Un **flag** qui nous permet de savoir si on doit afficher ou cacher les raccourcis
  - Deux variables **showShortcutsText** et **hideShortcutsText** qui contiennent le texte qui sera affiché à l'écran

```
01. flag = True
02.
03. showShortcutsText = "Appuyer sur \"i\"" pour afficher les raccourcis"
04. hideShortcutsText = "Appuyer sur \"i\"" pour cacher les raccourcis \n x : exit"
```



# Étape 3 – Afficher les raccourcis

- Afficher du texte sur la caméra

```
01. | camera = PiCamera()  
02. | camera.start_preview()  
03. | camera.annotate_text = showShortcutsText
```



Étape 1 – Préparation

Étape 2 – Contrôler la caméra

**Étape 3 – Afficher les raccourcis**

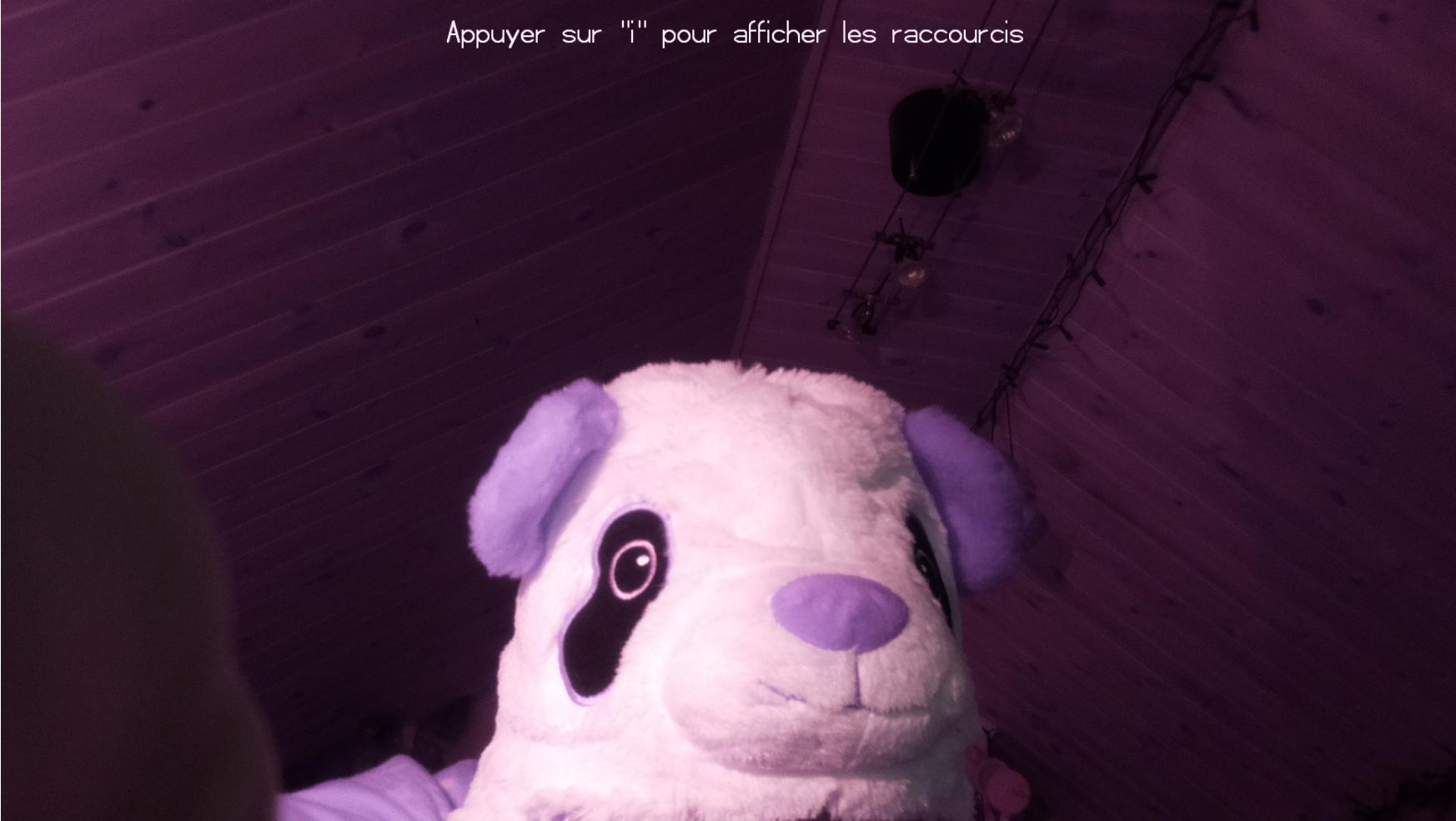
Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

# Étape 3 – Afficher les raccourcis

Appuyer sur "i" pour afficher les raccourcis



**WAJUG**

Étape 1 – Préparation

Étape 2 – Contrôler la caméra

**Étape 3 – Afficher les raccourcis**

Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

# Étape 3 – Afficher les raccourcis

Appuyer sur "i" pour afficher les raccourcis



**WAJUG**

# Étape 3 – Afficher les raccourcis

- Afficher / Cacher les raccourcis

- Re-voilà notre **flag** ;)

```
01. while True:  
02.     char = getch()  
03.  
04.     if(char == "i"):  
05.         if flag is True :  
06.             camera.annotate_text = hideShortcutsText  
07.             flag = False  
08.         else:  
09.             camera.annotate_text = showShortcutsText  
10.             flag = True
```



# Étape 4 – Rotation

Étape 1 – Préparation  
Étape 2 – Contrôler la caméra  
Étape 3 – Afficher les raccourcis  
**Étape 4 – Rotation**  
Étape 5 – Photo/Vidéo  
Étape 6 - Effets



WAJUG

# Étape 4 – Rotation

- Ajoutons dans notre boucle ces 4 lignes de code :
  - Pour faire une rotation de 90°, ajoutons/soustrayons simplement une valeur de **90** au *camera.rotation*

```
01. if(char == "a"):  
02.     camera.rotation = camera.rotation - 90  
03.  
04. if(char == "z"):  
05.     camera.rotation = camera.rotation + 90
```



# Étape 4 – Rotation

- N'oublions pas d'ajouter les informations des touches clavier dans notre variable ***hideShortcutsText*** ;)

```
01. showShortcutsText = "Appuyer sur \"i\" pour afficher les raccourcis"  
02. hideShortcutsText = "... raccourcis \n a : rotation gauche | z : rotation droite"
```



Étape 1 – Préparation

Étape 2 – Contrôler la caméra

Étape 3 – Afficher les raccourcis

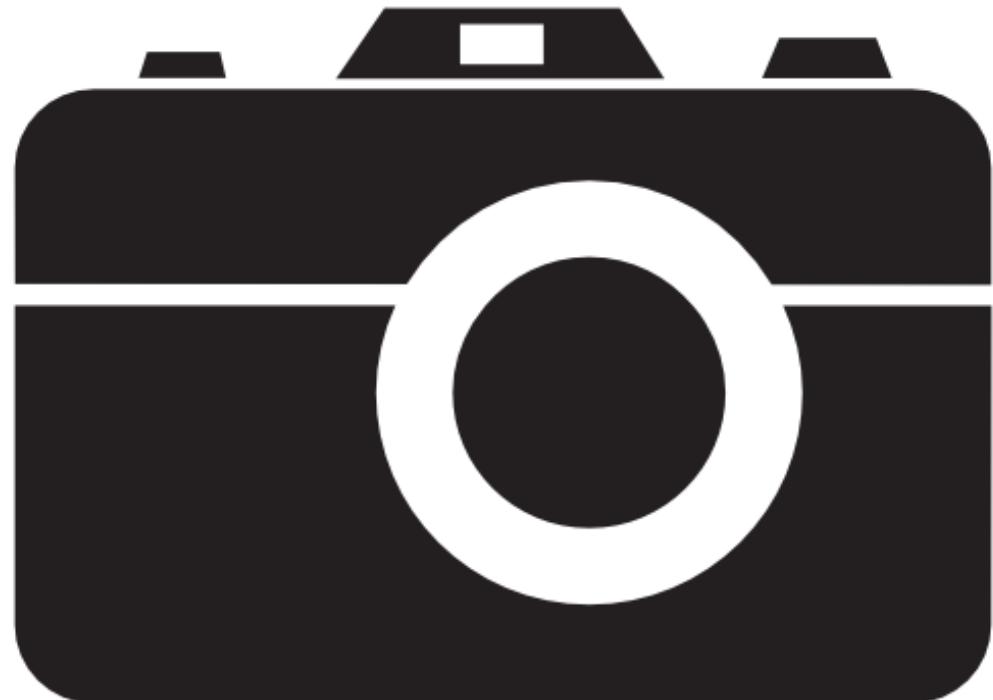
Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

# Étape 5 – Photo/Vidéo

- Que serait une caméra si nous ne pouvions prendre de photo/vidéo?



# Étape 5 – Photo/Vidéo

- Ajoutons à nos variables deux variables ***pictureNumber*** et ***videoNumber***
  - Ces deux variables sont des nombres initialisés à 0 qui vont s'incrémenter à chaque photo/vidéo prise pour avoir un nom unique

```
01. flag = True  
02. pictureNumber = 0  
03. videoNumber = 0
```

- Incrémenter :

```
01. pictureNumber = pictureNumber + 1  
02. videoNumber = videoNumber + 1
```



# Étape 5 – Photo/Vidéo

- Prendre une photo en cachant le texte info des raccourcis

```
01. if(char == "o"):  
02.     camera.annotate_text = ""  
03.     camera.capture('/home/pi/Desktop/image%s.jpg' %pictureNumber)  
04.     pictureNumber = pictureNumber + 1  
05.     if flag is False :  
06.         camera.annotate_text = hideShortcutsText  
07.     else:  
08.         camera.annotate_text = showShortcutsText
```



# Étape 5 – Photo/Vidéo

- Start/Stop l'enregistrement d'une vidéo en cachant le texte info des raccourcis

```
01. if(char == "p"):
02.     camera.annotate_text = ""
03.     camera.start_recording('/home/pi/Desktop/video%s.h264' %videoNumber)
04.     videoNumber = videoNumber + 1
05.
06.     while True:
07.         char2 = getch()
08.         if(char2 == "p"):
09.             camera.stop_recording()
10.             break;
11.
12.         if flag is False :
13.             camera.annotate_text = hideShortcutsText
14.         else:
15.             camera.annotate_text = showShortcutsText
```



# Étape 5 – Photo/Vidéo

- N'oublions pas d'ajouter les raccourcis dans le texte :

```
01. showShortcutsText = "Appuyer sur \"i\" pour afficher les raccourcis"  
02. hideShortcutsText = "... raccourcis \n o : prendre photo | p : prendre video"
```

Étape 1 – Préparation

Étape 2 – Contrôler la caméra

Étape 3 – Afficher les raccourcis

Étape 4 – Rotation

Étape 5 – Photo/Vidéo

Étape 6 - Effets

# Étape 6 – Effets

- Amusons-nous un peu avec les effets!



**WAJUG**

# Étape 6 – Effets

- Ajoutons à nos variables une liste d'effets *effectsList* et un compteur *i* qui permettra de la parcourir :

```
01. flag = True
02. pictureNumber = 0
03. videoNumber = 0
04. i = 0
05. effectsList = ['none', 'negative', 'sketch']
```



# Étape 6 – Effets

- M pour l'effet suivant

```
01. if(char == "m"):  
02.     if(i < 3):  
03.         i += 1  
04.         camera.image_effect = effectsList[i]  
05.     else:  
06.         i = 0  
07.         camera.image_effect = effectsList[i]
```



# Étape 6 – Effets

- L pour l'effet précédent

```
01. if(char == "l"):  
02.     if(i > 0):  
03.         i-=1  
04.         camera.image_effect = effectsList[i]  
05.     else:  
06.         i = 2  
07.         camera.image_effect = effectsList[i]
```



# Étape 6 – Effets

- Petite liste d'effets :

- none
- negative
- solarize
- sketch
- denoise
- emboss
- oilpaint
- hatch
- gpen
- pastel
- watercolor
- film
- blur
- saturation
- colorswap
- washedout
- posterise
- posterise
- colorpoint
- colorbalance
- cartoon
- deinterlace1
- deinterlace2

Thank you !

Merci à tous et bonne programmation !

