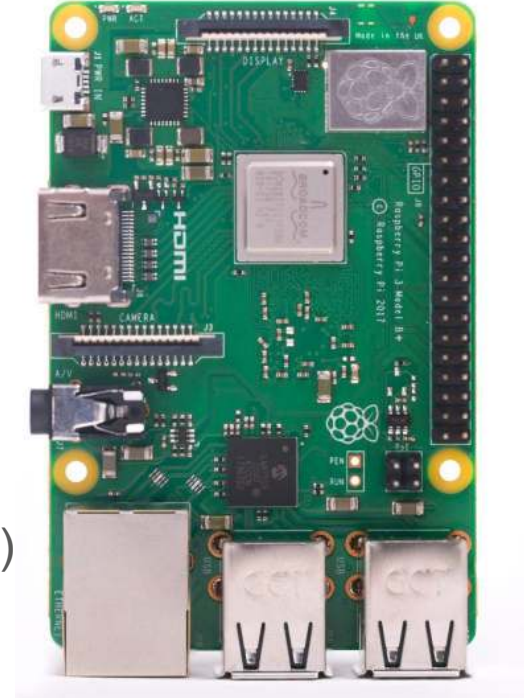


# Raspberry Pi

Les bases, ou comment débuter en douceur...  
(mais en gardant l'idée d'utiliser le RPi pour de l'embarqué)

Version du 18 mai 2019 - [www.makilab.org](http://www.makilab.org) - CC-BY-SA



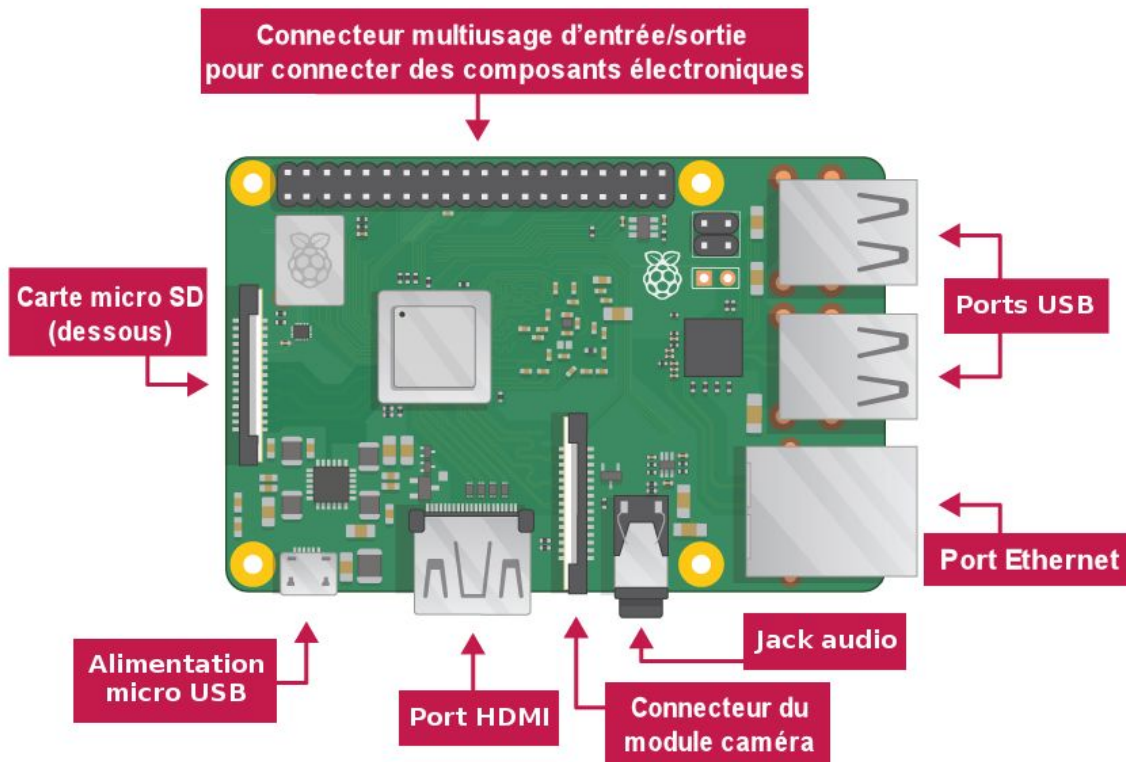
# Raspberry Pi, c'est quoi?

C'est un "vrai" petit ordinateur (PC), qui fonctionne sous Linux.

Son intérêt principal:

- pas cher (~40€)
- assez puissant
- avec des connecteurs entrée/sortie (GPIO)

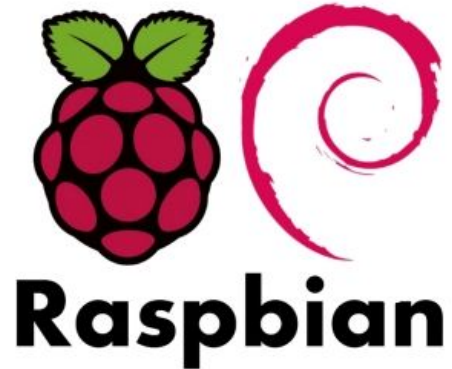
**General Purpose Input / Output**



# Au menu d'aujourd'hui:

- Installer le système d'exploitation Raspbian (Linux) sur la carte, et démarrer
- Se connecter à distance (sans écran ni clavier)
- Survivre sous Linux en ligne de commande
- Découvrir la programmation en Python
- Programmer la carte et ses entrées / sorties (GPIO)

# Installer le système d'exploitation



# Installer Linux (Raspbian)... sans écran ni clavier

C'est [la procédure "standard"](#), avec quelques petites adaptations:

1. Télécharger directement [une version de Raspbian](#)
2. Copier Raspbian sur une carte SD (par exemple avec [Etcher](#))
3. Activer SSH pour la connection à distance en plaçant un fichier nommé `ssh` sur la carte SD. (On reparlera de SSH plus tard...)
4. Configurer les paramètres Wifi sur le carte SD
5. (Optionnel et possible uniquement depuis un PC avec Linux): changer le nom d'hôte de la carte

# Configurer le Wifi directement sur la carte SD

Editer le fichier texte *wpa\_supplicant.conf* sur la partition *boot*, pour contenir:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=BE
```

```
network={
  ssid="MAKILAB"
  psk="MAKILAB0"
  key_mgmt=WPA-PSK
}
```

Le nom du réseau Wifi (Attention aux maj/min)

Le mot de passe (Attention aux maj/min)

# Changer le nom d'hôte (“hostname”)

Editer le fichier texte *hostname* sur la partition *rootfs* dans le répertoire */etc*, pour contenir le nouveau nom:

```
raspberrypi  
  
monnouveau nom
```

Possible à faire avant le premier démarrage de la carte que depuis un ordinateur avec Linux. Sinon, on le fera après le démarrage de la carte...

# OK... démarrons la carte!

Le RPi démarre dès qu'il est branché sur l'alimentation USB

- Insérer la carte microSD
- Brancher l'alimentation USB



Il faut une alimentation USB capable de délivrer 2,5 A:  
un chargeur GSM courant n'est **pas** suffisant!

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	400mA
Raspberry Pi 3 Model A+	2.5A	Limited by PSU, board, and connector ratings only.	350mA
Raspberry Pi 3 Model B+	2.5A	1.2A	500mA
Raspberry Pi Zero W/WH	1.2A	Limited by PSU, board, and connector ratings only.	150mA
Raspberry Pi Zero	1.2A	Limited by PSU, board, and connector ratings only.	100mA

Source [ <https://www.raspberrypi.org/documentation/faqs/#power> ]



# Se connecter à distance

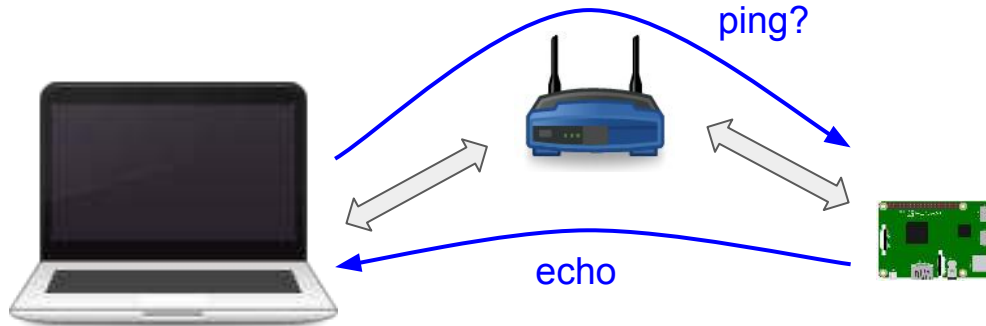
# Sans écran ni clavier... Testons la connexion

Dans un terminal (ou la cmd.exe sous windows) sur votre laptop

- `ping monnomdote.local`

Par défaut le nom d'hôte est *raspberrypi*

permet de savoir si le raspberry pi est bien connecté au réseau local (Wifi / Ethernet)



Note: les adresses “.local” ne fonctionnent sous Windows que si les [Services d'Impression Bonjour](#) (Apple) sont installés.

# Sans écran ni clavier... Établissons une connexion



Dans un terminal (ou via “mRemoteNG” sous windows)

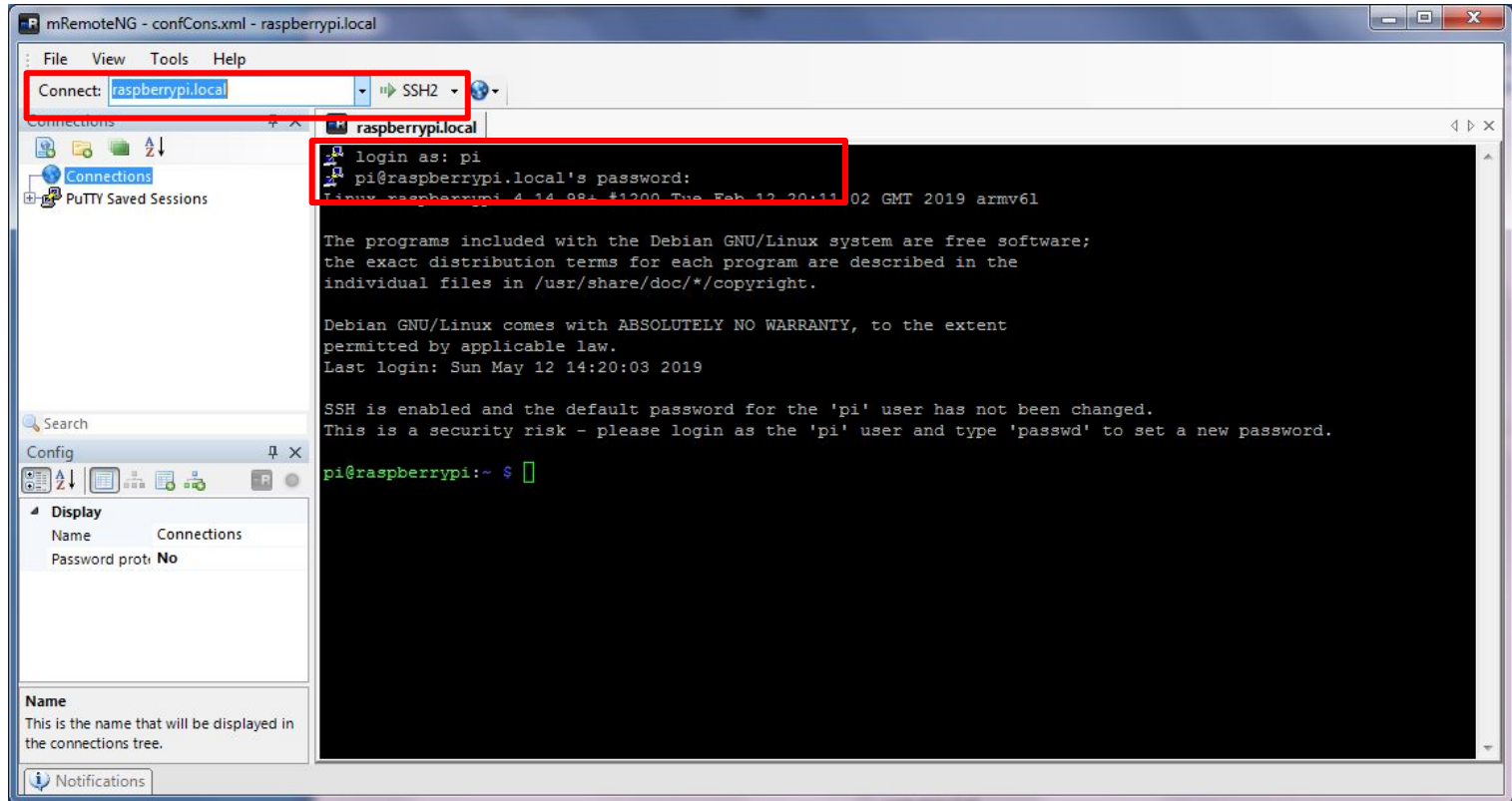
- `ssh pi@monnomdhote.local`

Par défaut le nom d'hôte est *raspberrypi*

Par défaut l'utilisateur est *pi* et  
le mot de passe *raspberrypi*

Note: les adresses “.local” ne fonctionnent sous Windows que si les [Services d'Impression Bonjour](https://support.apple.com/fr-fr/HT204491) (Apple) sont installés.

# SSH avec mRemoteNG



# Et pour l'arrêter?

Pas de bouton OFF...

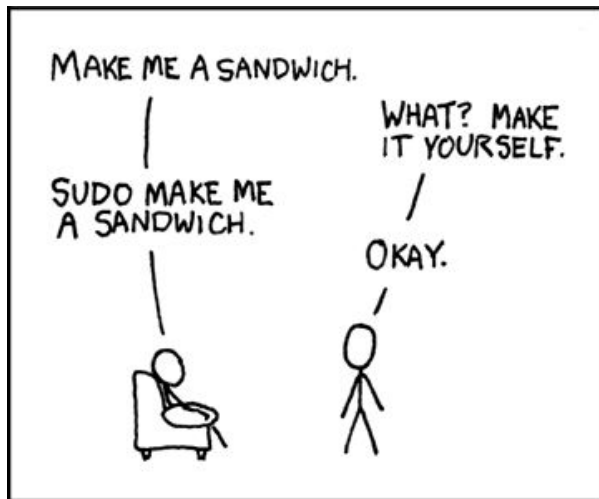
- En ligne de commande (si on est connecté au RPi):  
**sudo poweroff**
- Si pas de connection au RPi: pas d'autre solution que le débrancher...



En débranchant “sauvagement” l'alimentation d'un RPi, on risque de corrompre la carte SD. Au minimum, éviter de le faire lorsque la LED verte (indiquée “act”) clignote.

- Pour le redémarrer:  
**sudo reboot**

# C'est quoi ce "sudo"?



Il y a deux types d'utilisateurs sur un système Linux:

- le super-utilisateur (*root*)
- tous les autres.

La commande *sudo* permet d'exécuter une autre commande en tant que super-utilisateur



L'utilisateur "root" peut littéralement tout faire sur le système, donc aussi le casser, le corrompre, ... Réfléchissez-y à deux fois avant de taper une commande avec "sudo", surtout si elle est copié/collée d'internet...

# OK, quelques règles de bonne pratique

Modifier la configuration standard (le plus simple: via `sudo raspi-config` en ligne de commande):

- Changer le **mot de passe** par défaut de l'utilisateur par défaut
  - → Change user password
- Changer le **nom d'hôte** si ce n'est pas déjà fait
  - → Network options → Hostname
- Changer la **langue**, le **fuseau horaire**
  - → Localisation options → Change Locale → ajouter "fr\_BE.UTF-8" mais garder "en\_US.UTF-8"
  - → Localisation options → Change Timezone → Europe/Brussels

# Se connecter à distance en mode graphique

Il va falloir installer et configurer le nécessaire pour utiliser **VNC** (Virtual Network Computing):

Côté “serveur” (le RPi): via `sudo raspi-config`:

- → Interfacing options → VNC → Yes
- → Advanced option → Resolution → 1280x720 (ou autre)

Côté “client” (votre ordinateur):

- Un client VNC: [realVNC viewer](#)



1. Seule les versions “*Raspbian Stretch with desktop*” démarrent en mode graphique par défaut.
2. La version du protocole VNC du serveur utilisé par défaut sur RPi (RealVNC Server) n'est en pratique compatible qu'avec le client RealVNC Client :-)



# Se connecter à distance au système de fichier

Il va falloir installer et configurer le nécessaire pour utiliser **SFTP** (**SSH File Transfer Protocol**):

Côté **serveur** (le RPi):

rien à faire de plus, cela “passe” par une connection SSH

Côté **client** (votre ordinateur):

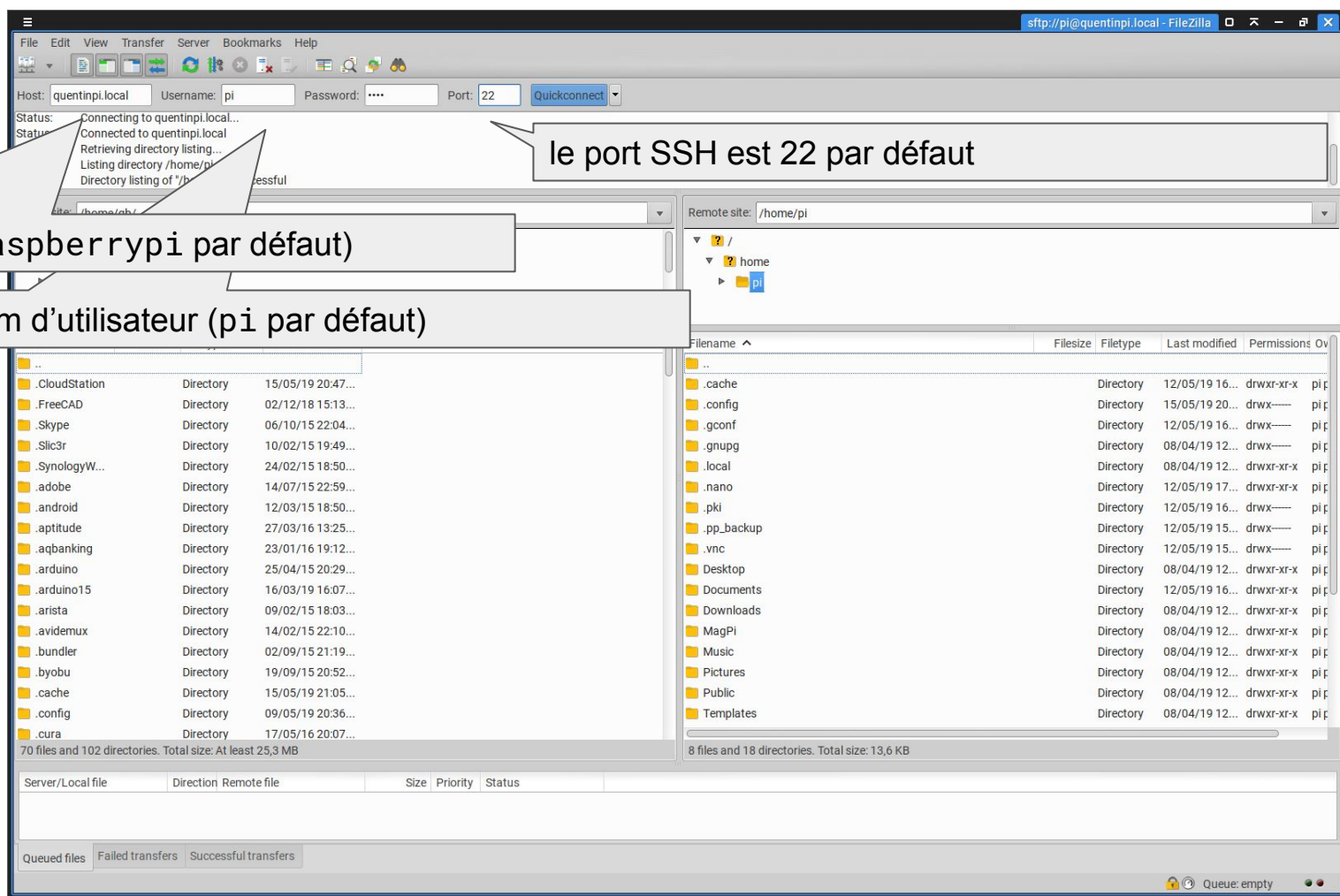
un client SFTP, par exemple [FileZilla](#) (fonctionne sous Linux/Windows/Mac OS X)

# FileZilla

Le nom d'hôte (raspberrypi par défaut)

Le nom d'utilisateur (pi par défaut)

le port SSH est 22 par défaut

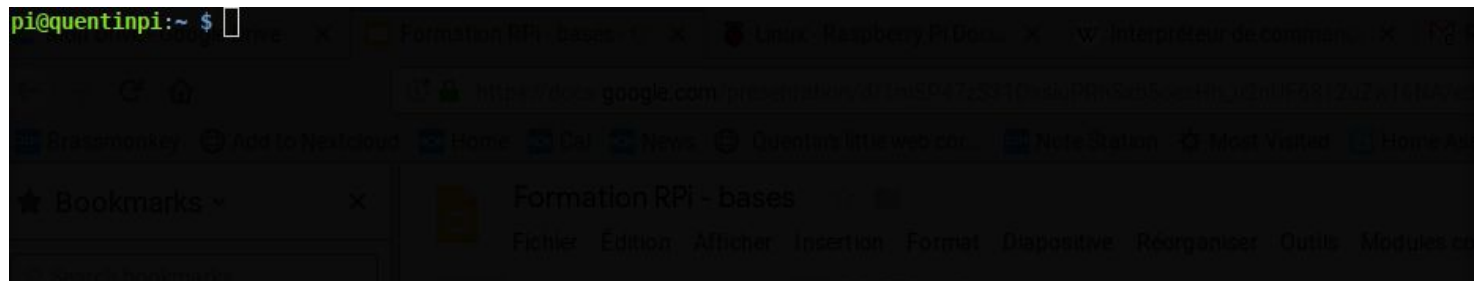


# Survivre en ligne de commande

# Survivre sous Linux en ligne de commande

“L’interpréteur de commande” (CLI en anglais - Common Line Interface) est l’interface de commande privilégiée sous Linux. On l’appelle aussi parfois “Terminal”.

Elle sera souvent votre seul moyen de dialoguer avec le RPi, surtout si il n’a pas de clavier ou d’écran connecté.



# Se déplacer...

- **pwd** affiche le répertoire actuel
- **ls** Liste les fichiers dans le répertoire actuel:
  - **-l** : affiche des infos additionnelles (permissions, date, taille)
  - **-a** : affiche tous les fichiers (y compris les fichiers “cachés” commençant par un . (point))
  - **-h**: affiche les tailles de fichiers de manière humaine (Mb, Gb, ...)
- **cd** change le répertoire actuel

Besoin d'aide? **man** affiche la page de manuel d'une commande

- L'interpréteur de commande (bash) permet l'autocomplétion avec la touche TAB, ne pas s'en priver!
- 2 répertoires spéciaux: “~” = répertoire home de l'utilisateur, et “-” le précédent répertoire

# L'arborescence classique sous Linux

```
pi@quentinpi:~$ ls -l /
total 71
drwxr-xr-x  2 root root 4096 Apr  8 12:01 bin
drwxr-xr-x  3 root root 2560 Jan  1 1970 boot
drwxr-xr-x 14 root root 3320 May 12 23:15 dev
drwxr-xr-x 112 root root 4096 May 12 17:34 etc
drwxr-xr-x  3 root root 4096 Apr  8 11:48 home
drwxr-xr-x 16 root root 4096 Apr  8 11:58 lib
drwx----- 2 root root 16384 Apr  8 12:40 lost+found
drwxr-xr-x  2 root root 4096 Apr  8 11:37 media
drwxr-xr-x  2 root root 4096 Apr  8 11:37 mnt
drwxr-xr-x  7 root root 4096 Apr  8 12:20 opt
dr-xr-xr-x 149 root root  0 Jan  1 1970 proc
drwx----- 5 root root 4096 May 12 17:32 root
drwxr-xr-x 27 root root  840 May 15 21:22 run
drwxr-xr-x  2 root root 4096 Apr  8 12:01/sbin
drwxr-xr-x  2 root root 4096 Apr  8 11:37/srv
dr-xr-xr-x 12 root root  0 May 15 21:45/sys
drwxrwxrwt 14 root root 4096 May 15 21:17/tmp
drwxr-xr-x 11 root root 4096 Apr  8 12:09/usr
drwxr-xr-x 11 root root 4096 Apr  8 12:43/var
pi@quentinpi:~$
```

**/bin** => Exécutables essentiels au système, utilisables par tous les utilisateurs (ls pwd cp)

**/boot** => fichiers permettant à Linux de démarrer

**/dev** => Point d'entrée de tous les périphériques (disque dur, écran, partition, consoles TTY)

**/etc** => contient les commandes et fichiers nécessaires à l'administrateur système (\*.conf...)

**/home** => Répertoire personnel des utilisateurs

**/lib** => contient les bibliothèques partagées essentielles au système lors du démarrage

**/media** => Contient les points de montages pour les médias amovibles

**/mnt** => Point de montage pour monter temporairement un système de fichiers

**/opt** => Emplacement pour des applications installées hors gestionnaire de paquets (logiciels optionnels)

**/proc** => Répertoire virtuel pour les informations système (états du noyau et des processus système)

**/root** => Répertoire personnel de l'administrateur (le seul qui n'est pas dans /home)

**/sbin** => Contient les programmes système essentiels utilisables par l'admin uniquement.

**/srv** => Données pour les services du système (ex: documents de comptes FTP)

**/tmp** => Répertoire fichier temporaires, effacé au redémarrage

**/usr** => Contient des programmes installés (/usr/bin) avec leur librairies (/usr/lib ou /usr/lib64) tels que firefox, libreoffice, ...)

**/var** => contient les données variables (fichiers de log) mais parfois les bases de données (/var/lib/mysql) et les pages de site web (/var/www/html)

Sous Linux, "everything is file"

# Manipuler les fichiers...

- **cp** copie un fichier
- **mv** déplace un fichier
- **rm** efface un fichier
- **mkdir** crée un répertoire
- **rmdir** supprime un répertoire si il est vide (ne contient pas de fichier)

Toutes ces commandes peuvent être utilisées avec les “expansions”

- **\*** remplace tous les caractères
- **?** remplace un caractères
- **[ . . . ]** remplace les caractères entre []: exemple [0-9]

# Editer les fichiers...

- **less** affiche le contenu d'un fichier texte (mais ne permet pas de l'éditer)  
(utiliser q pour quitter)
- **nano** est l'éditeur de texte par défaut: simple et efficace.

Mise en pratique: créer un répertoire "Scripts" et y mettre un fichier "lisezmoi.txt"

```
$> cd ~  
$> mkdir Scripts  
$> nano lisezmoi.txt
```

Ajouter le texte "Ce répertoire contiendra nos scripts python" et sauver

```
$> less lisezmoi.txt
```



# Installer et mettre à jour des programmes

Raspbian utilise le gestionnaire de paquet “APT”:

- **apt search** pour rechercher un paquet
- **apt show** pour les informations détaillées d'un paquet
- **sudo apt install** pour installer un paquet
- **sudo apt remove** pour désinstaller un paquet
- **sudo apt update** pour mettre à jour la liste des paquets (programmes)
- **sudo apt upgrade** pour mettre à jour les paquets installés à la dernière version

Un exemple pratique: installons les pages de manuels (man pages) en français

```
$> apt search manpages french
...
$> sudo apt install manpages-fr-extra
```

# S'initier à Python

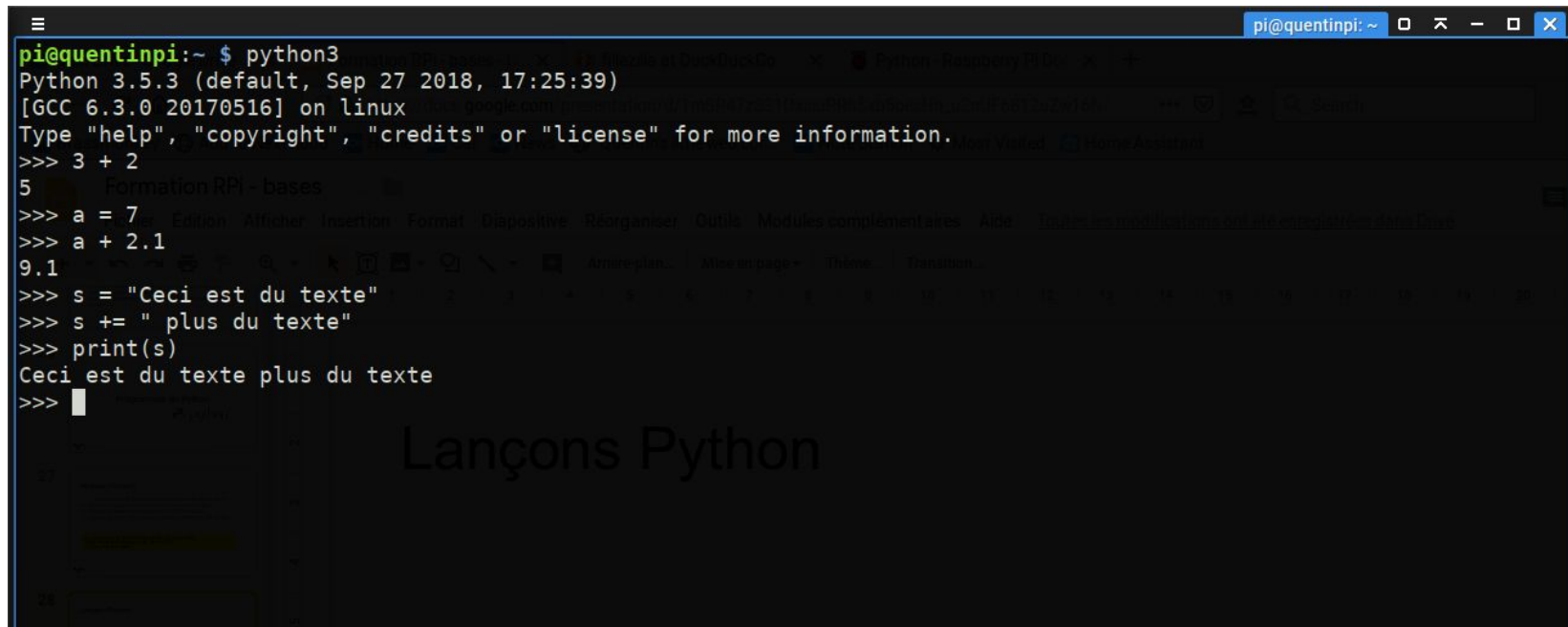


# Pourquoi Python?

- C'est le standard "de facto" pour la programmation sur Raspberry Pi.
- C'est un langage de programmation qui reste simple et lisible.
- Il permet de facilement programmer les GPIO's de la carte
- L'écosystème (les bibliothèques ou modules additionnels) est énorme

Il y a deux versions de Python: Python 2.x et Python 3.x (sorti en 2008).  
Python 2 n'est plus développé depuis 2.7 (sorti en 2010).  
→ Python 3 est recommandé!

# Lançons Python



```
pi@quentinpi:~ $ python3
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 + 2
5
>>> a = 7
>>> a + 2.1
9.1
>>> s = "Ceci est du texte"
>>> s += " plus du texte"
>>> print(s)
Ceci est du texte plus du texte
>>>
```

Lançons Python

# Un script (fichier .py)

```
$> cd ~/Scripts  
$> nano helloworld.py
```

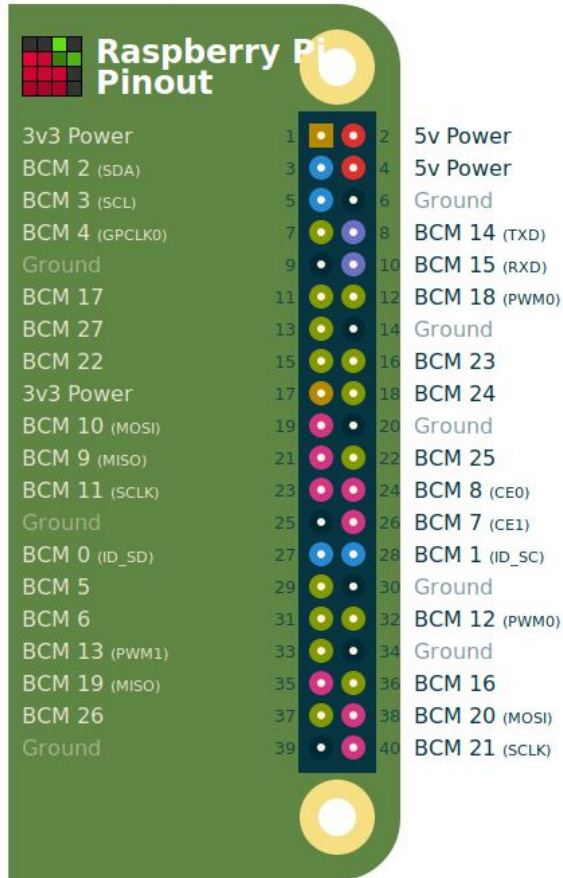
```
import datetime  
  
# Affiche un texte d'accueil, puis la date et l'heure courante  
print("Bonjour tout le monde")  
aujourd'hui = datetime.date.today()  
print("Nous sommes le", aujourd'hui)  
# weekday() renvoie le jour de la semaine, 0 pour lundi, 6 pour  
dimanche  
if aujourd'hui.weekday() == 5 or aujourd'hui.weekday() == 6:  
    print("Chouette c'est le weekend")  
else:  
    print("C'est la semaine")
```

```
$> python3 helloworld.py
```



# Les GPIO's et la programmation “physique”

# GPIO's



The diagram shows the 40-pin GPIO header of a Raspberry Pi. It is organized into two columns of 20 pins each. The left column is labeled with pin numbers 1 to 39, and the right column is labeled with pin numbers 2 to 40. Each pin is represented by a colored dot. The colors correspond to the pin's function: 3V3 Power (yellow), 5v Power (red), Ground (black), BCM 2 (SDA) (blue), BCM 3 (SCL) (blue), BCM 4 (GPCLK0) (green), BCM 14 (TXD) (purple), BCM 15 (RXD) (purple), BCM 17 (green), BCM 18 (PWM0) (green), BCM 22 (green), BCM 23 (green), BCM 24 (green), BCM 25 (green), BCM 8 (CE0) (pink), BCM 7 (CE1) (pink), BCM 1 (ID\_SC) (blue), BCM 12 (PWM0) (green), BCM 16 (pink), BCM 20 (MOSI) (pink), and BCM 21 (SCLK) (pink). The functions are listed to the left and right of the pin numbers.

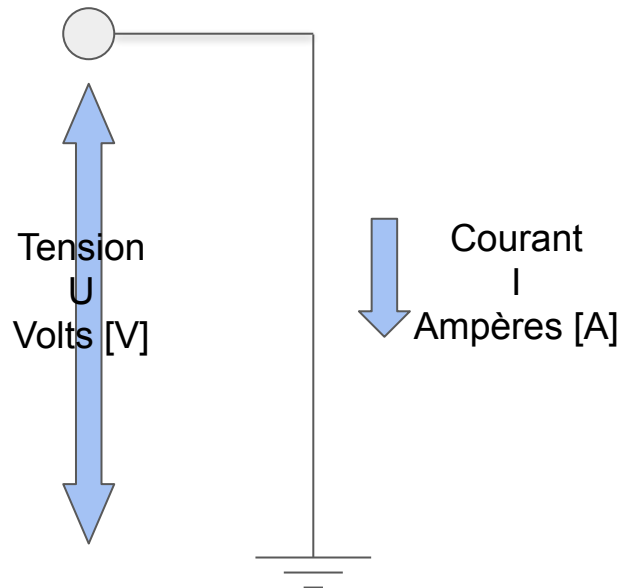
Pin	Function
1	3v3 Power
2	5v Power
3	BCM 2 (SDA)
4	5v Power
5	BCM 3 (SCL)
6	Ground
7	BCM 4 (GPCLK0)
8	BCM 14 (TXD)
9	Ground
10	BCM 15 (RXD)
11	BCM 17
12	BCM 18 (PWM0)
13	BCM 27
14	Ground
15	BCM 22
16	BCM 23
17	3v3 Power
18	BCM 24
19	BCM 10 (MOSI)
20	Ground
21	BCM 9 (MISO)
22	BCM 25
23	BCM 11 (SCLK)
24	BCM 8 (CE0)
25	Ground
26	BCM 7 (CE1)
27	BCM 0 (ID_SD)
28	BCM 1 (ID_SC)
29	BCM 5
30	Ground
31	BCM 6
32	BCM 12 (PWM0)
33	BCM 13 (PWM1)
34	Ground
35	BCM 19 (MISO)
36	BCM 16
37	BCM 26
38	BCM 20 (MOSI)
39	Ground
40	BCM 21 (SCLK)

```
$> pinout
```

# Power on!

Quelques notions d'électricité:

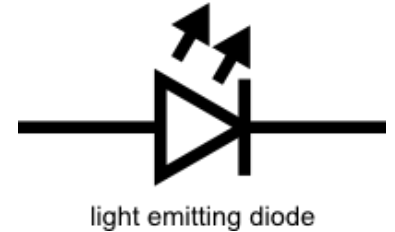
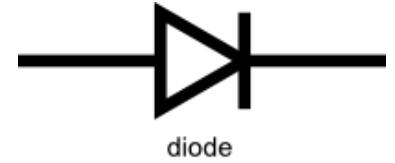
- C'est une forme d'énergie (comme la chaleur, la lumière, ...)
- Elle se déplace dans des conducteurs (comme les fils)
- On peut la convertir de et vers d'autres formes d'énergie
- AC/DC





# LED the fun begin

Allumons une LED



Anode +    - Cathode

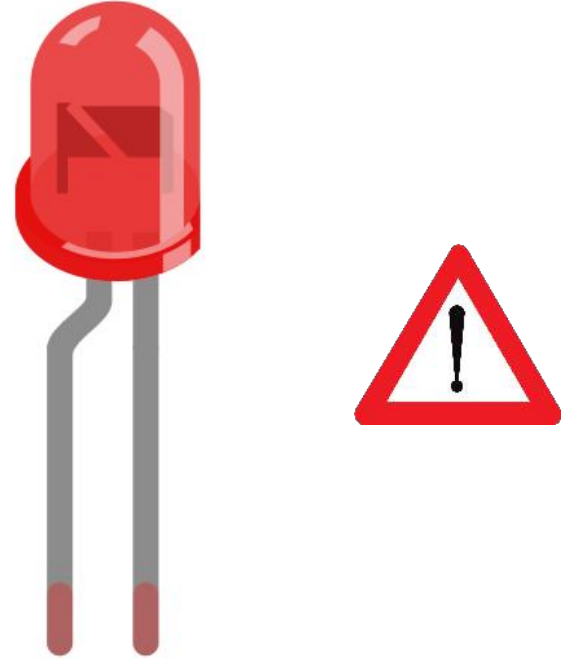
# LED the fun begin

Trop de courant tue une LED,  
en règle générale

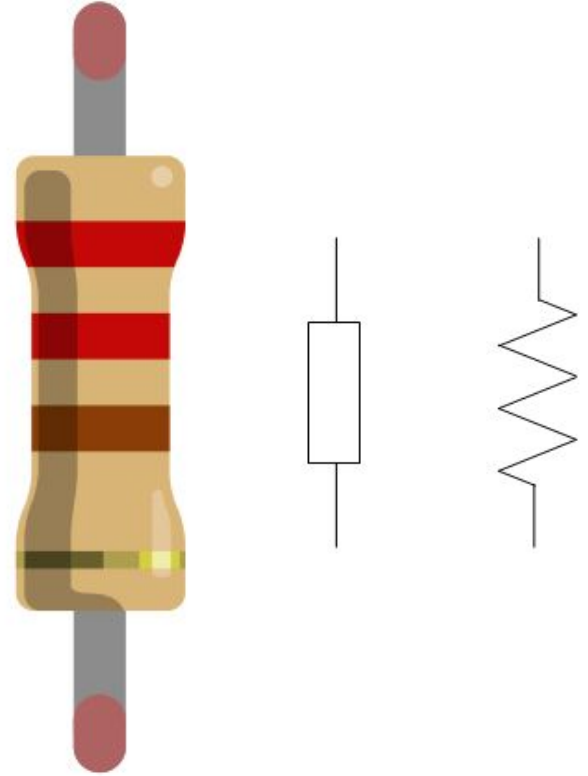
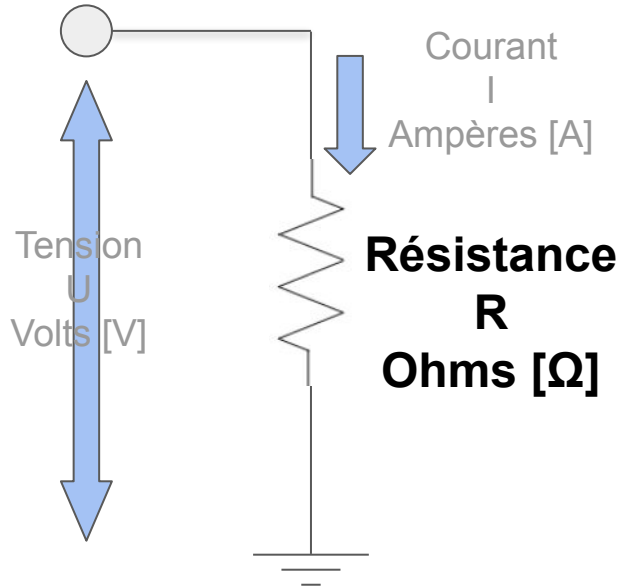
max **20mA**

et le Raspberry Pi peut supporter jusqu'à

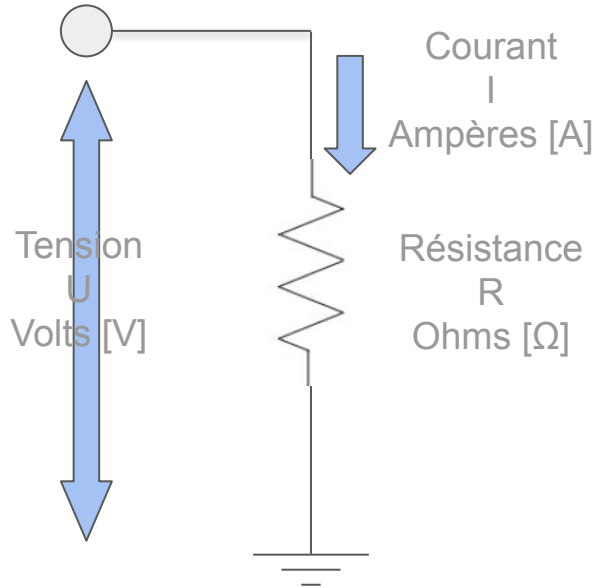
**50mA** par pin I/O



# Ohm my god



# Ohm my god

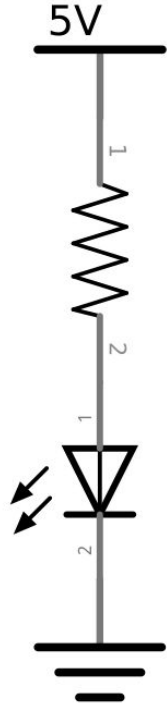


$$U = R * I$$

$$I = U / R$$

$$R = U / I$$

# En pratique



?

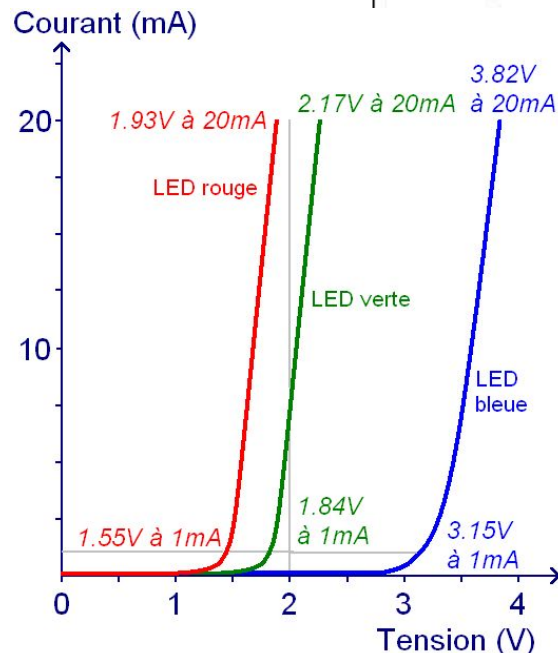
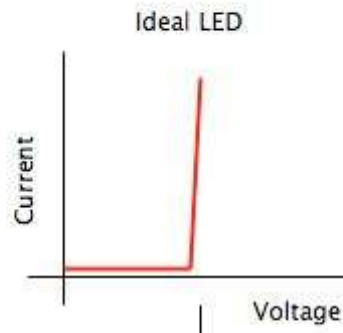
Quel est  $R$  pour que  $I$  soit  $\leq 20\text{mA}$

# Retour sur la LED

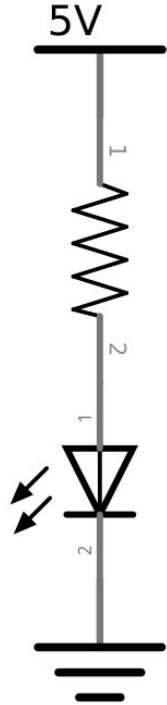
En 1<sup>ère</sup> approximation:

$$R_{LED} \approx 0 \Omega \text{ et } U_{LED} \approx \text{entre } 1.8 \text{ V et } 3.2 \text{ V}$$

DIALIGHT P/N	EMITTED COLOR	MATERIAL	LENS COLOR	LUMINOUS INTENSITY (mcd)			DOMINANT WAVELENGTH (nm)			FORWARD VOLTAGE (V)			VIEWING ANGLE
				If = 20 ma			If = 20 ma			If = 20 ma			
				MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
598-8010-107F	RED	AlInGaP	Water Clear	30	40	80	630	635	642	1.7	2.2	2.4	140
598-8020-107F	RED-ORANGE	AlInGaP	Water Clear	120	150	200	620	625	630	1.7	2	2.4	140
598-8030-107F	ORANGE	AlInGaP	Water Clear	70	-	150	600	-	610	1.7	2	2.4	140
598-8040-107F	YELLOW	AlInGaP	Water Clear	100	130	160	590	-	595	1.7	2	2.2	140
598-8050-107F	YELLOW	AlInGaP	Water Clear	100	130	160	583	-	590	1.7	2	2.4	140
598-8060-107F	YELLOW-GREEN	AlInGaP	Water Clear	20	40	60	570	-	575	1.8	2	2.4	140
598-8070-107F	GREEN	GaP	Water Clear	10	20	40	562	-	570	1.8	2	2.4	140
598-8081-107F	GREEN	InGaP	Water Clear	220	300	400	520	523	525	3	3.2	3.5	140
598-8091-107F	BLUE	InGaP	Water Clear	90	140	160	470	473	475	2.8	3.2	3.5	140



# En pratique

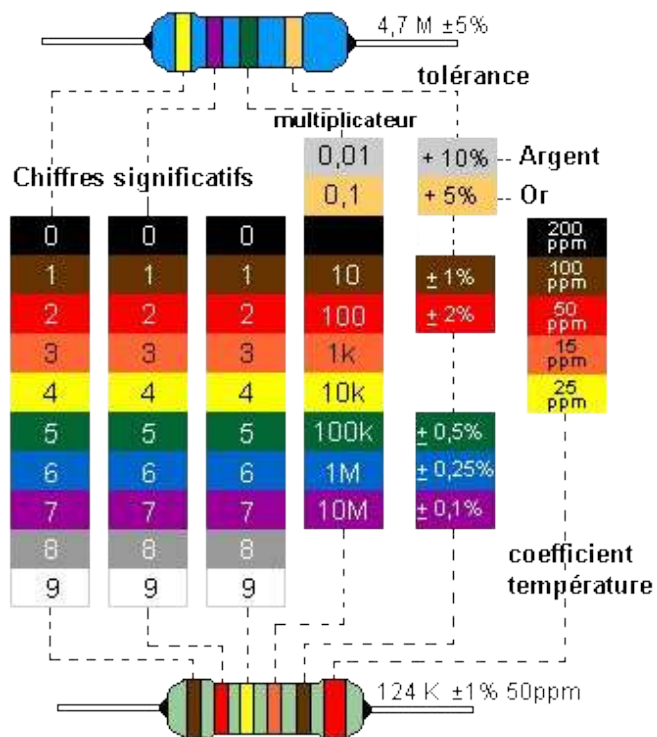


On peut la calculer précisément ...

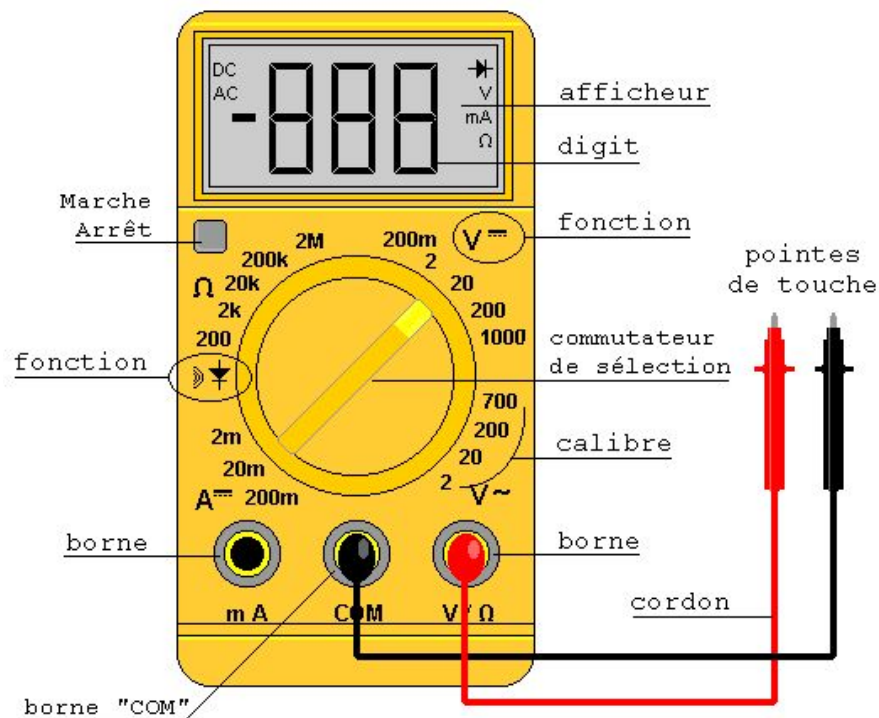
ou retenir que typiquement:

**220  $\Omega$**

# Ma résistance a quelle valeur?

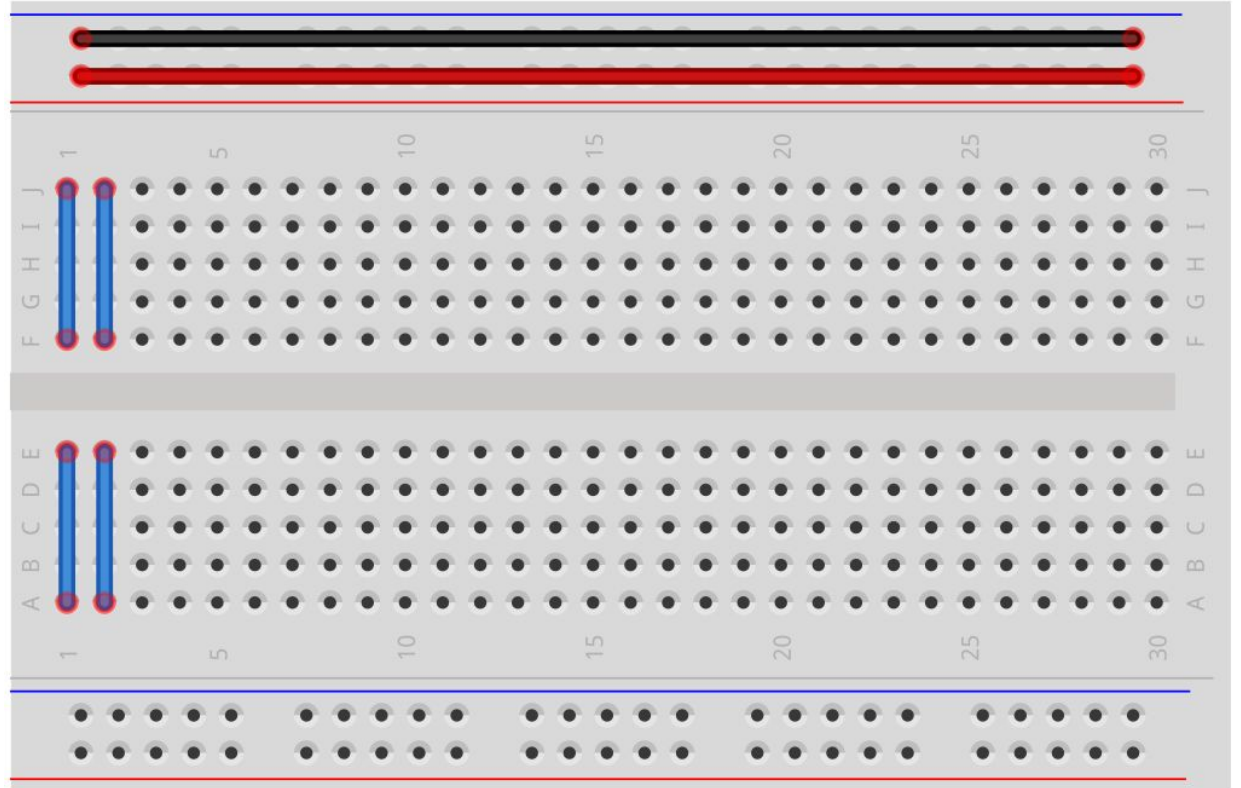


Ou...





# “Breadboard”

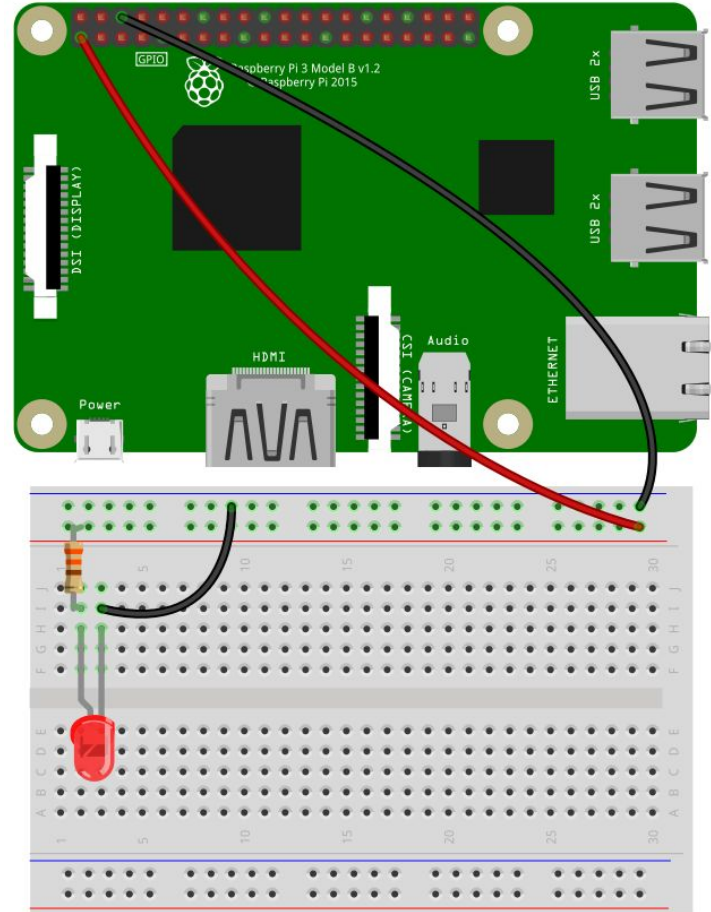


# Circuit LED

- Un circuit doit être “fermé” pour fonctionner.
- Toute l'énergie d'un circuit est utilisée/convertie, donc la tension chute toujours de la tension d'entrée à zéro.
- Le courant électrique cherche toujours le circuit de moindre résistance...



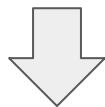
Court-circuit



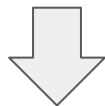
# Sécurité ?

Faible courants et tensions

Pas de composants capacitifs utilisés aujourd'hui



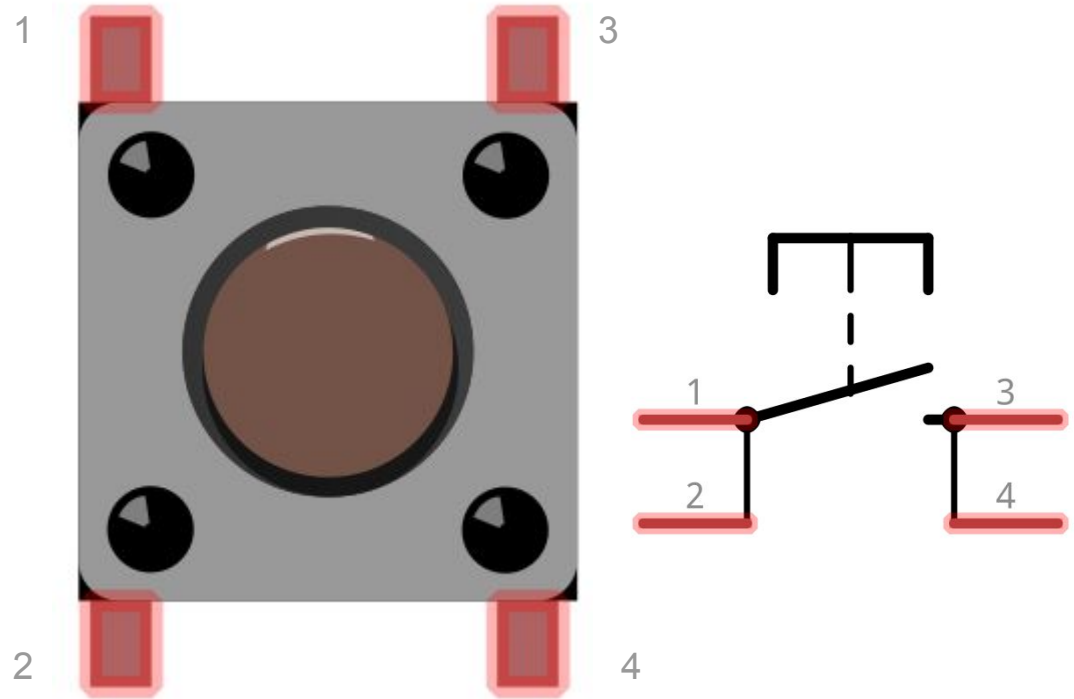
**Pas de risque pour vous...** mais un peu pour le matériel



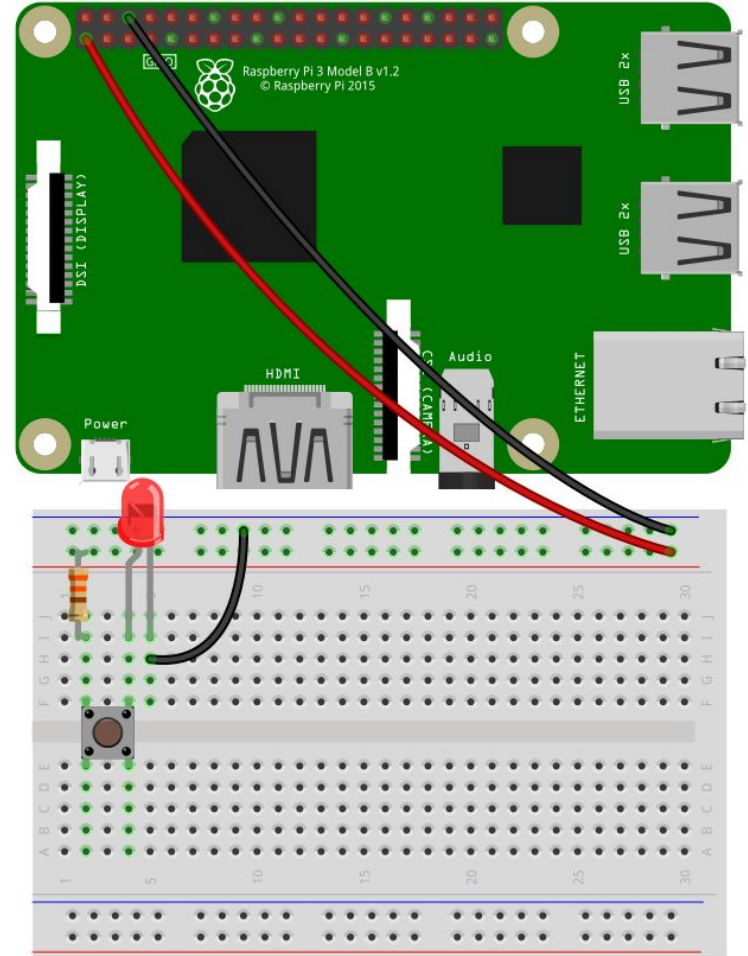
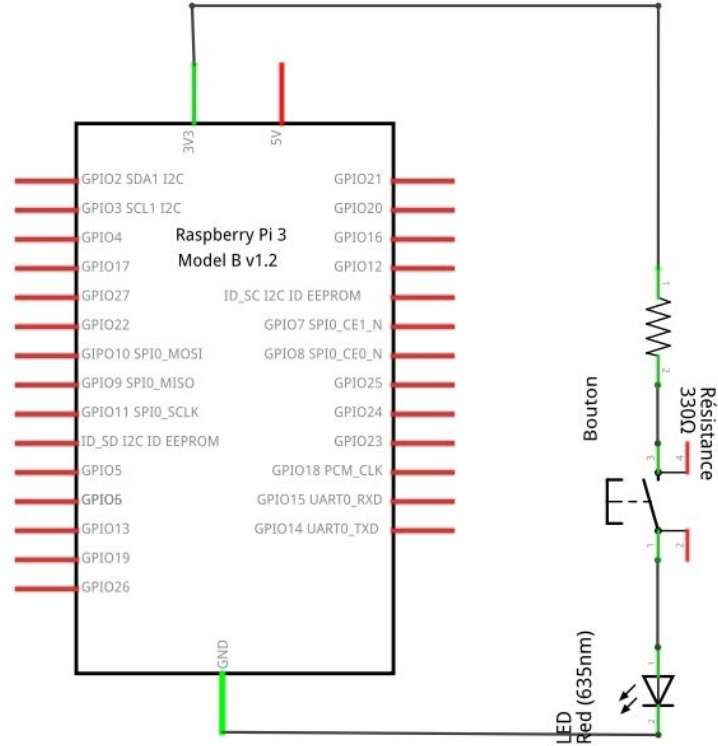
Conseillé de travailler hors tension

& de vérifier le montage avant mise sous tension

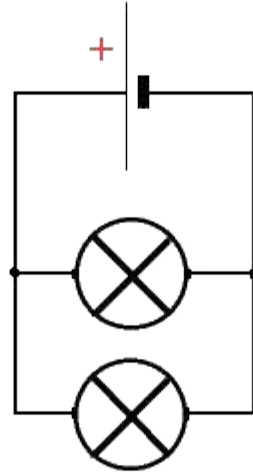
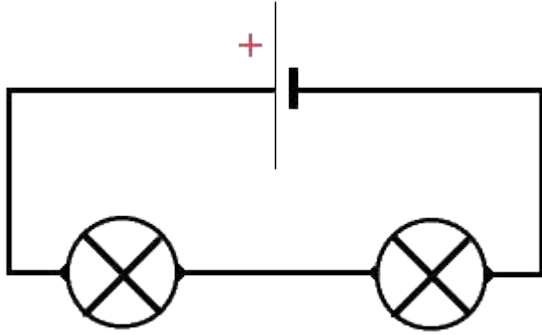
# On / Off



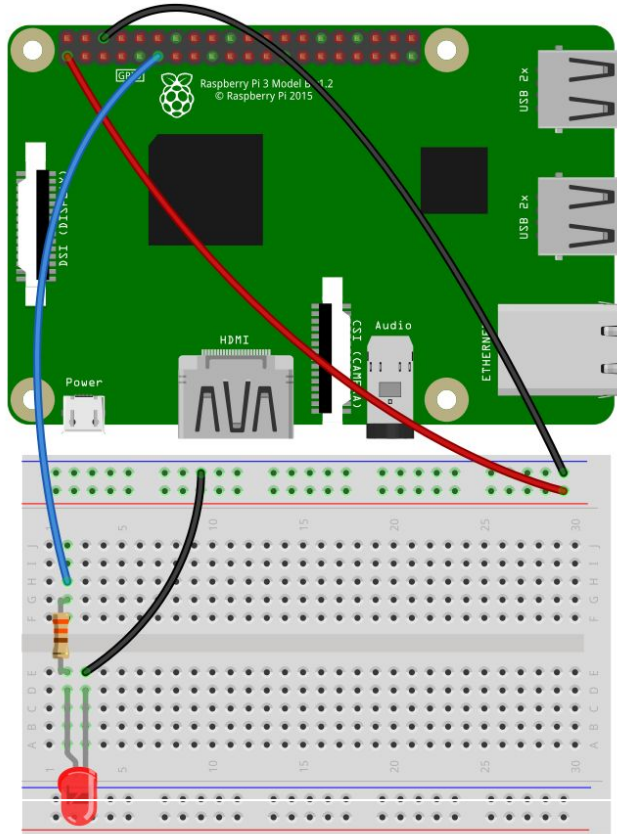
# Circuit On/Off



# Série vs Parallèle



# ~~Allumons~~ Programmons l'allumage d'une LED



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

```
led = LED(17)
```

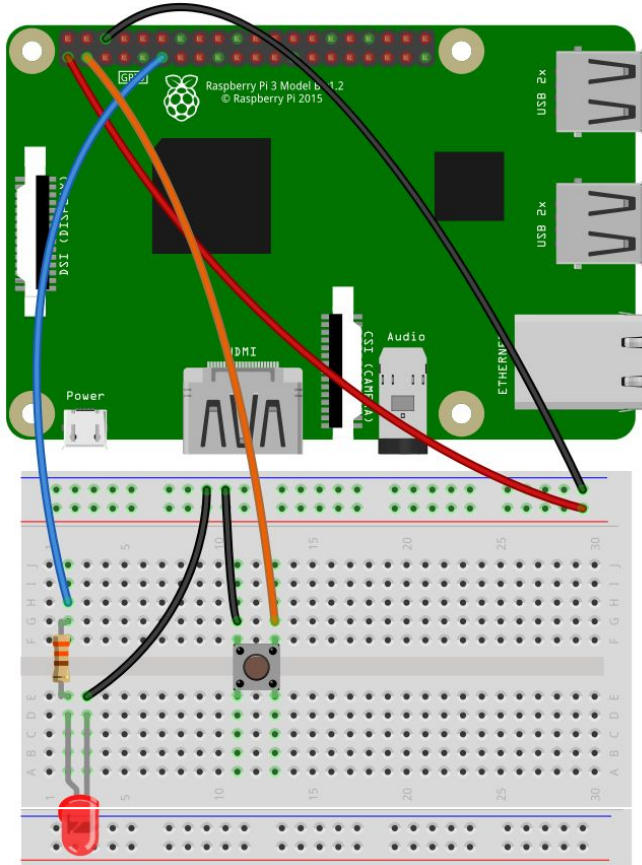
```
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```



Attention: les GPIO's du Raspberry Pi fonctionnent en 3.3v (au lieu de 5V sur l'Arduino)

Source: [ <https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/> ]

# Ajoutons un bouton



```
from gpiozero import LED, Button  
from signal import pause
```

```
led = LED(17)  
button = Button(2)
```

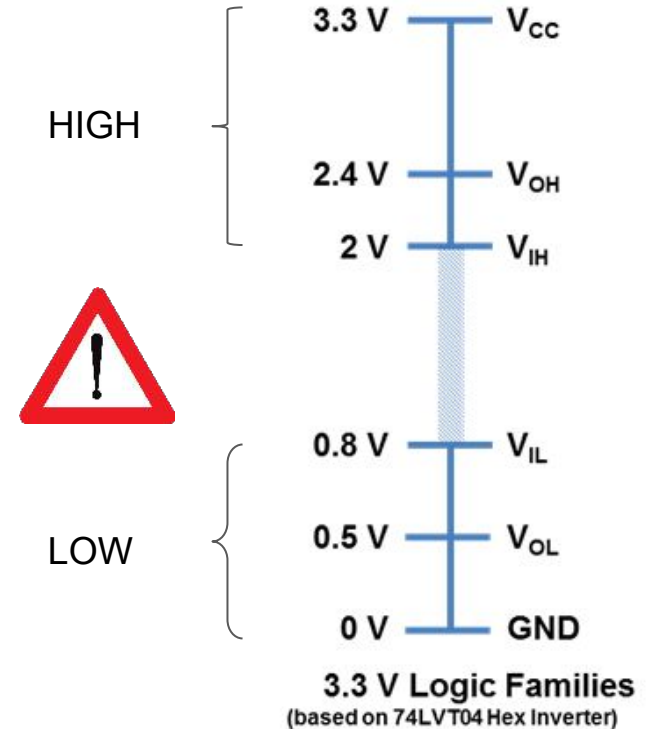
```
button.when_pressed = led.on  
button.when_released = led.off
```

```
pause()
```



# Encore un petit mot sur les entrées digitales

Attention à la zone grise entre les niveaux  
HIGH et LOW



# That's all folks

Formation faite au Makilab ([www.makilab.org](http://www.makilab.org) - [info@makilab.org](mailto:info@makilab.org) - Quentin Berten)

Diffusé sous licence



(<http://creativecommons.org/licenses/by-sa/4.0/>)