Programación en Python en la RaspberryPi

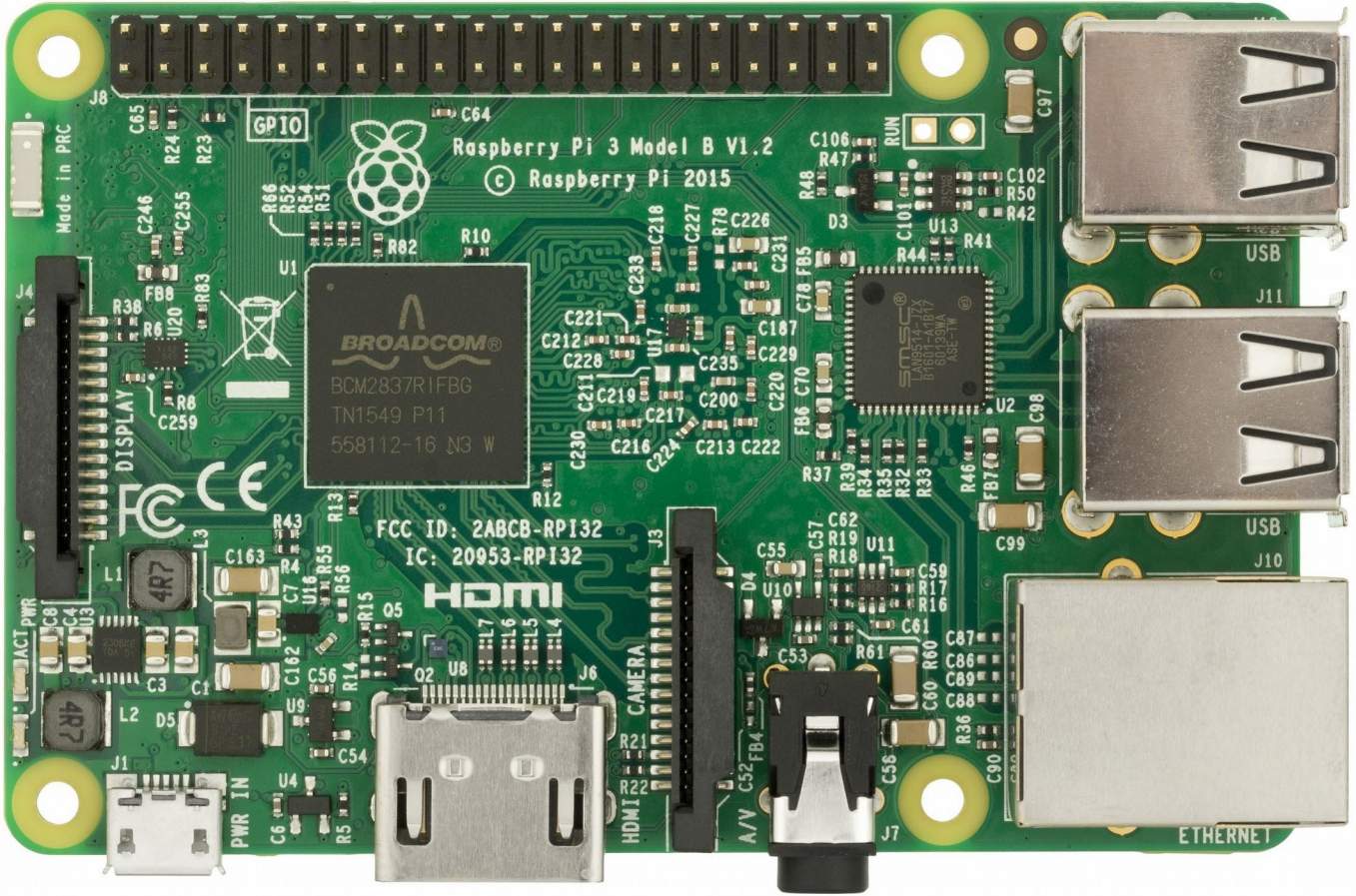
M. C. Miguelangel Fraga Aguilar

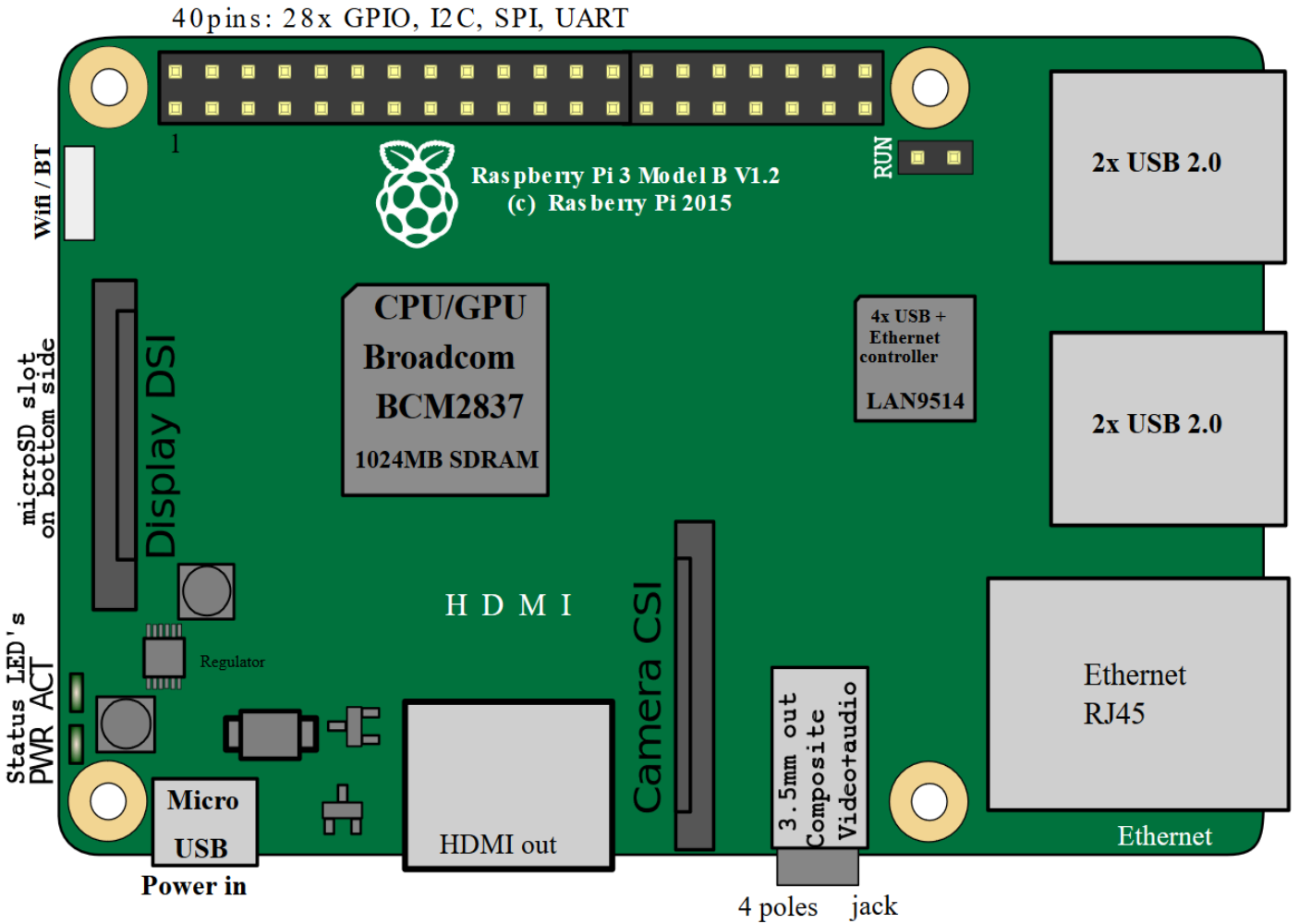
RaspberryPi

Historia de la Raspberry Pi

La Raspberry Pi es una computadora en una sola tarjeta (Single-Board Computer) creada por la Raspberry Pi Foundation para promover la enseñanza de la programación en escuelas y países en desarrollo.

Es muy popular fuera del campo de la educación, en particular en emulacion de juegos, centros de medios, sistemas embebidos y robotica


Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
BCM2837R1FBG
TN1549 P11
558112-16 N3 W
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
GPIO
DISPLAY
HDMI
CAMERA
HDMI
A/V
USB
USB
ETHERNET
PWR IN
Made in PRC
RUN
ACT PWR


40pins: 28x GPIO, I2C, SPI, UART
1
Wifi / BT
Raspberry Pi 3 Model B V1.2
(c) Raspberry Pi 2015
RUN
2x USB 2.0
2x USB 2.0
microSD slot
on bottom side
Display DSI
CPU/GPU
Broadcom
BCM2837
1024MB SDRAM
4x USB +
Ethernet
controller
LAN9514
H D M I
Camera CSI
Status LED's
PWR ACT
Regulator
Micro
USB
Power in
HDMI out
3.5mm out
Composite
Videotaudio
4 poles
jack
Ethernet
RJ45
Ethernet


HDMI
USB
PWR IN
RUN
802.11n

Raspberry Pi modelo 3B

- Cuatro núcleos ARM Cortex A de 64 bits
- 1GB de RAM
- Lector de Memoria uSD para almacenamiento no volátil
- WiFi, Bluetooth, Ethernet y 4 USB 2.0
- HDMI
- Alimentación por uUSB @ 2.5A
- Sistema operativo Linux, Windows, BSD

Raspbian

- Distribución de Linux, basada en Debian, optimizada para la Raspberry Pi.
- Disponible en versiones con y sin escritorio gráfico (Desktop y Headless server)
- Incluye una utilidad de configuración (Raspiconfig)

Alternativas a Raspbian

- Linux: Ubuntu, Fedora, Arch
- Media center: Open elec, OSMC
- Windows IOT

Instalación de Raspbian en una tarjeta uSD

- Es recomendable que la tarjeta sea categoría 10
- Descargar la imagen (con escritorio o Lite) en archivo zip
- Descomprimir usando 7zip para obtener el archivo img
- Descargar e instalar Etcher
- Grabar el archivo img en la tarjeta uSD usando Etcher
- El usuario por default es pi y pwd: raspberry

Comandos de linux

- man (manual): muestra el manual de usuario de un comando o función de C
- ls (list): muestra el contenido del actual. Modificadores -l muestra los permisos -a muestra archivos ocultos
- cd (change directory): cambia de directorio
- mkdir (make directory): crea un directorio
- mv (move): mueve o renombra un archivo
- cat (concatenate): muestra el contenido de un archivo de texto en la consola, si recibe más de un argumento, concatena los archivos correapondientes
- rm (remove): elimina un archivo. -R elimina un directorio con los archivos que contiene

Comandos de Linux (2)

- rmdir (remove directory): elimina un directorio vacio.
- chown (change owner): cambia el propietario de un archivo o directorio
- chmod (change mode): cambia los permisos de un archivo o directorio
- nano: un simple editor de texto de linea de comandos
- sudo: Ejecuta un comando como super usuario (root) o administrador
- sudo poweroff: apaga la raspberry pi
- sudo reboot: reinicia la raspberry pi

Lenguaje Python

Lenguaje de programación Python

- Python es un lenguaje de programación de alto nivel, interpretado, con variables con tipo de datos dinámico.
- Permite programar usando los paradigmas de programación imperativa, orientada a objetos o funcional
- Se puede extender fácilmente por medio de módulos escritos en C o C++

Historia de Python

- Desarrollado por Guido van Rossum a finales de los años 80, liberado por primera vez en 1991.
- Su diseño pone énfasis en que el código sea legible (import this).
- Python 3 liberado en 2009, no compatible con python 2

Aplicaciones

- Enseñanza de programación (MIT, OLPC)
- Administración de servidores (Google, Yahoo!, Intel)
- Programación web (Google, Apache, IBM)
- Cómputo científico (numpy, scypy)
- Inteligencia artificial (scikit-learn, orange, pandas, chainer)
- Lenguaje de guiones (Gimp, Blender, Ubuntu, Red Hat, LO)

Popularidad

- TIOBE Index: https://www.tiobe.com/tiobe-index/
- IEEEB Spectrum Programming Language Rating: https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

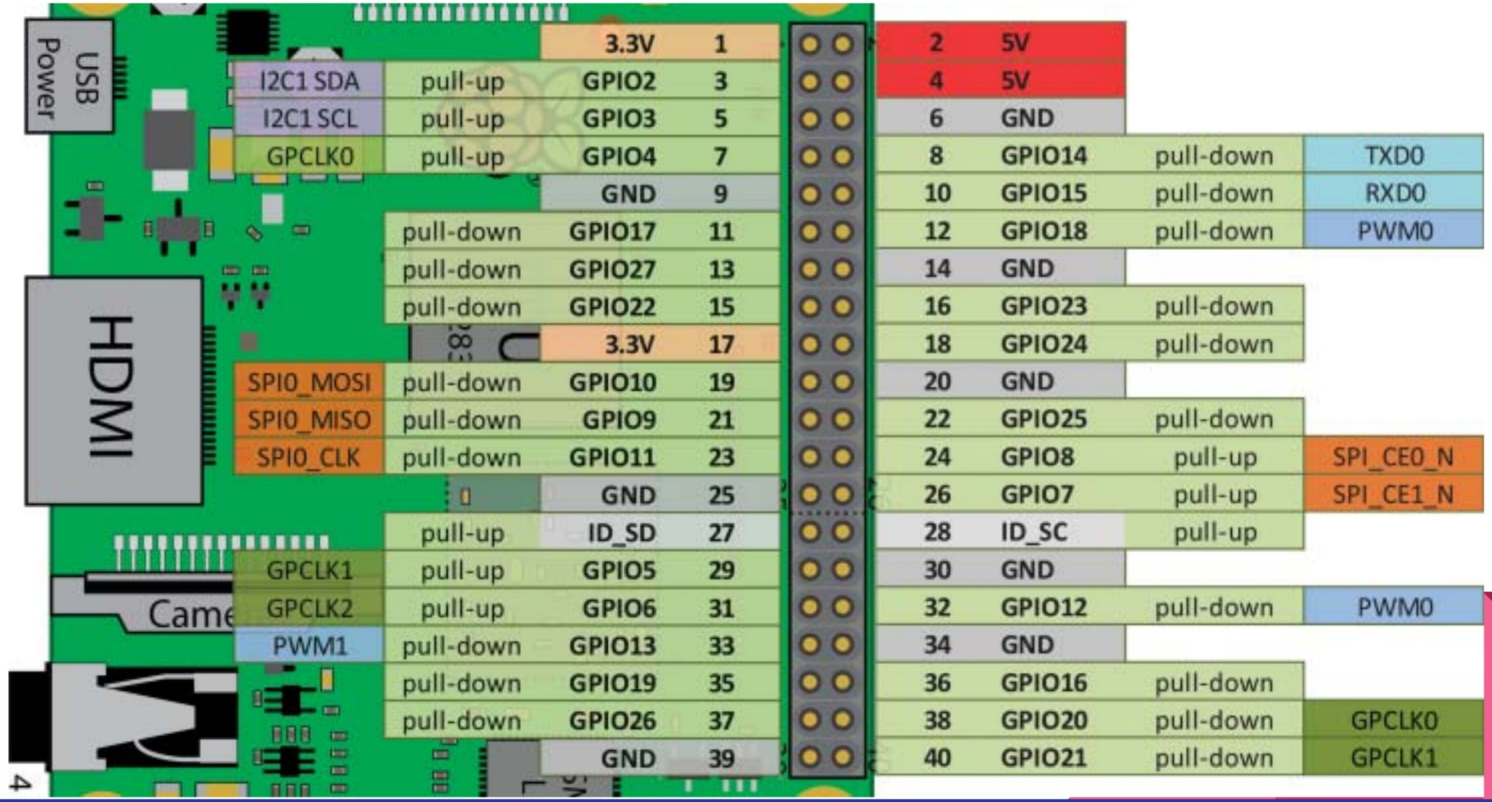
Distribuciones

- Oficial: python.org
- ActivePython: https://www.activestate.com/activepython
- Anaconda: https://www.anaconda.com
- Enthough canopy: https://www.enthought.com/product/canopy/
- WinPython: https://winpython.github.io

Primer programa de ejemplo

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 nombreUsuario=input("¿Cuál es su nombre? ")
5 print("Bienvenido usuario: ", nombreUsuario)
6
```

Conector de expansión para RP1 B+, 2 y 3



Biblioteca para acceso a GPIO

>>> import RPi.GPIO as GPIO #importa la libreria de GPIO
#usar número de terminal no de GPIO
>>>GPIO.setmode(GPIO.BOARD)
#configura como salida en bajo a la terminal 11
>>>GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
>>>GPIO.output(11,GPIO.HIGH) #pone la salida en alto

Módulo time y RPi.GPIO

import time

time.time() regresa el tiempo transcurrido en segundos desde el primero de enero de 1970 como un número de punto flotante

time.sleep() Suspende la ejecución del Script por el tiempo especificado como parámetro (número de segundos expresado como número de punto flotante)

El módulo RPi.GPIO se instala en python 3 con:
sudo apt-get install python3-rpi.gpio

Estructuras de control básicas

Las estructuras de control más comunes son if, if - else y while, que son controladas por una condición lógica,separada del bloque controlado por :

El inicio y el fin de el bloque controlado depende únicamente de la indentación

Ejemplo:

```
if a>=10:
    a=0
    print(a)
print("Hola")
```

Ejemplos de uso de GPIO (Parpadeo)

```
#!/usr/bin/env python
import RPi.GPIO as GPIO #importa el modulo de GPIO de
import time
GPIO.RPI_INFO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
for indice in range(10):
#while True:
    GPIO.output(11, GPIO.HIGH)
    time.sleep(0.5)
    GPIO.output(11, GPIO.LOW)
    time.sleep(0.5)
GPIO.cleanup()
```

Ejemplo de GPIO (Entrada por PushButton)

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(13,GPIO.IN,GPIO.PUD_UP)
GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
while True:
    entrada=GPIO.input(13)
    if entrada == GPIO.LOW:
        print("Se presiono el boton.")
        GPIO.output(11,GPIO.HIGH)
        time.sleep(0.2)
        GPIO.output(11, GPIO.LOW)
        while entrada == GPIO.LOW:
            entrada=GPIO.input(13)
```

Tipos básicos de Pyton

- Importante: Python es sensible a mayusculas y minusculas
- Numericos: int (1 45 -678), float (12.234 -43.56), complex (-1.23+34.9j 56.1-156j)
- Cadenas de caracteres: “ Hola”, ‘Mundo’
- / división de punto flotante: 3/2=1.5
- // división entera: 3//2=1
- type(variable) regresa el tipo de variable

Ejemplo (Método de Newton)

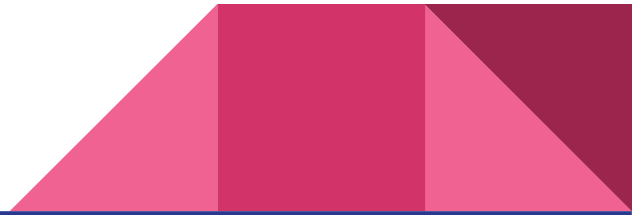
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math

iteracion=0
x0=0.1
f=math.sin(x0)-math.exp(x0)+2
while abs(f)>=0.001:
    iteracion=iteracion+1
    print("i", iteracion)
    df=math.cos(x0)-math.exp(x0)
    x1=x0-(f/df)
    x0=x1
    f=math.sin(x0)-math.exp(x0)+2
    print("x1 ", x1)
    print("f ",f)
```

Cadenas

- + concatena cadenas
- “ se puede escribir directamente en cadenas delimitadas con ’ ‘
- ‘ se puede escribir directamente en cadenas delimitadas con “ “
- indexacion: s1=’Hola’
- s1[0] -> ‘H’
- s1[:2] -> ‘Ho’
- s1[2:] -> ‘la’
- s1[1:3] -> ‘ol’



Estructuras de datos - Listas (1)

- Las listas se usan para agrupar valores de uno o múltiples tipos de datos
- miLista=[‘a’, 2, “b”, 3.5, -6]
- Se indexan para obtener un solo elemento: miLista[0] -> ‘a’
- Se permite la indexación multindice (slicing): miLista[1:3] -> [2,’b’]
- Índices negativos son referidos al final de la lista miLista[-1] -> -6
- Índices vacíos crean una copia de la lista miLista[:]
- Se puede usar un valor de incremento de los índices después del inicio y el fin miLista[0:5:2]
- Se pueden modificar uno o varios miembros de la lista: miLista[1:3]=miLista[-3:-1]

Estructuras de datos - Listas (2)

- len(myList) retorna el número de elementos de la lista
- Se puede añadir al final un elemento con miLista.append(10) -> [‘a’, 2, “b”, 3.5, -6, 10]
- Se puede insertar un elemento en una posición dada con miLista.insert(3,’Hola’)
  [‘a’, 2, “b”, ‘Hola’, 3.5, -6, 10]
- Se puede remover un elemento del final de la lista con miLista.pop() o de una posición en particular con miLista.pop(3)

Estructuras de datos - Tuples

- Similares a las listas, pero no pueden modificarse (son inmutables), por lo que el acceso a ellas es más rápido.
- Se declaran con paréntesis unTuple=(1,’a’,2)
- o sin paréntesis otroTuple=1,2,’a’
- Se pueden indexar del mismo modo que las listas

Estructuras de datos - conjuntos

- Un conjunto (set) es una colección sin orden que no tienen elementos repetidos.
- Se pueden efectuar con ellos las operaciones matemáticas de la teoria de conjuntos
- La función set() convierte una lista en un conjunto eliminando sus elementos duplicados

Estructuras de datos - Diccionarios

- Se usan para almacenar parejas de clave-valor indexadas por las claves.
- Los valores de un diccionario pueden extraerse usando sus claves como índices
- arduinos={‘uno’:328, ‘mega’:2560, ‘lily’:128}
  arduinos[‘lily’] -> ‘128’
  arduinos.keys()

Funciones

Una función de un bloque de código organizado, probado y reutilizable. Permiten que los programas sean modulares.

Python incluye muchas funciones internas, tal como printf, pero también se puede construir funciones propias personalizadas, llamadas *funciones definidas por el usuario*

La definición de una función comienza por la palabra clave def seguida por el nombre de la función y paréntesis

Los parámetros de entrada se colocan entre los parentesis

Funciones (2)

El bloque de codigo dentro de la función comienza con : y va indentado

Opcionalmente, el primer enunciado puede ser la cadena de documentación de la función o la “docstring”

El enunciado return termina la función, regresando de forma opcional un valoral código que llamó a la función

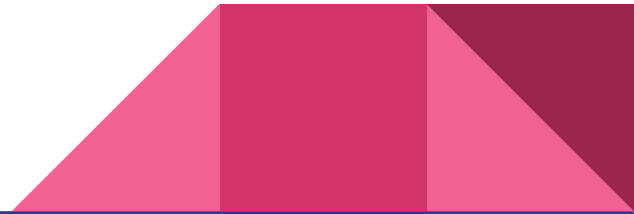
Argumentos de línea de comandos

El modulo sys provee acceso a los argumentos de la línea de comandos por medio de sys.argv

sys.argv es una lista que contiene los argumentos de la línea de comandos

sys.argv[0] es el nombre del programa

len(sys.argv) es el número de argumentos de la línea de comandos



Ejemplo de argumentos de la línea de comandos

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#

def main(args):
    print(args)
    return 0

if __name__ == '__main__':
    import sys
    sys.exit(main(sys.argv))
```

Módulos de Python para interfaces serie

SPI - spidev
Se instala con: sudo apt-get install python3-spidev

I2C - smbus Se instala con: sudo apt-get install python3-smbus

serie - serial https://github.com/pyserial/pyserial
Se instala con sudo apt-get install python3-serial
Documentación: https://pythonhosted.org/pyserial/