

## **Que es Python?**

Es un lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo web.

## **Características del lenguaje:**

### **Propósito general**

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.

### **Multiplataforma**

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

### **Interpretado**

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

### **Interactivo**

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

### **Orientado a Objetos**

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.

### **Funciones y librerías**

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.

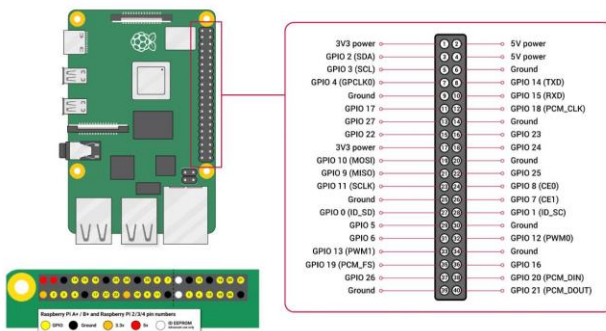
## Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave `begin` y `end`. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

## Programación en Python:

### 1.- Que es GPIO:

Es un conjunto de pines que varían en número según la versión de la Raspberry Pi. De todos modos, la forma de organizarlos siempre es la misma y se distribuyen en dos filas. Cada uno de estos pines tiene una función específica o son de propósito general.



### 2.- Que es setwarnings:

Aunque parece ser un error lo que nos manda al ejecutar nuestro programa, no es un error, sino que te avisa que intentas redefinir como salidas pines que ya estaban definidos previamente y te informa que puedes desactivar estas advertencias con el comando: **`gpio.setwarnings(False)`**.

### 3.- Que es cleanup():

Libera todos los pines que se hayan usado y pone a Falso su estado eléctrico.

### 4.- Que es setmode:

Nos sirve para establecer el modo con el que se va a referir a los pines del Raspberry para esto se utiliza la función de la librería `setmode(GPIO.BOARD | GPIO.BCM)`

### 5.- Que es setup:

Nos sirve para definir uno de los pines del Raspberry como entrada o salida para esto se utiliza la función `setup()` que es muy parecida a la función `pinMode` de Arduino.

```
GPIO.setup(4,GPIO.OUT)
```

```
GPIO.setup(4,GPIO.IN)
```

## 6.- Que es Try:

Para el manejo de excepciones los lenguajes proveen ciertas palabras reservadas, que nos permiten manejar las excepciones que puedan surgir y tomar acciones de recuperación para evitar la interrupción del programa o, al menos, para realizar algunas acciones adicionales antes de interrumpir el programa.

En el caso de Python, el manejo de excepciones se hace mediante los bloques que utilizan las sentencias try, except y finally.

Dentro del bloque try se ubica todo el código que pueda llegar a levantar una excepción, se utiliza el término levantar para referirse a la acción de generar una excepción.

## 7.- Que es while True:

Un bucle while permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición, es decir, mientras la condición tenga el valor True.

La sintaxis del bucle while es la siguiente:

```
while condicion:
```

```
    cuerpo del bucle
```

La ejecución de esta estructura de control while es la siguiente:

Python evalúa la condición:

Si el resultado es True se ejecuta el cuerpo del bucle. Una vez ejecutado el cuerpo del bucle, se repite el proceso (se evalúa de nuevo la condición y, si es cierta, se ejecuta de nuevo el cuerpo del bucle) una y otra vez mientras la condición sea cierta.

Si el resultado es False, el cuerpo del bucle no se ejecuta y continúa la ejecución del resto del programa.

## 8.- Que es la función sleep() del módulo time en Python:

Nos sirve para retrasar la ejecución de ciertas partes del código de nuestro programa, funciona igual que el delay en Arduino.

## Comandos en Linux:

### 1. comando pwd:

Usa el comando pwd para encontrar la ruta del directorio (carpeta) de trabajo actual en el que te encuentras. El comando devolverá una ruta absoluta (completa), que es básicamente una ruta de todos los directorios que comienzan con una barra diagonal (/).

### 2. comando ls:

El comando ls se usa para ver el contenido de un directorio. Por defecto, este comando mostrará el contenido de tu directorio de trabajo actual.

### 3. comando mkdir:

Usa el comando `mkdir` para crear un nuevo directorio: si escribes `mkdir Musica`, creará un directorio llamado `Musica`.

#### **4. comando touch:**

El comando `touch` te permite crear un nuevo archivo en blanco a través de la línea de comando de Linux.

#### **5. comando cd:**

Para navegar por los archivos y directorios de Linux, usa el comando `cd`. Te pedirá la ruta completa o el nombre del directorio, dependiendo del directorio de trabajo actual en el que te encuentres.

#### **6. comando cp:**

Usa el comando `cp` para copiar archivos del directorio actual a un directorio diferente.

#### **7. comando mv:**

El uso principal del comando `mv` es mover archivos, aunque también se puede usar para cambiar el nombre de los archivos.

#### **8. comando rmdir:**

Si necesitas eliminar un directorio, usa el comando `rmdir`. Sin embargo, `rmdir` solo te permite eliminar directorios vacíos.

#### **9. comando rm:**

El comando `rm` se usa para eliminar directorios y el contenido dentro de ellos. Si solo deseas eliminar el directorio, como alternativa a `rmdir`, usa `rm -r`.

#### **10. comando sudo:**

Abreviatura de «SuperUser Do» (SuperUsuario hace), este comando te permite realizar tareas que requieren permisos administrativos o raíz.