

# Editores

Tienes dos opciones, online o local :

## Programar online con <https://python.microbit.org> (recomendado)

Entramos en <https://python.microbit.org/> y el editor online nos permite trabajar ;

1. Una biblioteca de códigos que nos permitirá seleccionar y usar para programar de forma guiada
2. Un simulador para ver cómo se ejecutaría nuestro código
3. Un botón para enviar a la microbit real
4. Botones para guardar nuestro código de forma local y abrir los existentes.

[2024-09-18 22\\_40\\_17-micro\\_bit Python Editor.png](#)

En este curso utilizaremos el editor online microbit.org

## Programar en local con MU

Es un editor muy sencillo, se descarga en <https://codewith.mu/> y permite su instalación en Windows, Linux y Apple.

[2024-07-04 18\\_44\\_27-\(1\) Exploring the Arduino Nano ESP32 \\_ MicroPython & IoT Cloud - YouTube.pr](#)

Fuente <https://codewith.mu/> CC-BY-NC-SA

La primera vez que lo ejecutamos (tarda algo la primera vez) nos pide el **modo** que se puede cambiar en cualquier momento:[2024-09-18 22\\_28\\_05-Configuración.png](#)1 Escribimos el código

2 Lo comprobamos

3 Flasheamos, es decir enviamos el código al Microbit (conectarlo previamente)

4 Cuando sale el mensaje *Código copiado al microbit* **procedemos a resetearlo** para que la placa ejecute el programa.

ATENCIÓN ES IMPORTANTE **RESETEAR LA MICRO:BIT** tienes un botón de **reset** al lado del conector de USB para no estar desconectando y conectando. Una vez reseteado tu programa funcionará.

[2024-09-18 22\\_34\\_58-Configuración.png](#)

```
from microbit import *  
  
while True:  
    display.scroll("Hola Mundo")
```

## OTROS EDITORES DE PYTHON QUE NO SON COMPATIBLES CON PYTHON MICROBIT

Vamos a ver este programa escribo en <https://python.microbit.org/>

```
# Imports go at the top  
from microbit import *  
while True:  
    if pin0.is_touched():  
        display.show(Image.HEART)  
    else:  
        display.show(Image.NO)
```

Lo que hace es :

<https://www.youtube.com/embed/ul2p9HazV1Y>

### **EL MISMO CÓDIGO EN MAKECODE-PYTHON**

Makecode a pesar de que esta orientado a programar con bloques, **tiene su sección de Python**  
[2024-09-23 10\\_37\\_07-Microsoft MakeCode for micro\\_bit.png](#)

Al darle en Python (arriba a la derecha), muestra este código

```
def on_forever():  
    if pins.digital_read_pin(DigitalPin.P0) == 1:  
        basic.show_icon(IconNames.HEART)  
    else:  
        basic.show_icon(IconNames.NO)
```

```
basic.forever(on_forever)
```

Como se puede ver **makecode python no es compatible con <https://python.microbit.org/>** ya lo dice en su tutorial <https://microbit-micropython.readthedocs.io/en/v2-docs/>

[2024-09-23 10\\_46\\_05-BBC micro\\_bit MicroPython documentation — BBC micro\\_bit MicroPython 2 do](#)

## EL MISMO CÓDIGO CON PYTHON DE TINKERCAD

Tinkercad <https://www.tinkercad.com/> es una herramienta estupenda de simulación pues es muy realístico, igual que Maquecode, este muy orientado a la programación en bloques pero también tiene su sección de código python

[2024-09-23 10\\_51\\_49-Circuit design PIN0-CORE-CREW - Tinkercad.png](#)

Si le das la opción de bloque+código intenta muestra los bloques traducidos a código, pero si le das la opción sólo código **pierdes** la programación en bloques, Esto ya lo vimos en <https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/software> en los párrafos escritos en naranja.

El código generado vemos que **no es compatible con Python microbit**

```
# Python code
#

def on_pulsed_p0_high():
    basic.show_icon(IconNames.Heart)
pins.on_pulsed(DigitalPin.P0, PulseValue.HIGH, on_pulsed_p0_high)

def on_pulsed_p0_low():
    basic.show_icon(IconNames.No)
pins.on_pulsed(DigitalPin.P0, PulseValue.LOW, on_pulsed_p0_low)
```

---

Revision #23

Created 2024-09-18 22:16:29 CEST by Javier Quintana

Updated 2025-11-06 14:55:36 CET by Javier Quintana