

## Contenidos:

Introducción	7
1. Mostrar texto en los ledes de micro:bit Programas: 1a, 1b	11
2. Contar con micro:bit. Las instrucciones <b>while</b> y <b>for</b> Programas: 2a, 2b, 2c	13
3. Utilizar los botones de micro:bit. La instrucción <b>if</b> . Control de led RGB con los botones de micro:bit. Programas: 3a, 3b	17
4. Mostrar imágenes con los ledes de micro:bit. Programas: 4a, 4b, 4c, 4d, 4e	23
5. Listas. Imágenes preinstaladas en micro:bit. Programas: 5a, 5b, 5c, 5d	29
6. Música y sonido con micro:bit . Programas: 6a, 6b, 6c, 6d, 6e, 6f, 6g	33
7. Funciones. Mostrar números y niveles en los ledes. Programas: 7a, 7b, 7c	43
8. Brújula, campo magnético y temperatura. Reutilización de una función. Programas: 8a, 8b	53
9. Radio. Utilización la radio y el acelerómetro. Programas: 9a, 9b (2 programas)	57
10. Proyecto final: sensor de temperaturas utilizando dos micro:bit y la radio. Archivos de datos. Elaboración de gráficas. Elaboración de un archivo de texto en micro:bit. Diccionarios en Python. La sucesión de Fibonacci. Programas 10 a (4 programas) y 10b.	61

# 1

## Mostrar texto en los led

En este primer capítulo aprenderemos a mostrar un mensaje de texto a través de los 25 ledes de que dispone micro:bit.

A lo largo de todo el libro, el código de Python se mostrará con este formato, indicando el nombre del programa.

### Programa 1a

```
from microbit import *
display.scroll("¡Hola!")
```

Puedes modificar este sencillo código sustituyendo el "¡Hola!" por el texto que prefieras. Además, podemos hacer que el mensaje se repita continuamente modificando la última línea:

### Programa 1b

```
from microbit import *
display.scroll('¡Hola!', wait=False, loop=True)
```

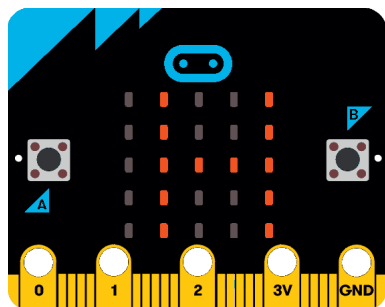
Escribe el código de Python en el editor online (<http://python.microbit.org/v/1>) y haz clic en *Download* para descargar el archivo con extensión .hex que utiliza micro:bit. Puedes guardar este archivo en tu ordenador.

Conecta la micro:bit a un puerto USB del ordenador mediante el cable, y copia el archivo *microbit.hex* en micro:bit. El led de la parte trasera de micro:bit parpadeará mientras se realiza la

transferencia del archivo. Una vez que el archivo se haya descaragado en micro:bit, el texto que hemos escrito en la instrucción comenzará a aparecer en sus ledes.

Si hay algún error en el código, por ejemplo, no has cerrado las comillas o los paréntesis, micro:bit mostrará un mensaje de error en sus ledes, indicando en qué línea de código está el error.

micro:bit no muestra caracteres de otros idiomas distintos del inglés, como por ejemplo *¡¿ñÑçÇ*.



En lugar del editor online, quizá prefieras utilizar el editor Mu, que puedes descargar gratuitamente en la dirección <https://codewith.mu/>. En este caso haz clic en *Flash* para descargar el archivo con extensión .hex a la micro:bit.

Si abres el archivo .hex con un editor de texto, puedes ver que las instrucciones de nuestro programa en Python se han convertido en una serie de letras y números para micro:bit:

```
:020000040000FA
:1000000000400020ED530100295401002B54010051
:1000100000000000000000000000000000000000E0
```

# 3

## La instrucción *if*

En ocasiones, necesitaremos que nuestro programa “decida” qué hacer, en función del resultado de una operación, de si pulsamos el botón A o B, o del valor de la señal que micro:bit reciba desde otro dispositivo electrónico. Para ello, Python dispone de la instrucción `if`.

El siguiente programa es un ejemplo de utilización de la instrucción `if` para detectar qué botón pulsamos en micro:bit.

### Programa 3a

```
from microbit import *
while True:
    if button_a.is_pressed() and button_b.is_pressed():
        display.scroll("AB")
        display.scroll("adios!", delay=80)
        break
    elif button_a.is_pressed():
        display.scroll("A")
    elif button_b.is_pressed():
        display.scroll("B")
    else:
        display.show("?")
```

Este programa es más largo que los que hemos visto hasta ahora, pero si te fijas verás que consiste también en un bucle `while`. La diferencia es que dentro del bucle se ejecutan más instrucciones que en el programa anterior, y en una nueva instrucción: la instrucción `if`. Además, este programa detecta si pulsamos un botón de micro:bit, y realiza una acción distinta según qué botón pulsemos.

# 4

## Imágenes con los led de micro:bit

Hemos visto cómo micro:bit permite mostrar texto en sus ledes. Se puede mostrar información mediante los ledes de micro:bit de otra manera: podemos “dibujar” una imagen con los 25 ledes mediante un programa.

Disponemos de 5 filas de 5 puntos cada una para dibujar nuestra imagen. Además, podemos asignar la intensidad de cada led con un valor entre 0 (apagado) y 9 (máxima intensidad). El siguiente programa muestra una figura con los 25 ledes y diferentes intensidades, formando un degradado en diagonal con los ledes..

### Programa 4a

```
from microbit import *  
figura = Image("56789:"  
               "45678:"  
               "34567:"  
               "23456:"  
               "12345")  
display.show(figura)
```

La figura del programa anterior también se podría indicar con una sola línea de código:

```
figura = Image("56789:45678:34567:23456:12345")
```

Cada fila de ledes se representa con una cadena de texto de 5 números entre 0 (apagado) y 9 (encendido al máximo).