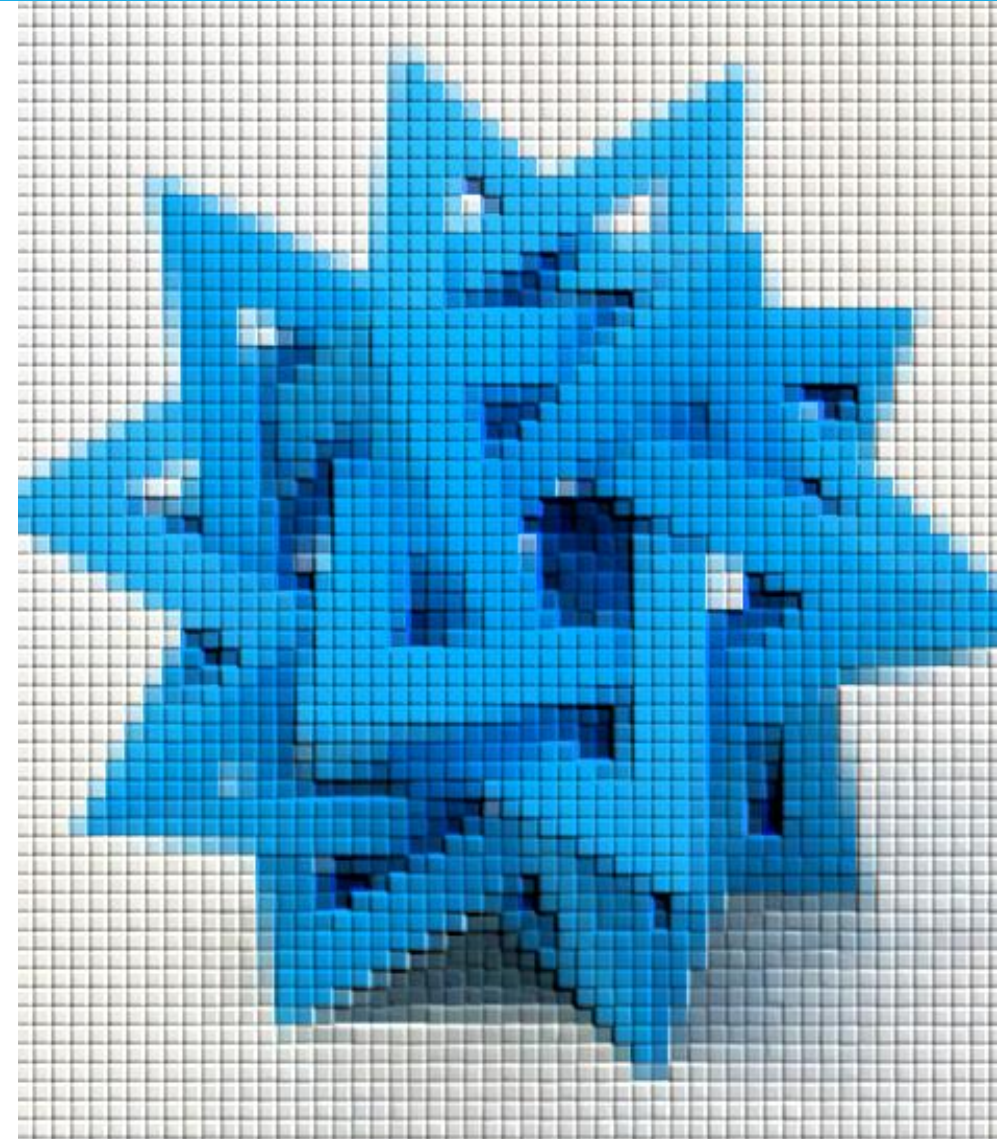


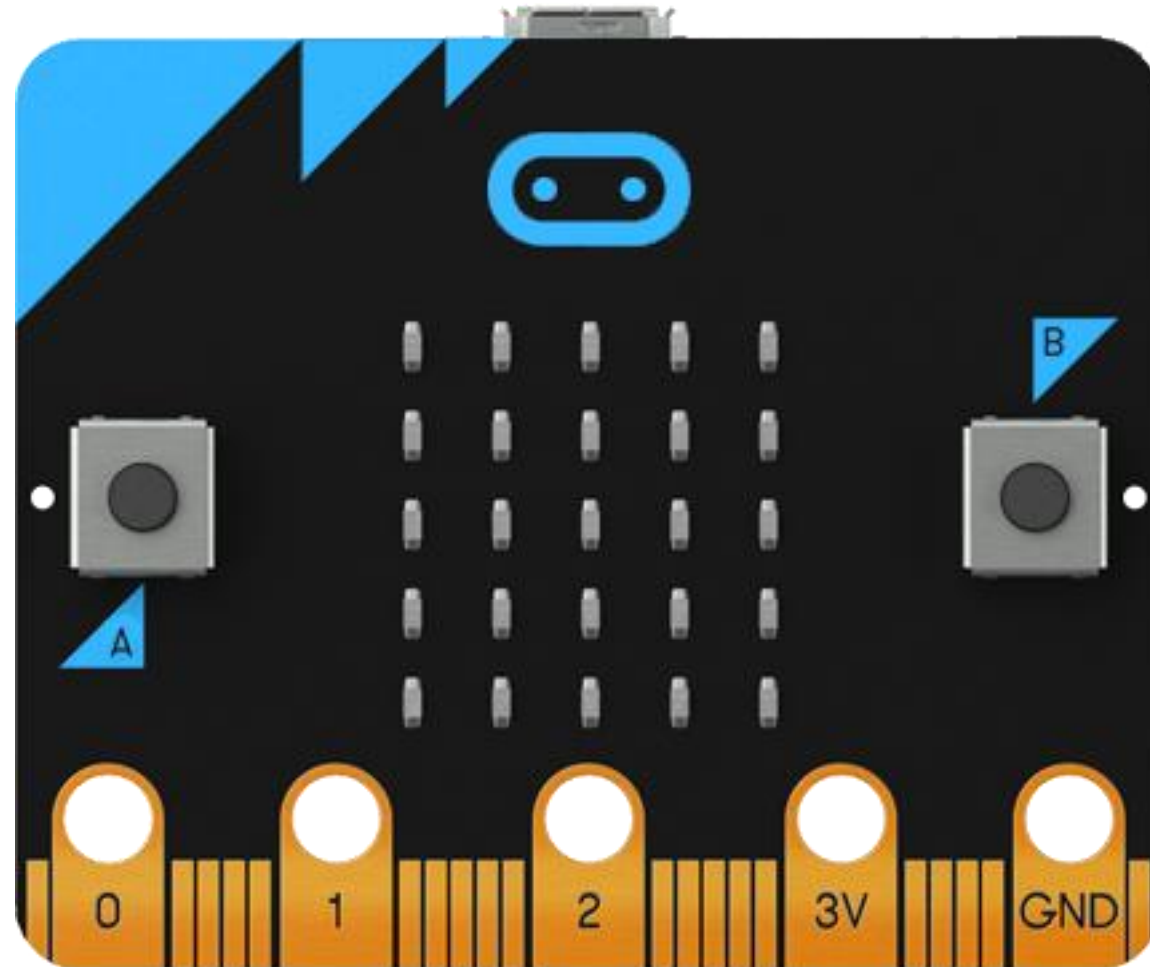
from python import ...more

Micro-giochi

ESPLORAZIONI IN PYTHON PER I PIÙ CURIOSI

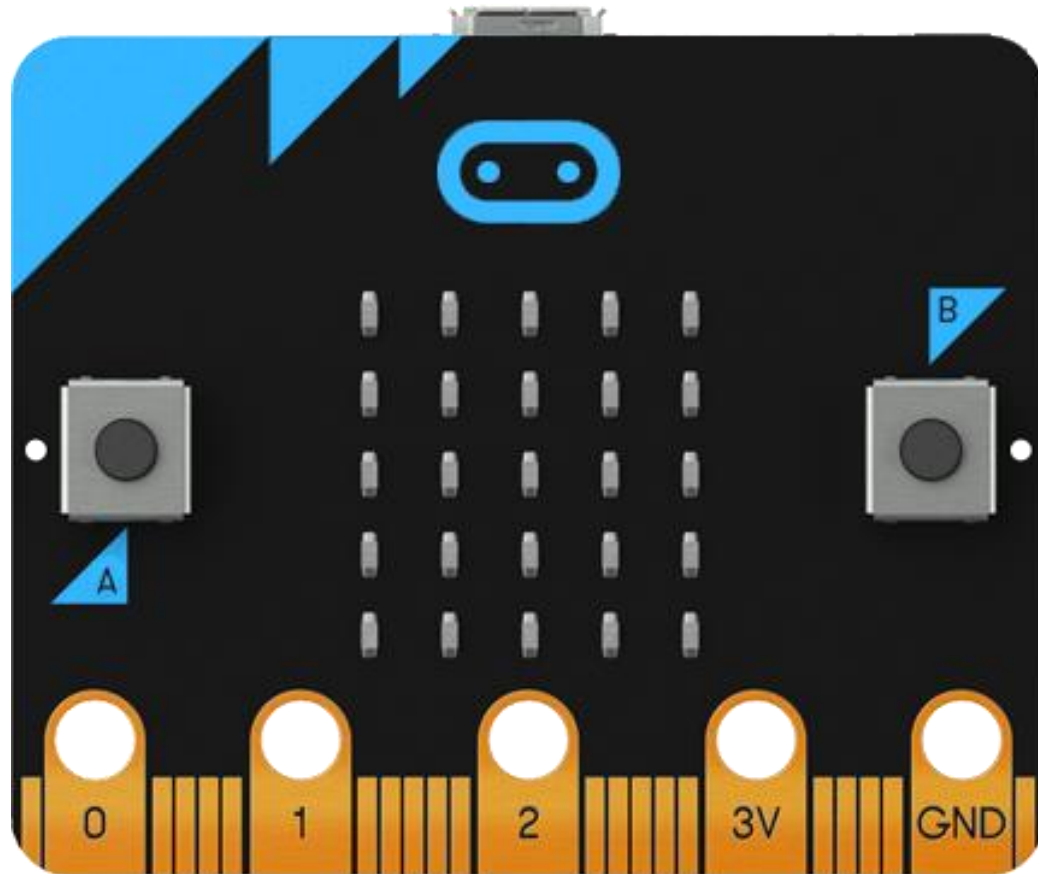


micro:bit



<https://microbit.org/>

micro:bit



- Inputs and outputs
- LEDs
- Buttons
- Radio
- Accelerometer
- Compass
- Light sensor
- Temperature sensor
- Pins
- Sound
- USB interface
- Processor

<https://microbit.org/>

MicroPython & PyBoard



MicroPython FORUM DOCS QUICK-REF DOWNLOAD STORE CONTACT

MicroPython

MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised to run on microcontrollers and in constrained environments.

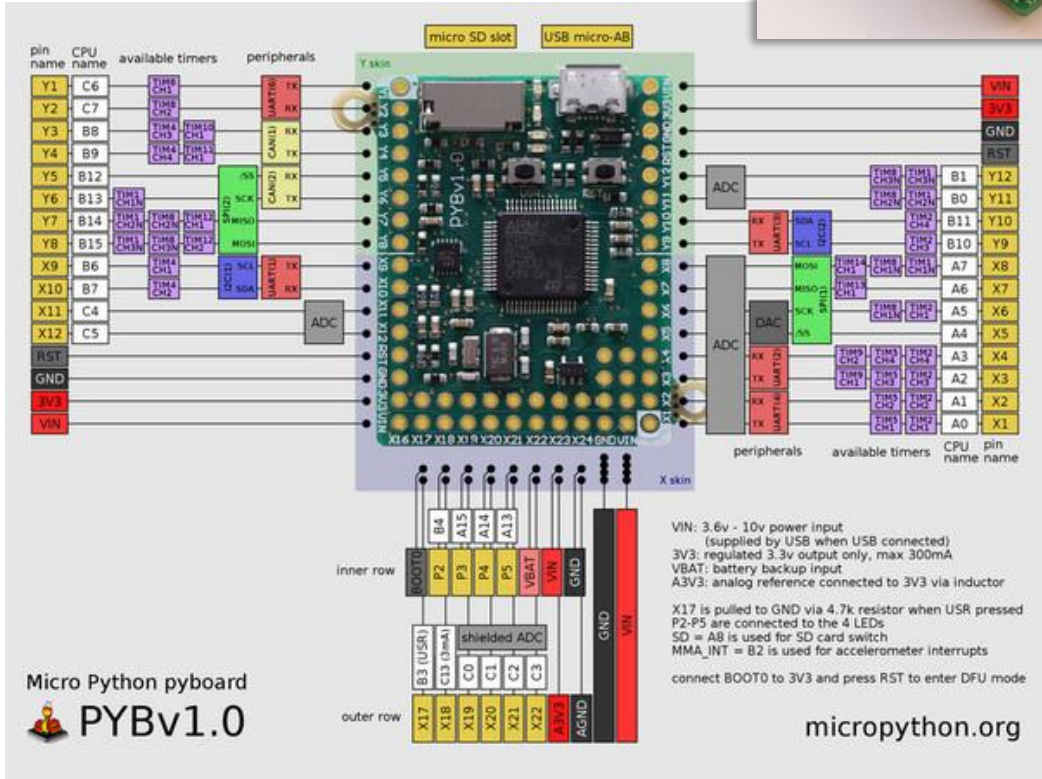
The MicroPython *pyboard* is a compact electronic circuit board that runs MicroPython on the bare metal, giving you a low-level Python operating system that can be used to control all kinds of electronic projects.

MicroPython is packed full of advanced features such as an interactive prompt, arbitrary precision integers, closures, list comprehension, generators, exception handling and more. Yet it is compact enough to fit and run within just 256k of code space and 16k of RAM.

MicroPython aims to be as compatible with normal Python as possible to allow you to transfer code with ease from the desktop to a microcontroller or embedded system.

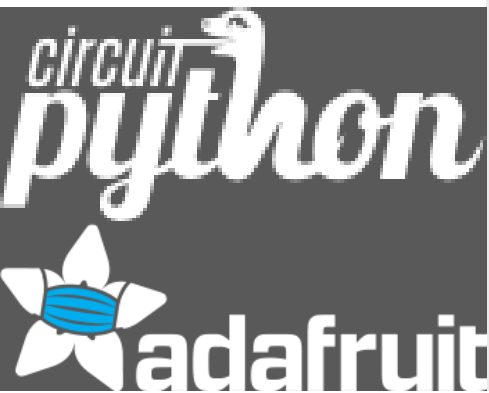
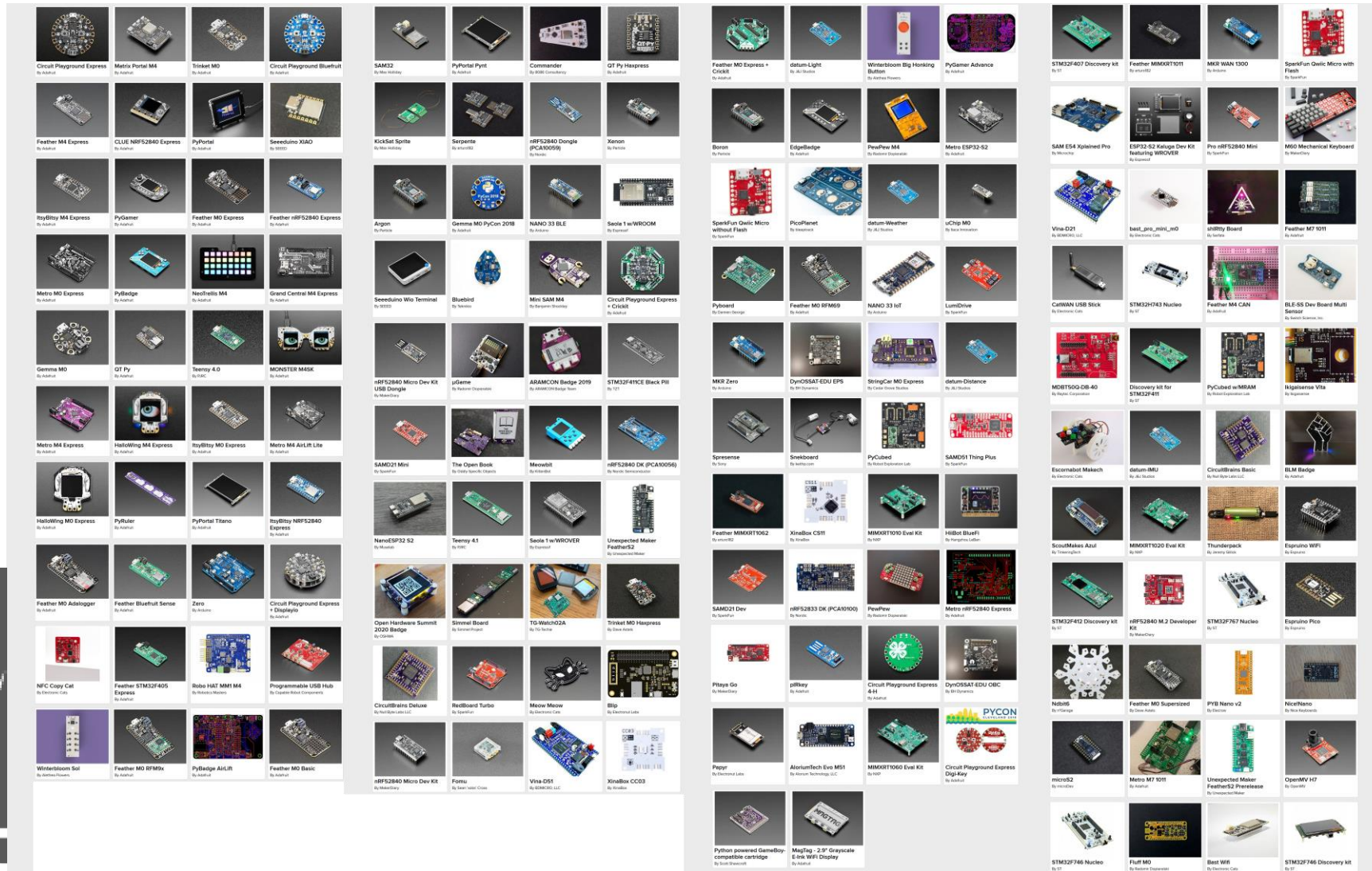
TEST DRIVE A PYBOARD BUY A PYBOARD USE MICROPYTHON ONLINE

Proper Python with hardware-specific modules



<http://micropython.org/>

CircuitPython - <https://circuitpython.org/>



BBC micro:bit MicroPython

The screenshot shows the top navigation bar with the title "BBC micro:bit MicroPython" and version "v1.0.1". Below it is a search bar labeled "Search docs". A dark sidebar on the left contains a "TUTORIALS" section with links to Introduction, Hello, World!, Images, Buttons, Input/Output, Music, Random, Movement, Gestures, Direction, Storage, Speech, Network, Radio, and Next Steps. Below that is an "API REFERENCE" section with links to micro:bit MicroPython API, Microbit Module, Accelerometer, Audio, Bluetooth, Buttons, Compass, and Display. At the bottom of the sidebar are "Read the Docs" and "v: v1.0.1" buttons.

Docs » BBC micro:bit MicroPython documentation

[Edit on GitHub](#)

BBC micro:bit MicroPython documentation

Welcome!

The BBC micro:bit is a small computing device for children. One of the languages it understands is the popular Python programming language. The version of Python that runs on the BBC micro:bit is called MicroPython.

This documentation includes lessons for teachers and API documentation for developers (check out the index on the left). We hope you enjoy developing for the BBC micro:bit using MicroPython.

If you're a new programmer, teacher or unsure where to start, begin with the tutorials.

The comic strip is titled "First Steps with MicroPython" by Mike Rowbitt. It consists of three panels. The first panel shows a man with a speech bubble that says "MicroPython was created by Damien...". The second panel shows a BBC micro:bit with a speech bubble that says "It works with the BBC micro:bit.". The third panel shows a yellow character with a speech bubble that says "Everything you need to know about MicroPython on the BBC micro:bit is found in this documentation." Below the comic is a small text: "Generated by Python Comics: MAKE YOUR OWN".

To get involved with the community subscribe to the microbit@python.org mailing list (<https://mail.python.org/mailman/listinfo/microbit>).

Note

This project is under active development. Please help other developers by adding tips, how-tos, and Q&A to this document. Thanks!

<https://microbit-micropython.readthedocs.io>

Documentazione

The screenshot shows the 'User guide' page for Python on the BBC micro:bit website. The page has a dark header with the 'micro:bit' logo and navigation links like 'Get started', 'Projects', 'Lessons', 'Let's code', 'Impact', 'Buy', and 'News'. Below the header, there are sub-navigation links: 'First steps', 'User guide' (which is highlighted), 'BBC micro:bit in school', and 'Home learning'. The main content area is titled 'User guide' and includes a sidebar with a table of contents: Overview, Features in depth, Remote teaching, Safety, and Guide to mobile apps. Under 'Python guide', there are sections for 'Why Python?' (explaining its use in education and machine learning), 'The micro:bit Python editor' (describing an online editor for teachers and learners), and 'Hello, World!' (introducing the first steps).

This screenshot shows the 'BBC micro:bit MicroPython v1.0.1' documentation page. It features a search bar at the top and a dark sidebar with a table of contents. The 'TUTORIALS' section includes links for Introduction, Hello, World!, Images, Buttons, Input/Output, Music, Random, Movement, Gestures, Direction, Storage, Speech, Network, Radio, and Next Steps. The 'API REFERENCE' section lists links for the MicroPython API, Microbit Module, Accelerometer, Audio, Bluetooth, Buttons, Compass, and Display. At the bottom, there is a 'Read the Docs' button.

This screenshot shows the 'BBC micro:bit MicroPython documentation' page. It includes a search bar and a 'Edit on GitHub' link. The main heading is 'BBC micro:bit MicroPython documentation'. Below this, there is a 'Welcome!' section followed by a paragraph explaining that the BBC micro:bit is a small computing device for children and that the documentation covers lessons for teachers and API documentation for developers. A 'Note' section at the bottom states that the project is under active development and encourages users to help by adding tips, how-tos, and Q&A. A 'First Steps with MicroPython' section by Mike Rowbit is also visible, featuring a photo of a man and a code snippet: `from microbit import * # Build your code here! display.scroll("Hello, world!")`.

<https://microbit.org/get-started/user-guide/python/>

<https://microbit-micropython.readthedocs.io/>

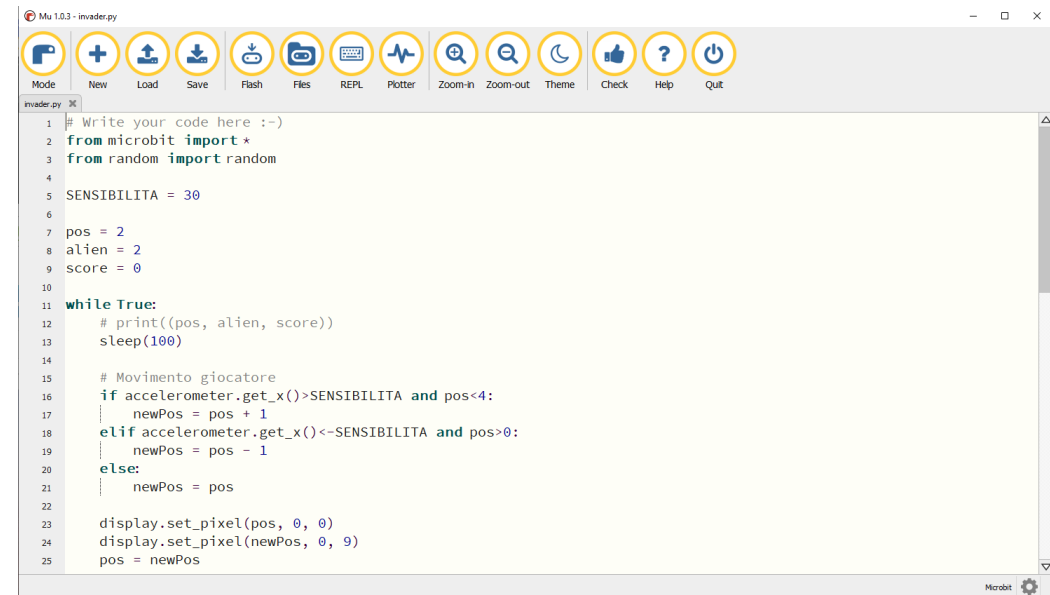
Editor

ON-LINE IN UN BROWSER
(EDITOR LIMITATO)



<https://python.microbit.org/>

MU: IDE INSTALLATO
(WIN/MAC/LINUX/RASPBERRY)



<https://codewith.mu/>



Struttura base

```
from microbit import *
```



```
# Codice di inizializzazione
```

```
# (eseguito una volta all'inizio)
```

```
while True:
```

```
    # Corpo principale del programma
```

```
    # (eseguito in loop all'infinito)
```

Esempio

```
from microbit import *  
  
while True:  
    if button_a.is_pressed():  
        display.show(Image.HAPPY)  
    if button_b.is_pressed():  
        display.show(Image.SAD)
```



Principali oggetti e metodi

■ display

- `.scroll(string)`
- `.show(Image.IMGNAME)` o `.show(Image(hexString))`
- `.read_light_level()`
- `.clear()`
- `.set_pixel(x, y, value)`

■ button_a, button_b

- `.is_pressed()`
- `.get_presses()`

■ sleep(ms)

■ compass

- `.calibrate()`
- `.heading()`

■ accelerometer

- `.was_gesture('gesture name')`
- `.current_gesture()`
- `.get_x()`, `.get_y()`, `.get_z()`

■ temperature()

■ radio

- `.config(group=123, channel=7)`
- `.on()` / `.off()`
- `.receive()`
- `.send('msg')`

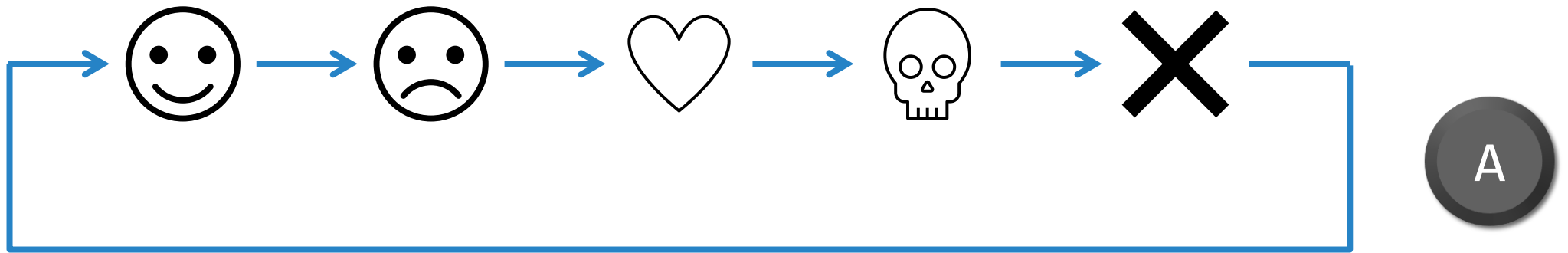
Esempio 1

- Quando l'utente preme il bottone A
 - mostrare una faccia felice
- Quando l'utente preme il bottone B
 - mostrare una faccia triste
- Quando l'utente preme entrambi i bottoni
 - cancellare il display



Secret message: passo 1

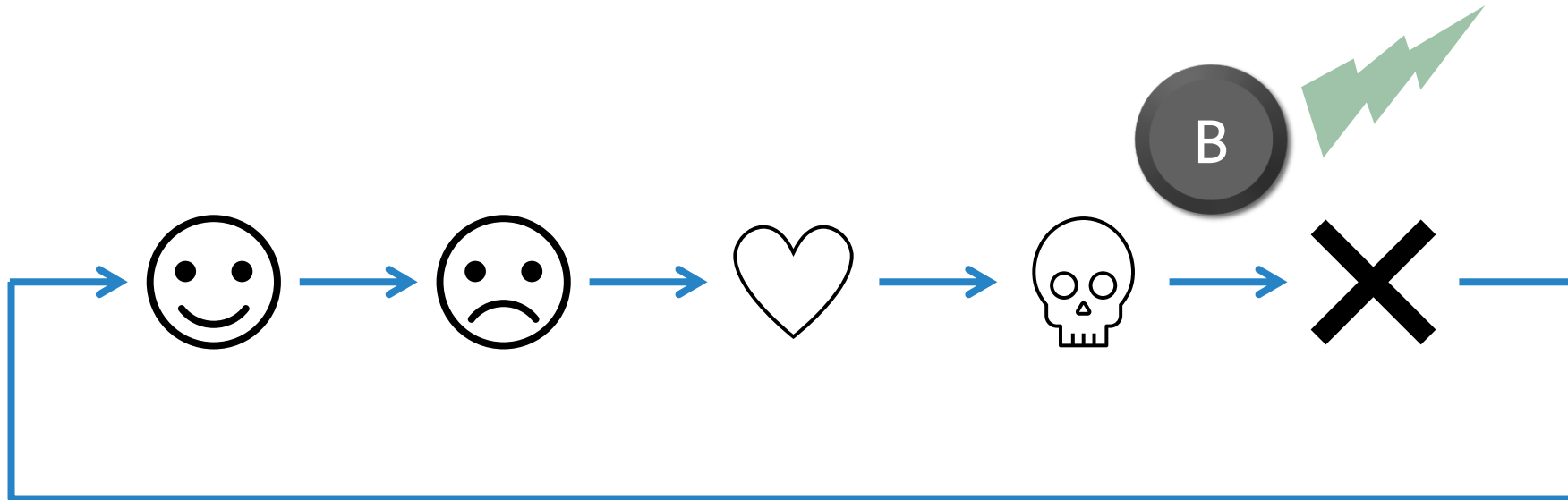
- Ogni volta che l'utente preme il tasto **A**, modificare l'immagine visualizzata, iterando ciclicamente tra 5 immagini a vostra scelta
- Esempio:



- Quando si «**scuote**» il micro:bit, cancellare il display

Secret message: passo 2

- Quando l'utente preme il tasto **B**, il simbolo attualmente visualizzato viene spedito *via radio* ad un altro micro:bit
- Alla ricezione del simbolo, il secondo micro:bit lo visualizzerà



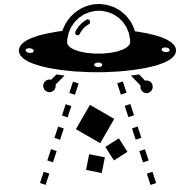
Alien Destroyer: passo 1

- Illuminare un pixel in alto al centro
 - È la nostra astronave!
- Quando si inclina il micro:bit a destra, spostare il pixel di una colonna a destra
- Quando si inclina il micro:bit a sinistra, spostare il pixel di una colonna a sinistra



Alien Destroyer : passo 2

- Quando l'utente preme A, l'astronave «spara» un raggio laser verso il basso
- Illuminare tutti i pixel sotto la stessa verticale dell'astronave
- Giocare con la tempistica e l'illuminazione per ottenere un effetto realistico



Alien Destroyer : passo 3

- Creare un «nemico»
- Il nemico vive sulla riga più in basso del display
- Ad ogni passo, il nemico sta fermo, oppure si sposta a destra o sinistra in modo random
 - Tarare le probabilità in modo da avere un movimento non troppo lento e non troppo rapido



Alien Destroyer : passo 4

10

- Quando il nostro raggio laser colpisce l'alieno, guadagniamo un punto
 - Visualizzare il punteggio
- Quando il nostro raggio laser manca l'alieno, il punteggio viene azzerato
 - Visualizzare un'icona negativa
- Quanti punti riuscite a fare?