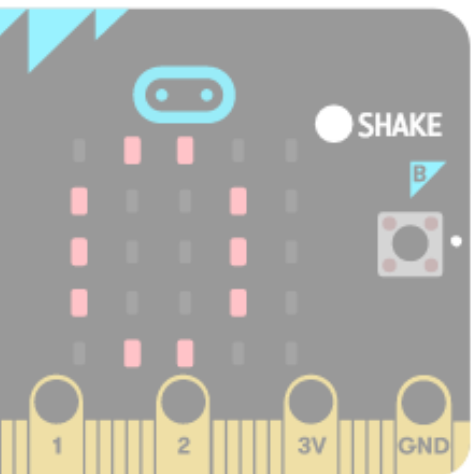


Microbit

<http://microbit.org/fr/>

Microbit

- designé par la BBC
- programmation textuelle (python) / graphique
- programmation online / offline
- code / électronique
- synthèse sonore / vocale
- accéléromètre



Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

Advanced

Programmation graphique

on start

show string " Hello! "

show number 0

on shake

clear screen

if Random = 2

then show string " YES "

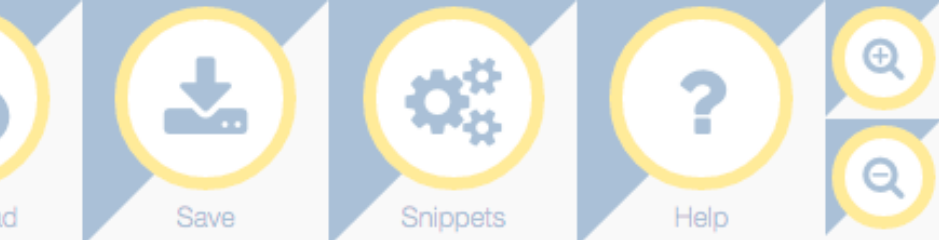
else if Random = 1

then show string " NO "

else show string " I DON'T KNOW "

show number 8

Getting Started



Add your Python code here. E.g.

```
from microbit import *  
import music
```

```
notes = [  
    'c4:1', 'e', 'g', 'c5', 'e5', 'g4', 'c5', 'e5', 'c4', 'e'
```

Programmation textuelle

```
while True:
```

```
    display.scroll('Hello, World!')  
    display.show(Image.HEART)  
    sleep(2000)  
    music.play(notes)
```

Code With Mu

Mu is a simple code editor for beginner programmers. It's written in Python and works on Windows, OSX, Linux and Raspberry Pi.

Download now

Help

Code with MU

<https://codewith.mu/>

Less is More

Mu has only the most essential features, so users are not intimidated by a baffling interface.

Keep it Simple

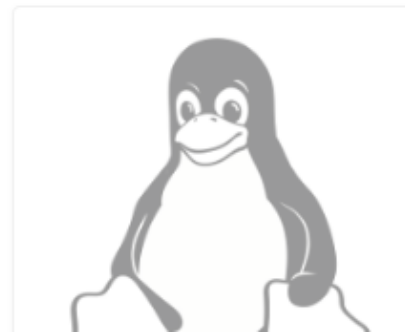
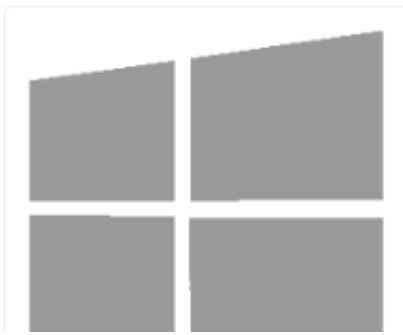
It's quick and easy to learn Mu ~ complexity impedes a novice programmer's first steps.

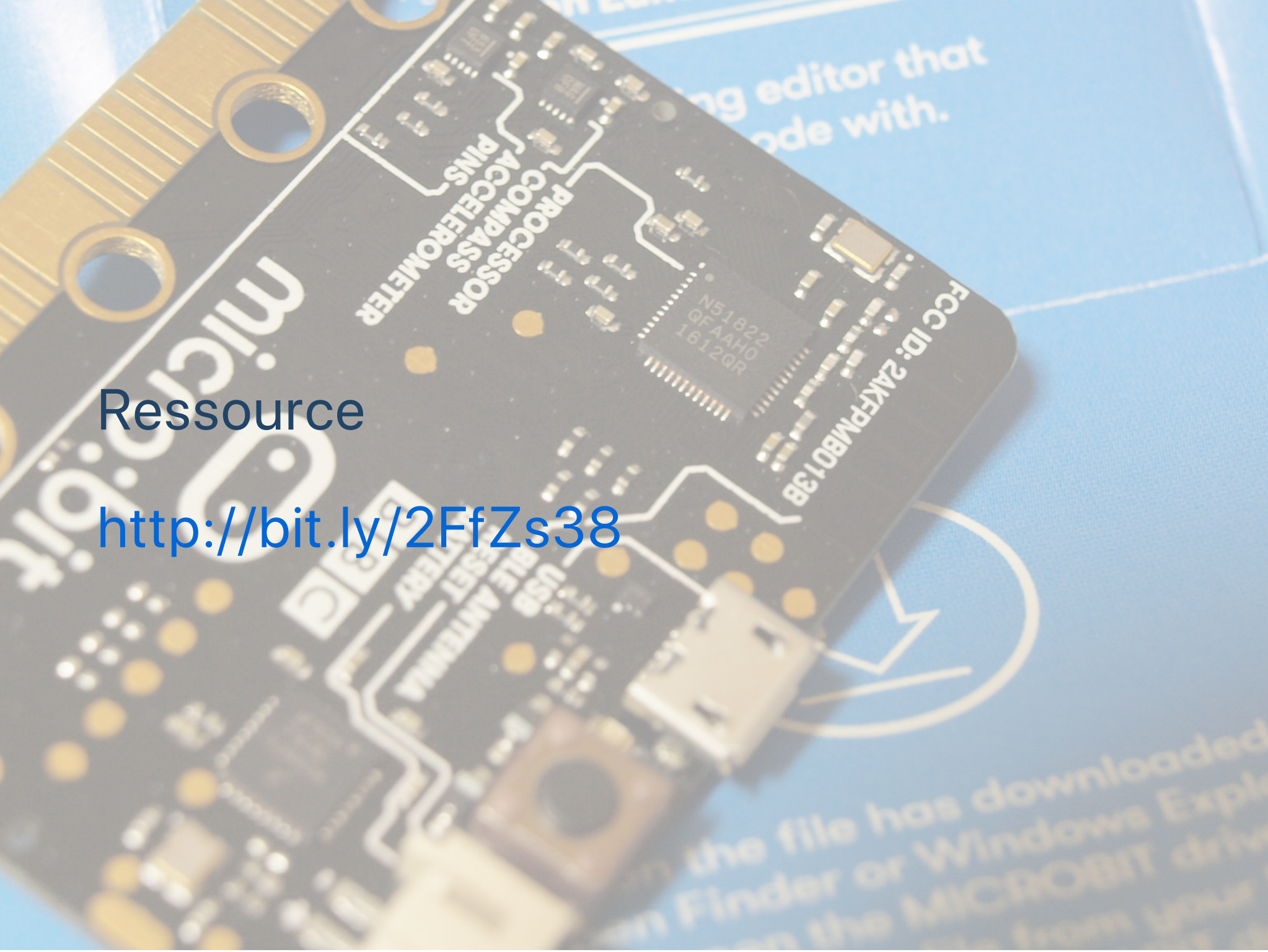
Path of Least Resistance

Whatever the task, there is always only one obvious way to do it with Mu.

Have fun

Learning should inspire fun ~ Mu helps learners quickly create and test working code.





Ressource

<http://bit.ly/2FfZs38>

```
... SavedState[] newArray(int size) {  
    return new SavedState[size];  
}
```

Hello World !

```
... TabHost(Context context) {  
    // Note that we call through to the version that takes a AttributeSet  
    // because the simple Context constructor can result in a NullPointerException  
    super(context, null);  
}
```

Bonjour à tous !

```
from microbit import *  
display.scroll("Hello, World!")
```

```
public FragmentTabHost(Context context, AttributeSet attrs) {  
    super(context, attrs);  
    initView(context, attrs);  
}
```

Images

Utilisation de la matrice de leds pour créer les images

```
from microbit import \*  
display.show(Image.HAPPY)
```

Exemples d'images

```
Image.HEART  
Image.HAPPY  
Image.COW  
Image.XMAS  
Image.PACMAN
```


Liste d'images

```
from microbit import *  
display.show(Image.ALL_CLOCKS, loop=True, delay=100)
```

Créer ses propres images

```
from microbit import *  
  
boat = Image("05050:"  
             "05050:"  
             "05050:"  
             "99999:"  
             "09990")  
  
display.show(boat)
```

de 0 à 9 -> luminosité

Boucler un programme / créer un badge

```
# création d'un badge

while True:
    display.show(Image.HEART)
    sleep(2000)
    display.scroll("Hello world !")
    sleep(50)
    display.show(Image.HAPPY)
    sleep(2000)
```

Makey Makey like

```
from microbit import *

while True:
    if pin0.is_touched():
        display.show(Image.HAPPY)
    else:
        display.show(Image.SAD)
```

Musique

Connecter un buzzer / piezzo / haut-parleur aux bornes GND et 0, 1 ou 2

```
import music
music.play(music.NYAN)
```

Exemples

- music.PRELUDE
- music.NYAN
- music.BIRTHDAY
- music.PYTHON

Créer une mélodie

```
import music

tune = ["C4:4", "D4:4", "E4:4", "C4:4", "C4:4", "D4:4",
        "E4:4", "C4:4", "E4:4", "F4:4", "G4:8", "E4:4",
        "F4:4", "G4:8"]

music.play(tune)
```

Syntaxe

note / octave : durée

Silence : "R:4" (R = rest).

Utiliser l'accéléromètre

- X : de gauche à droite
- Y : d'avant vers l'arrière
- Z : de haut en bas

Exemple pour générer un son

```
from microbit import *  
import music  
  
while True:  
    music.pitch(accelerometer.get_y(), 10)
```

Utiliser les boutons

```
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.HAPPY)
    elif button_b.is_pressed():
        break
    else:
        display.show(Image.SAD)

display.clear()
```


Synthèse vocale

```
import speech  
speech.say("Hello, world")
```

Paramètres de la synthèse vocale

- speed : vitesse de la voix (0-255)
- pitch : voix grave ou aigüe (0-255)
- throat : profondeur (0-255)
- mouth : ouverture de la bouche (0-255)

Par exemple :

```
import speech

speech.say("Hello, world", speed=120, pitch=100,
throat=100, mouth=200)
```

UP|CYCLE

Merci !

COMMONS

jfcauche@gmail.com / [@jeffakakaneda](#)

upcyclecommons.com