

# Creatief programmeren met Python & micro:bit

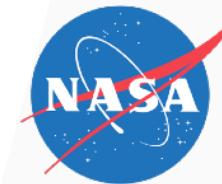
Tim Molderez & Mathijs Saey



# Wat gaan we doen?

- Introductie tot programmeren met een volledige

programmeertaal:  python™



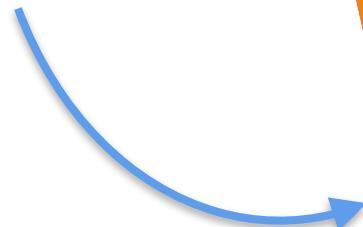
- Introductie tot *embedded systemen*:

- computer om knoppen, lampen, sensors, motors, machines, enz. aan te sturen m.b.v. software



# Wat is het doel?

- Er is geen doel!
- **Have fun!**

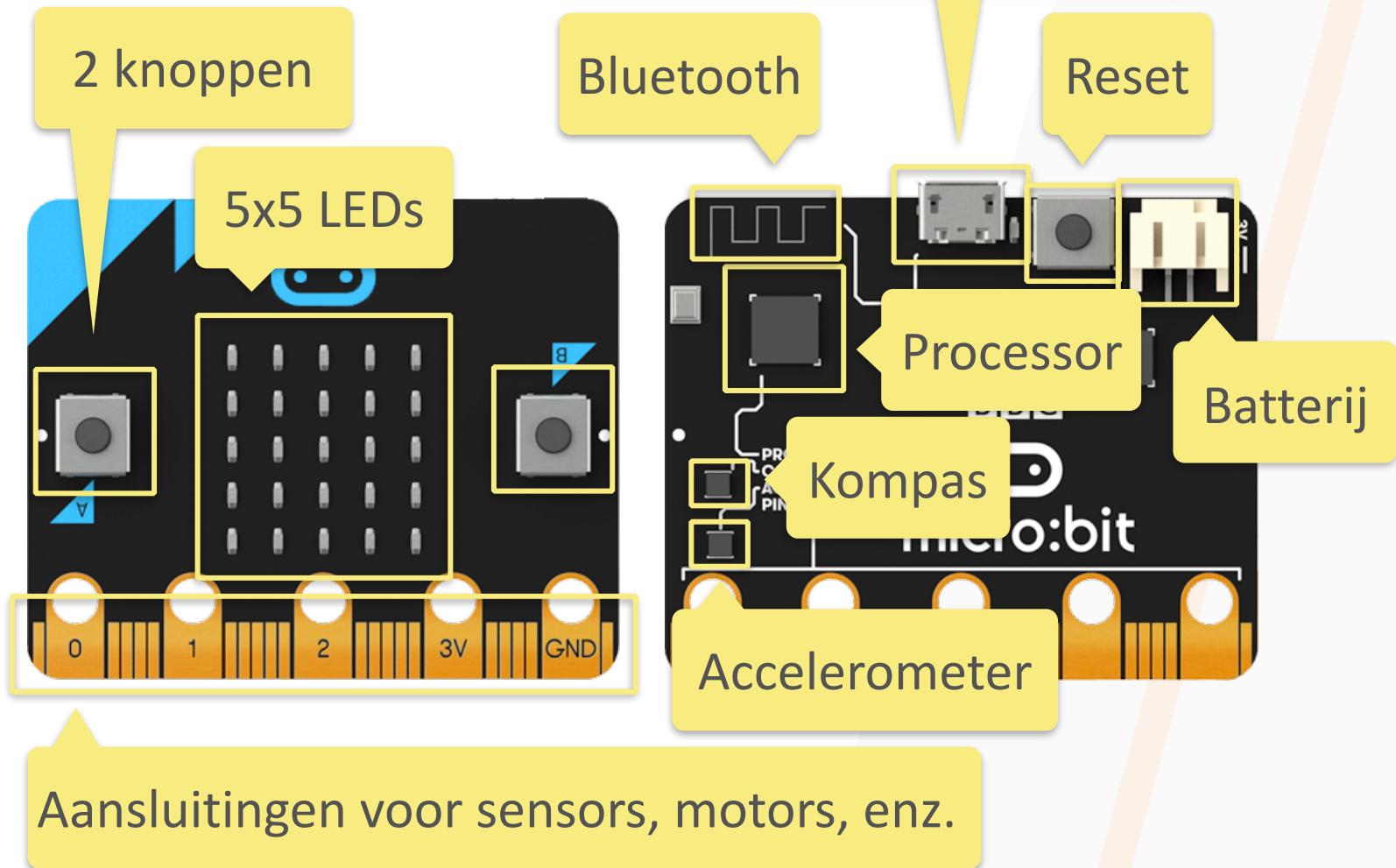


- (Sorry als je per ongeluk iets bijleert :))



Documentatie: <http://timmolderez.be/microbit>

# micro:bit



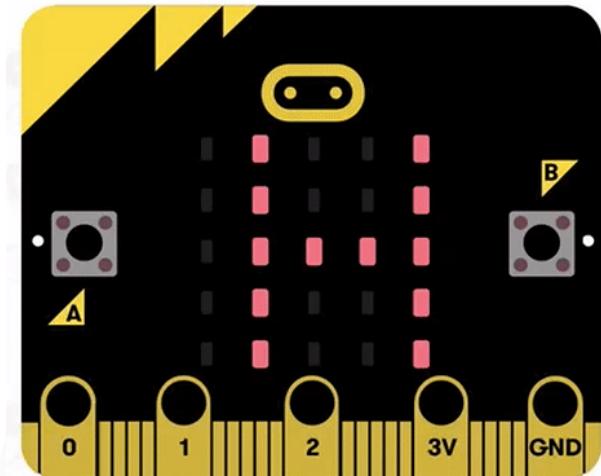
# Mijn eerste programma

```
from microbit import *\n\ndisplay.scroll("Hello!")
```

Toont Hello! op het scherm

Opent de "vocabulaire" om de basisfunctionaliteit van de micro:bit aan te kunnen spreken

Maybe add a sneak peek slide before this one, to show what kind of stuff students'll be able to make at the end.



# Het programma uitvoeren

1. Sluit de micro:bit aan met de USB-kabel
2. Open de "Mu" programmeeromgeving 
3. Geef de programmacode in:

```
from microbit import *
display.scroll("Hello!")
```
4. Klik op "Flash" om het programma te uploaden
5. Na enkele sec. zal het programma starten



# Meerdere instructies

```
from microbit import *  
display.scroll("Hello!")  
sleep(3000) ← Wacht 3000 millisec.  
display.scroll("Bye :))")
```

# Waardes onthouden

```
from microbit import *  
name = "Joske"
```

Vanaf nu is variabele `name` gelijk aan de waarde Joske

```
display.scroll("Hello, ")
```

```
display.scroll(name)
```

Gebruik de huidige waarde van `name`

In het algemeen:

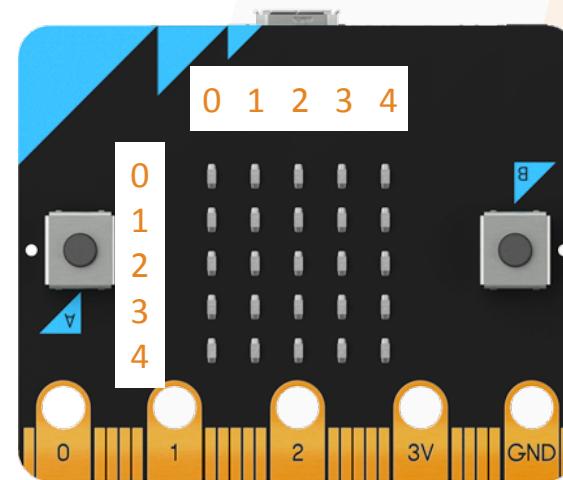
`variabele-naam = waarde`

# Tekeningen maken

```
X-coordinaat [0-4]          Helderheid [0-9]  
from microbit import *  
  
h = 9  
  
display.set_pixel(1,1,h)  
display.set_pixel(3,1,h)  
  
h = 3  
  
display.set_pixel(0,3,h)  
display.set_pixel(1,4,h)  
display.set_pixel(2,4,h)  
display.set_pixel(3,4,h)  
display.set_pixel(4,3,h)
```

Y-coordinaat [0-4]

Schakel de individuele LEDs van het scherm in/uit



# Variabelen verhogen

```
from microbit import *
```

```
x = 1
```

```
display.set_p
```

```
x = x + 1
```

Variabele x is vanaf nu gelijk aan:

$$x + 1 \leftrightarrow 1 + 1 \leftrightarrow 2$$

```
sleep(1000)
```

```
display.set_pixel(x,2,9)
```

```
x = x + 1
```

$$x + 1 \leftrightarrow 2 + 1 \leftrightarrow 3$$

```
sleep(1000)
```

```
display.set_pixel(x,2,9)
```

```
x = x + 1
```

$$x + 1 \leftrightarrow 3 + 1 \leftrightarrow 4$$

```
sleep(1000)
```

```
display.scroll("Go!")
```

Need to add gifs on every slide  
showing the output of every program...

# Herhaling

```
from microbit import *
```

```
x = 1
```

```
while (x <= 3):
```

Enkel als  $x \leq 3$ , dan...

```
    display.set_pixel(x,2,9)
```

```
    x = x + 1
```

```
    sleep(1000)
```

```
display.scroll("Go!")
```

```
from microbit import *  
  
x = 1  
  
display.set_pixel(x,2,9)  
  
x = x + 1  
  
sleep(1000)  
  
display.set_pixel(x,2,9)  
  
x = x + 1  
  
sleep(1000)
```

... moet je dit doen.

```
x = x + 1  
  
sleep(1000)  
  
display.scroll("Go!")
```

# Keuzes maken

```
from microbit import *
```

Als ik nu op de A-knop druk, dan...

```
if button_a.is_pressed():
```

... moet je dit doen ...

```
    display.show(Image.HAPPY)
```

```
else:
```

... en anders, ...

```
    display.show(Image.SAD)
```

... doe je dit.

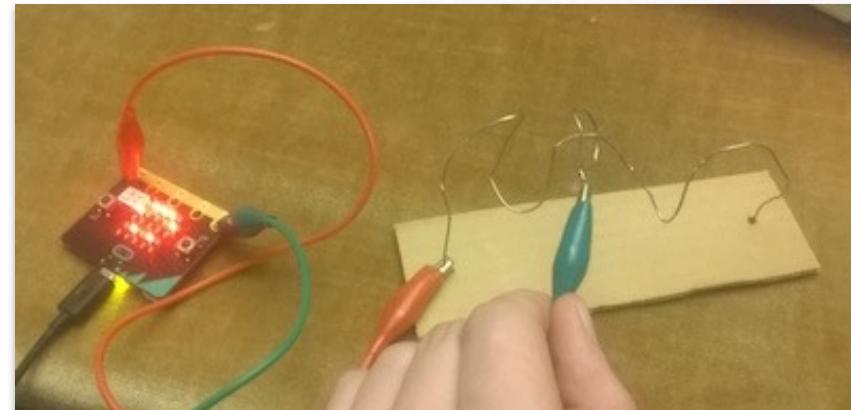
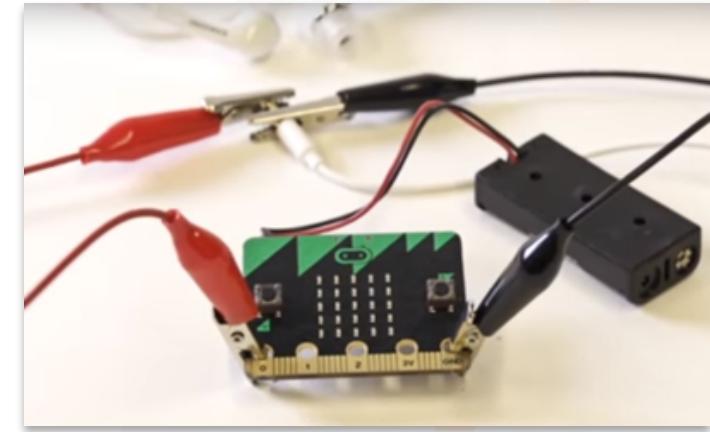
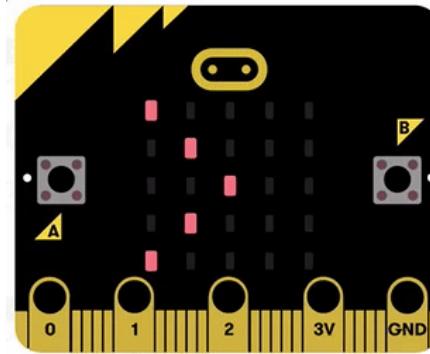
# Keuzes + herhaling

```
from microbit import *\n\nwhile True:\n    if button_a.is_pressed():\n        display.show(Image.HAPPY)\n    else:\n        display.show(Image.SAD)
```

while **True**: Altijd voldaan... → Deze herhaling stopt nooit!

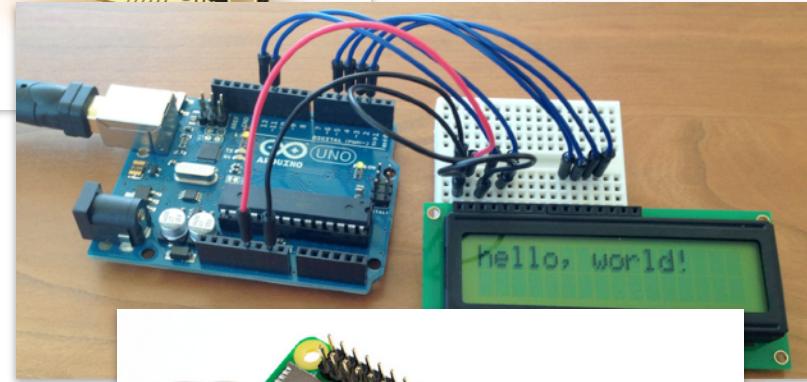
# Nu is het aan jullie :)

- Animaties maken
- Bibberspel
- Muziek maken
- Dobbelen
- Balansspel
- Kompas
- Geheugenspel
- Tetris

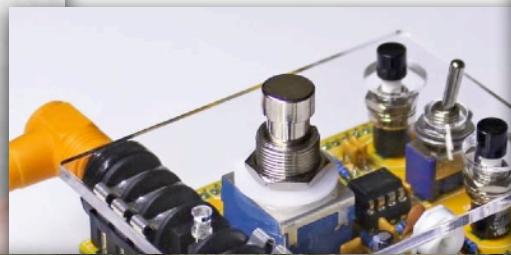
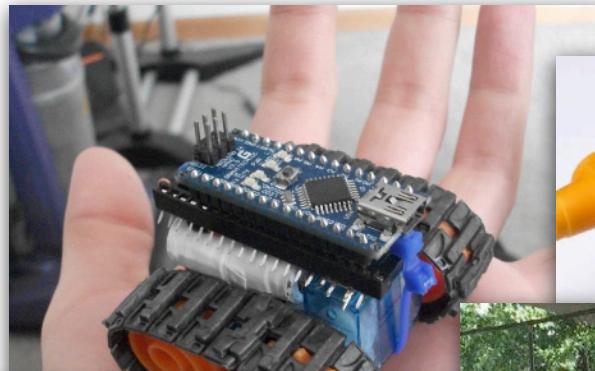


# De smaak te pakken?

- Micro:bit
- Arduino
- Raspberry Pi



# Om een idee te geven..



# Mocht je vragen hebben



@timmolderez

@mathsaey



Tim.Molderez@vub.be

Mathijs.Saey@vub.be