



#### Unit 6 : micro:bit met Python

#### Oefenblad 1 : Het Display

In deze les maak je je eerste Python-programma's voor het op verschillende manieren aansturen van het 'micro:bit'-display.

Deel 1: alien encounter

Deel 2: plaatjes weergeven

#### Doelen :

- Het display van de micro:bit gebruiken met de opdrachten `.show()`, `.scroll()` en `.show(image)`;
- De display snelheid instellen met `sleep(ms)`.

#### 1. Controleer voordat je begint het volgende:

- De TI-Nspire CX II heeft OS **5.3** of hoger.
- Je beheerst de beginselen van het programmeren in Python.
- De micro:bit is verbonden met de TI-Nspire CX II.
- Je hebt de instructies gevolgd voor het instellen van de micro:bit, zie:

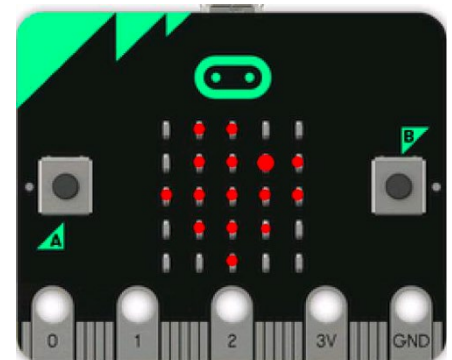
<https://education.ti.com/en/teachers/microbit>

*Deze procedure hoeft maar één keer gevolgd te worden, maar zorg dat je op de hoogte blijft van eventuele updates.*



#### 2. Als alles goed is gegaan ziet je micro:bit er zo uit, nadat je hem hebt aangesloten op je TI-Nspire:

Op het display van de micro:bit zie je het TI-logo: de staat Texas met een heldere stip op de plek van Dallas.



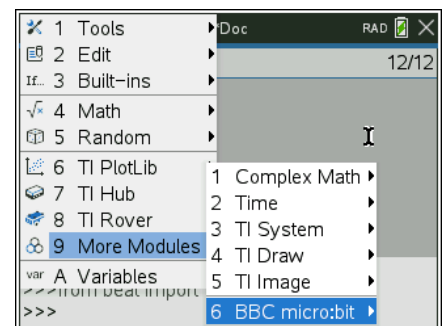
#### 3. Tot slot:

De micro:bit staat in de Python-bibliotheek en kun je als volgt vinden:

Druk in de Python-editor op **[menu] > More Modules**.

Je ziet dat de **'BBC Micro:bit'-module** hieraan is toegevoegd.

*Merk op dat in OS 5.3 en hoger Python-modules die in de **Pylib**-map zijn opgeslagen, in het menu worden toegevoegd aan de TI-modules.*





#### 4. Deel 1: Alien encounter

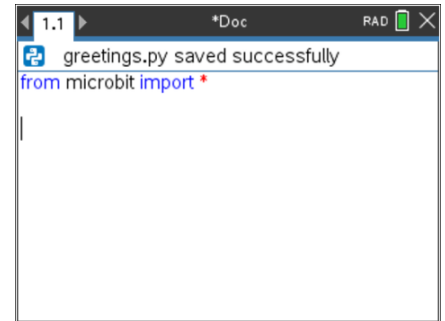
Zoals gebruikelijk bij een eerste programma begin je met een bericht op het 'micro:bit'-display.

Begin met een nieuw 'TI-Nspire'-document en kies voor **Add Python>New** om een nieuw programma te starten en noem het bijvoorbeeld: '**greetings**'.

Typ nu in de Python-editor **[menu]>More Modules>BBC Micro:bit** om de module te importeren:

```
from microbit import *
```

*Tip: Als de mededeling 'micro:bit not connected' verschijnt, koppel dan de micro:bit los en sluit hem opnieuw aan.*



```
1.1 | *Doc | RAD | [Battery Icon] | [Close Icon]  
greetings.py saved successfully  
from microbit import *
```

#### 5. Om een tekst op het display van de micro:bit te tonen, gebruik je de opdracht

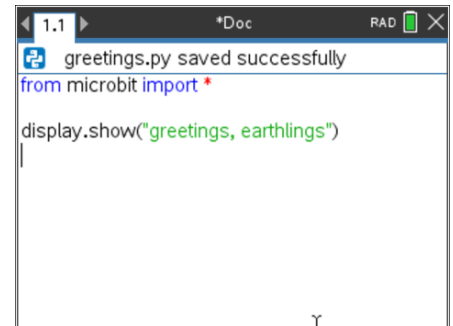
```
display.show(Image or text)
```

Je kunt deze opdracht vinden met:

**[menu]>More Modules>BBC Micro:bit>Display>Methods**

De opdracht wordt ingevoegd als **display.show(Value)** waarbij 'value' vervangen kan worden door bijvoorbeeld tekst. Typ hier tussen aanhalingstekens: "greetings, earthlings"

Als je dit programma uitvoert (bijvoorbeeld met **[ctrl] [R]**) zie je de letters één voor één verschijnen op het display.



```
1.1 | *Doc | RAD | [Battery Icon] | [Close Icon]  
greetings.py saved successfully  
from microbit import *  
display.show("greetings, earthlings")  
|
```



## In 10 minuten coderen - Python

MICRO:BIT EN TI-NSPIRE™ CX II

## UNIT 7: OEFENBLAD 1

### LEERLINGACTIVITEIT

6. Een betere manier om tekst weer te geven is:

**Display.scroll("greetings, earthlings")**

Ook deze opdracht kun je vinden bij:

**[menu]>More Modules>BBC Micro:bit>Display>Methods**

Voorzie de eerste opdracht van een commentaarsymbool om deze te negeren.

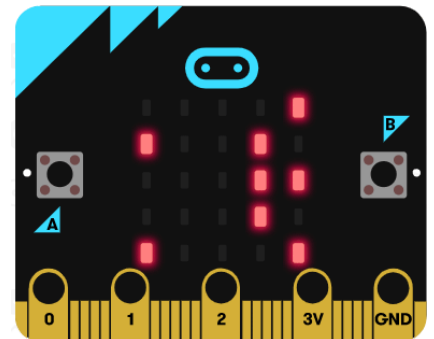
*Je kunt natuurlijk ook gewoon 'show' vervangen door 'scroll' met het toetsenbord.*

```
greetings.py saved successfully
from microbit import *
#display.show("greetings, earthlings")
display.scroll("greetings, earthlings")
```

7. De **.scroll()** -methode laat de tekst scrollen (bewegen) van rechts naar links, zodat het makkelijker wordt om deze te lezen. Je kunt de snelheid aanpassen met behulp van een extra parameter **delay**:

**Display.scroll("greetings, earthlings", delay = 200)**

Deze opdracht vertraagt het scrollen met 200 milliseconden (ofwel 0.2 seconden). Probeer ook andere waarden.



### 8. Deel 2: Hartslag

Open een nieuw document en kies **Add Python>New** om een nieuw programma te starten. Noem het bijvoorbeeld **'hartslag'**. Begin weer met het importeren van de 'micro:bit'-module:

**from microbit import \***

```
hartslag.py
from microbit import *
```



## In 10 minuten coderen - Python

### MICRO:BIT EN TI-NSPIRE™ CX II

## UNIT 7: OEFENBLAD 1

### LEERLINGACTIVITEIT

9. Om een hartsymbool op het 'micro:bit'-display weer te geven gebruik je de opdracht **display.show()**

Je kunt deze vinden bij:

**[menu]>More Modules>BBC Micro:bit>Display>Methods**

Tussen de haakjes vul je dan in **Image.HEART**.

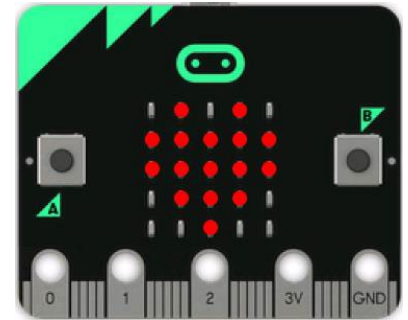
Je kunt dit intypen, maar je kunt het ook in het menu vinden:

**[menu]>More Modules>BBC Micro:bit>Display>Images>Set 1>HEART**

```
1.1 1.2 *Doc RAD 4/4
hartslag.py
from microbit import *
display.show(Image.HEART)
```

10. Voer het programma uit (**[ctrl] [R]**), dan zie je een hartvorm op het 'micro:bit'-display verschijnen.

Het hartsymbool blijft op het display staan, totdat er weer een andere opdracht gegeven wordt. Ook als het programma gestopt is.



11. Ga terug naar de Python-editor en voeg de opdracht toe om het kleine hartsymbool weer te geven:

**display.show(Image.HEART\_SMALL)**

Je kunt dit symbool vinden bij:

**menu]>More Modules>BBC Micro:bit>Display>Images>Set 1**

Maar je kunt ook de vorige regel kopiëren en plakken en dan **\_SMALL** toevoegen (let er dan wel op dat je hoofdletters gebruikt).

```
1.1 1.2 *Hartslag RAD 7/7
hartslag.py
from microbit import *
display.show(Image.HEART)
display.show(Image.HEART_SMALL)
```

12. Voer het programma uit. Dan zie je eerst het grote hart en dan direct daarna het kleine.

13. We maken nu een lus (loop) om het hart te laten “kloppen”.

Voeg vóór de twee opdrachten de volgende opdracht toe:

**while get\_key() != “esc”:**

deze kun je vinden met:

**[menu]>More Modules>BBC micro:bit>Commands**

Zorg ervoor dat de twee display-opdrachten beide inspringen (dit betekent dat beide opdrachten in een lus worden uitgevoerd).



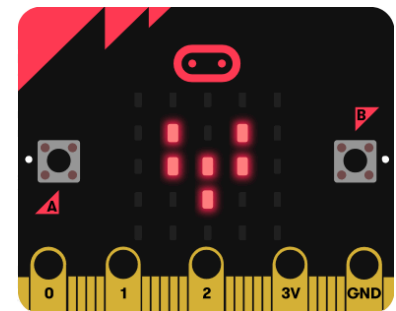
```
1.1 1.2 *Hartslag RAD 6/6
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    display.show(Image.HEART_SMALL)
```

14. Voer het programma weer uit, het hart klopt nu.

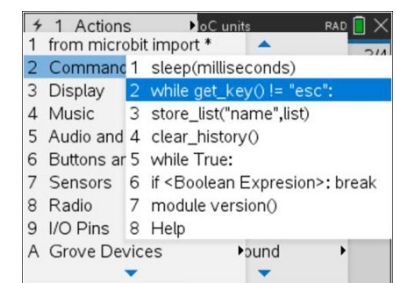
Je kunt het programma onderbreken met de escape-toets.

*Tip: Mocht het zo zijn dat je programma in een oneindige lus zit dan kun je met de **[home/on]** -toets het programma toch stoppen.*



15. In het ‘micro:bit’- **Commands** -menu kun je enkele nuttige Python-opdrachten vinden.

*Enkele van deze opdrachten staan ook in de menu’s van andere modules.*

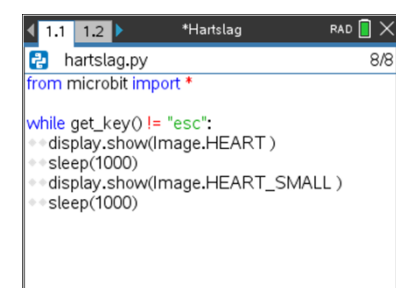


16. Om de snelheid van de hartslag te veranderen kun je twee keer de opdracht **sleep()** toevoegen. Achter elke display-opdracht één **sleep(1000)** betekent een wachttijd van 1000 milliseconden ofwel 1 seconde.

De opdracht kun je vinden in:

**[menu]>More Modules>BBC micro:bit>Commands**

*Tip: Let ook hier weer op het inspringen.*



```
1.1 1.2 *Hartslag RAD 8/8
hartslog.py
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    sleep(1000)
    display.show(Image.HEART_SMALL)
    sleep(1000)
```

17. **Extra:** Probeer ook eens verschillende gezichten uit.  
Gebruik dezelfde structuur als het hartslagprogramma maar gebruik nu verschillende gezichtsuitdrukkingen uit Set 1.

