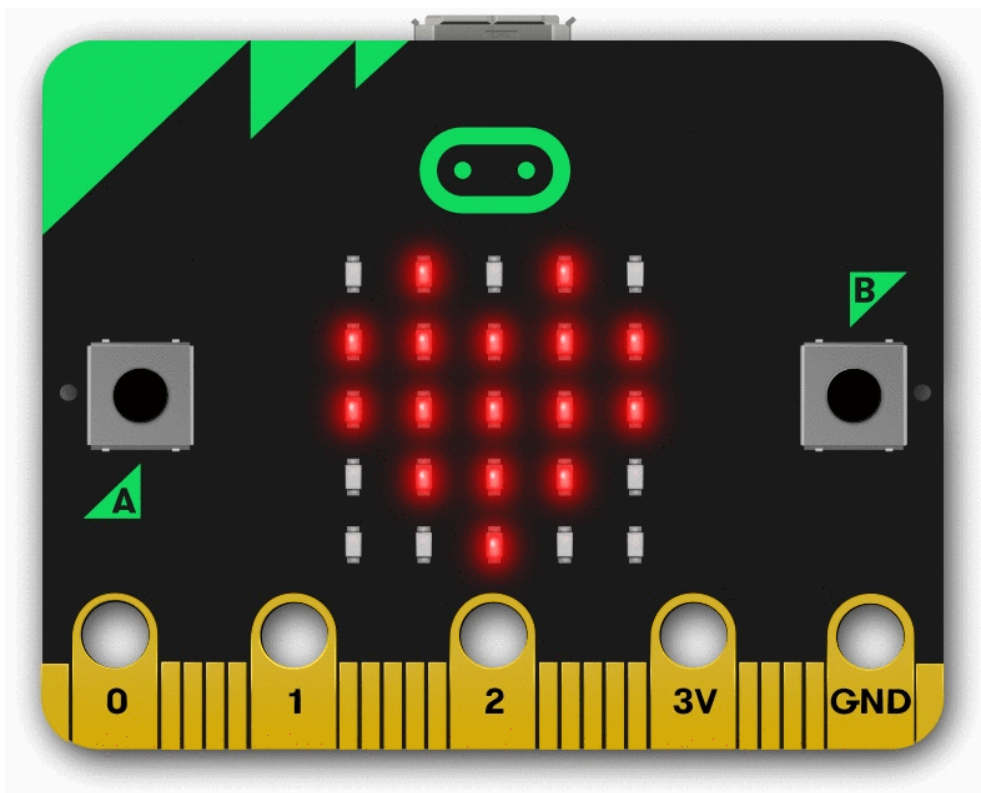


Greijdanus Zwolle

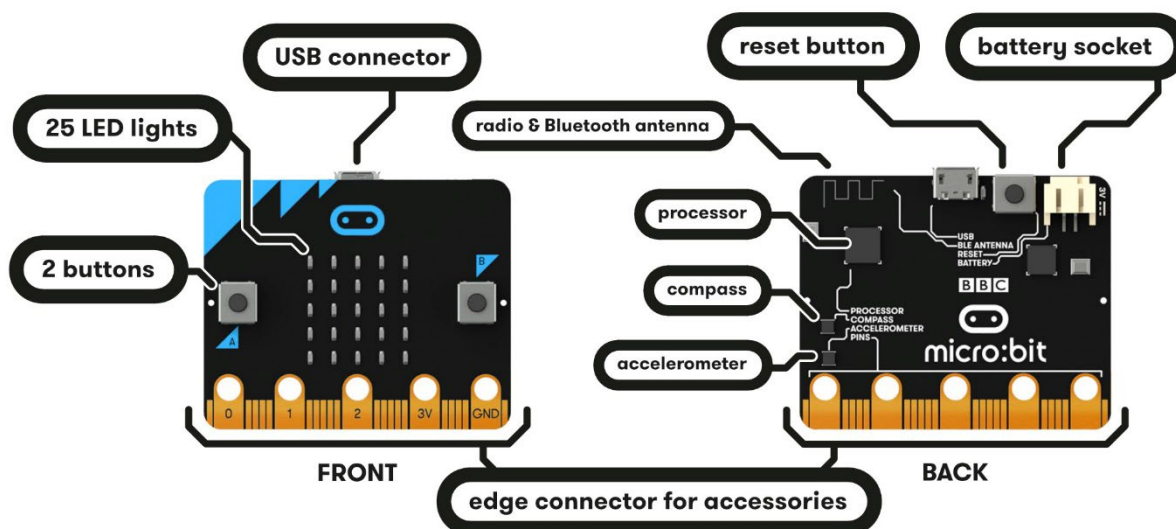
Lesbrieven micro:bit

Kennismaking met embedded systemen



H. Prins
31-10-2019

Lesbrief 1 Kennismaken met de Micro:bit



Doelen

- Ik weet wat een microcontroller is
- Ik kan eenvoudige code schrijven voor de Micro:bit
- Ik kan de code uitvoeren op een Micro:bit
- Ik kan een aansluitschema lezen en gebruiken met behulp van een breadboard

Introductie

In deze lesbrief maak je kennis met de Micro:bit. De Micro:bit is een voorbeeld van een embedded system die je zelf kan programmeren. Zie het als een eenvoudige mini-computer. Steeds meer apparaten worden tegenwoordig uitgerust met systemen die lijken op de Micro:bit. Denk maar aan alle slimme apparaten bij jou in huis. Dit is lang niet altijd zichtbaar omdat embedded systems onopvallend zijn weggewerkt. Een aantal andere eigenschappen van embedded systems zijn:

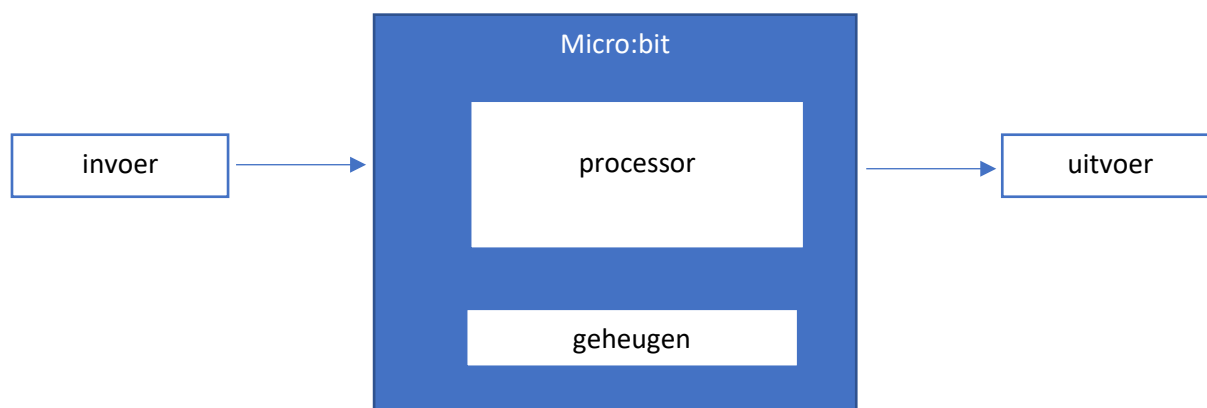
- dat ze energiezuinig moeten zijn,
- flexibel zijn zodat je het apparaat kan aanpassen aan de wensen van de gebruiker,
- en betrouwbaar zijn.

Opdrachten

1) Intro opdracht

Maak een lijst van slimme apparaten die allemaal op de markt zijn.

Hieronder zie je in een schema hoe de computer en dus ook de Micro:bit werkt. Er is invoer denk aan de twee knoppen (button) op de Micro:bit en er is uitvoer naar bijvoorbeeld de 25 leds op.



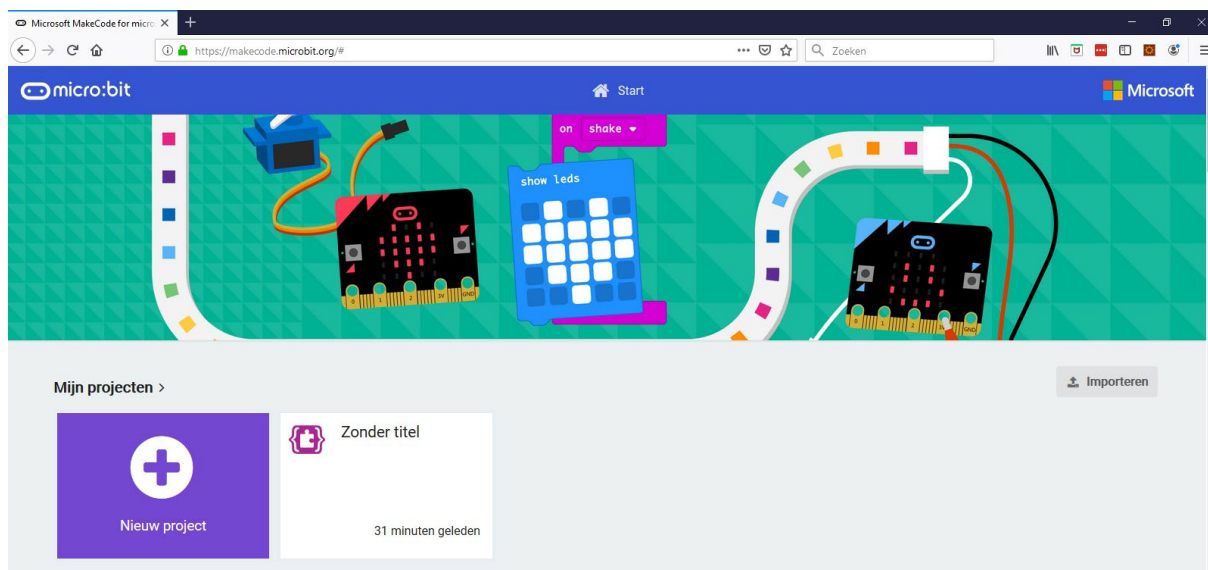
2) Intro opdracht

Maak een lijst van alle invoer en alle uitvoer mogelijkheden die standaard aanwezig zijn op de Micro:bit.

Invoer	Uitvoer

3) Opdracht kennismaken met MakeCode

Om de Micro:bit te kunnen programmeren heb je een programmeertaal nodig. In deze lesbrief maken we gebruik van een taal die lijkt op Scratch. Dit is een taal die met blokjes werkt die je in het scherm kan slepen. We maken gebruik van MakeCode (<https://makecode.microbit.org/>).



makecode.microbit.org startscherm

In deze eerste opdracht gaan we een startscherm programmeren. Daarvoor klik je op **nieuw project**.

TIP: let op de kleur van de blokken, daarmee kun je zien uit welke groep het blokje komt.

4) Opdracht led scherm bedienen

In de eerste opdracht maken we kennis met de uitvoer op het ledschermje. Maak onderstaande code na in MakeCode. Je kan direct rechts in het scherm zien wat er gebeurt. In de volgende opdracht gaan we de code ook op een echte Micro:bit uitvoeren.



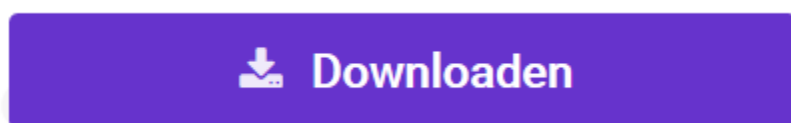
5) Opdracht exporteren naar Micro:bit

Als je onderstaand stappenplan volgt dan kan je de door jou gemaakte code uitvoeren op een echte Micro:bit.

Stap 1 Sluit de Micro:bit met de USB kabel aan op de computer.



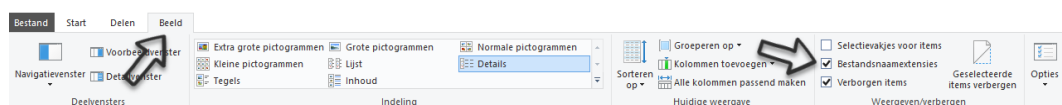
Stap 2 Klik op



Stap 3 Het bestand komt in de Download map van je computer. Het bestand heeft de extensie .hex

Naam	Gewijzigd op	Type	Grootte
▼ Vandaag (4)			
microbit-Zonder-titel.hex	31-10-2019 11:36	Microsoft MakeCode binary file	616 kB

LET OP: Je ziet extensies alleen als je dit ook hebt aangevinkt in het lint van de verkenner:



Stap 4 Verplaats het .hex bestand naar de MICROBIT schijf (deze verschijnt net als een USB stick.)



Alternatief

Je kan ook via de Store van Microsoft een app installeren van MakeCode. Deze app lijkt op de website van MakeCode. Het voordeel is dat deze app direct de Micro:bit herkent en je dus ook direct de code via de USB kan versturen.

Je kan de Micro:bit ook via een app op je telefoon of tablet bedienen.

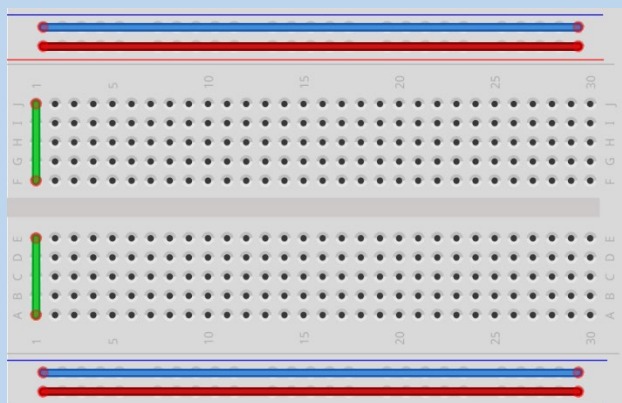
6) Opdracht led scherm

Verzin een leuk en creatief effect met de ledjes op de Micro:bit.

Wat is een breadboard?

Hieronder zie je een breadboard. Je kan het vergelijken met een printplaatje. Het is niet anders dan elektronische verbindingen. Aan de onder en bovenkant is de hele rij met elkaar verbonden (rode en blauwe lijn). In het middendeel zie je een aantal genummerde rijen en kolommen met een letter. De kolommen A t/m E zijn met elkaar verbonden en de kolommen F t/m J zijn met elkaar verbonden.

Het voordeel van een Breadboard is dat je allerlei elektronische verbindingen makkelijk kan testen. Je kan zonder problemen knoppen, ledjes en weerstandjes in de breadboard prikken. Met behulp van jumperkabels kan je verschillende rijen onderling verbinden of aansluiten op een Micro:bit (als deze tenminste is voorzien van een speciaal extension board).



Breadboard

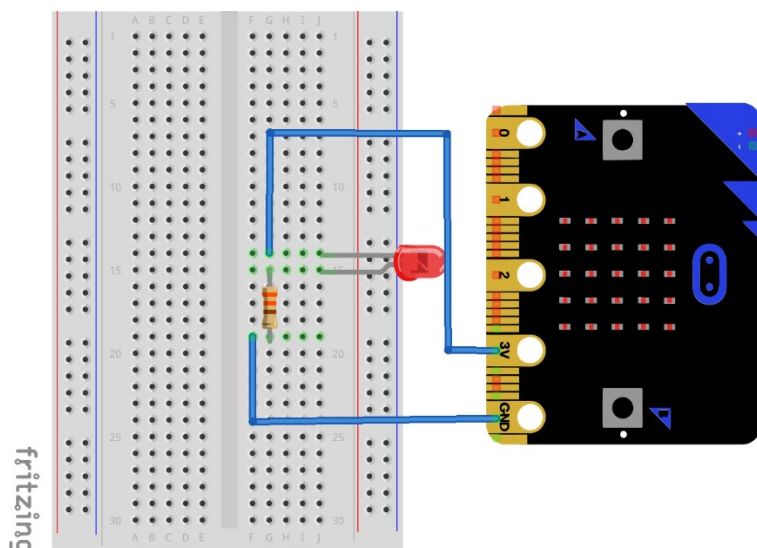


Printplaat

7) Led aansluiten

Je kan ook van alles aansluiten op een Micro:bit. Dit zijn bijvoorbeeld sensoren (invoer) zoals brand detectie, afstandmeters, temperatuurmeter enz.. Maar het kunnen ook actuatoren (uitvoer) zijn zoals ledjes, buzzers en motoren. In deze opdracht ga je een ledje aansluiten.

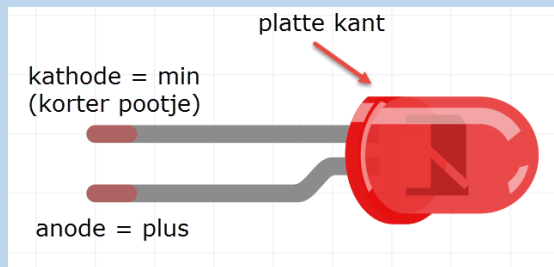
De gebruikte weerstand is **330 Ohm**. Doe je dit verkeerd dan kan je de led als de Micro:bit om zeep helpen. Als je de Micro:bit op de computer aansluit gaat de led als het goed is branden. (brandt de led niet, sluit de led dan andersom aan (zie kader over de led)).



Wat is een led?

Een eigenschap van een LED is dat de elektriciteit maar op één manier door de led kan stromen. Dus van plus naar min. Het maakt dus uit hoe je de led aansluit op de Micro:bit. Hieronder staat een plaatje waarop je kan zien wat het plus en min pootje is.

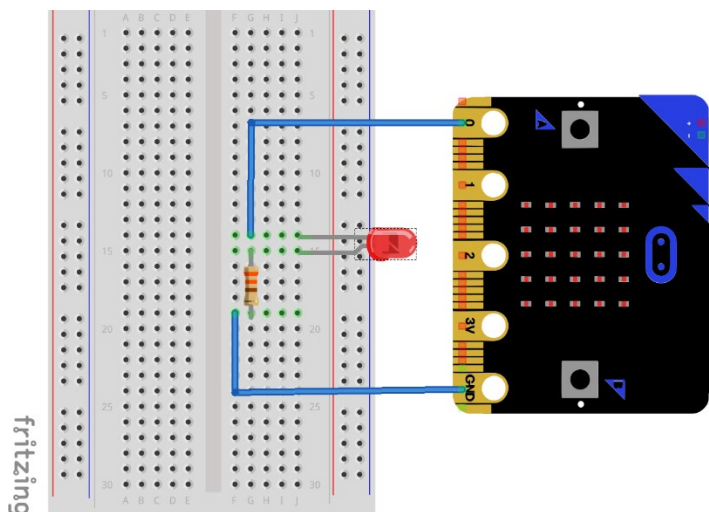
Veel lampsoorten branden op basis van de hoeveelheid spanning die ze krijgen (volt). Bij een led is dat niet zo, deze brandt op basis van de hoeveelheid stroom (ampère).



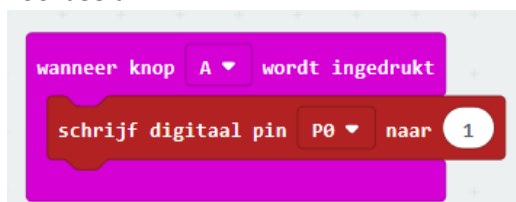
8) Led besturen met Micro:bit

In de vorige opdracht hebben we de led aangesloten op de 3V aansluiting van de Micro:bit. Op deze aansluiting geeft de Micro:bit altijd ongeveer 3,3 volt. De GND staat voor Ground, dit moet je zien als de plaats waar je min draad aansluit om de cirkel rond te krijgen.

Om de led aan en uit te kunnen zetten moet je de led aansluiten op een andere pin. In dit geval pin 0. Deze geeft standaard geen spanning.



a) Programmeer knop A zodat deze zorgt dat op pin 0 spanning komt. Hieronder staat het voorbeeld.



b) Programmeer knop B zodat de led ook weer kan worden uitgezet.

Achtergrond: waarom een weerstand van 330 Ohm?

Dit is een stukje Natuurkunde voor de liefhebbers.

- Een led brandt als deze een bepaalde hoeveelheid **stroom** (I) krijgt, bij een standaard led is dat ongeveer 20mA (0,02 ampère)
- De **spanning** (U) die de Micro:bit levert is 3,3 volt
- Er is 1 rode led, een rode led gebruikt ongeveer 1,6 volt

- Weerstand (R) bereken je door

$$R = (U_{bron} - U_{led}) / I$$

$$R = (3,3 - 1,6) / 0,02 = 85 \text{ Ohm}$$

Je neemt dan een weerstand van bijvoorbeeld 100 Ohm (hoger is veiliger). In de tekening hierboven gebruiken ze 330 Ohm omdat een led ook prima brandt met minder ampère. Als je naar voorbeeldschakelingen zoekt zie je deze verschillen regelmatig. Het is de moeite waard om in deze opdracht ook de 100 Ohm weerstand te proberen.



Lesbrief 2 Leren programmeren

Doelen

- Ik ken de drie basisstructuren van een programmeertaal
- Ik weet wat variabelen zijn en kan ze gebruiken
- Ik kan een herhaling programmeren
- Ik kan een keuze programmeren
- Ik kan een event programmeren

Introductie

Als je gaat programmeren heb je in basisstructuren te maken. Hieronder staan ze uitgelegd in deze lesbrief gaan we daar mee aan de slag.

<p>Opeenvolging</p>	<p>Dit is het eenvoudigste en hebben we tot nu toe vooral gedaan. We lopen van boven naar beneden de code door.</p>  <p>In bovenstaand voorbeeld zie je daarvan een uitwerking. Als de Micro:bit wordt aangezet dan loopt de Micro:bit bovenstaande vier blokjes door en stopt dan.</p>
<p>Herhaling</p>	<p>Je kan ook code een aantal keren herhalen. Dat is bijvoorbeeld nuttig als je de tafel van 2 wilt uitreken:</p> $1 \times 2 = 2$ $2 \times 2 = 4$ <p>Enz enz, daar zit een bepaalde herhaling in. In onderstaand voorbeeld is ook een herhaling. Het groene blok wordt 5 (!) keer uitgevoerd, van 0 t/m 4. Die waarde wordt in een variabele index gezet.</p>  <p>De uitvoer is: hartje, 0, hartje, 1, hartje, 2, hartje, 3, hartje, 4</p> <p>Er zit in deze code nog een herhaling. Het blok 'de hele tijd' dat er omheen zit is een 'eeuwige' herhaling.</p>

<p>Keuze</p>	<p>Als je gaat programmeren wil je gebruik maken van keuzes. Daarmee breiden we bovenstaand voorbeeld uit. Als index 2 is dan wil ik niet een hartje laten zien maar uitroeptekens. Dat doe ik als volgt:</p>  <p>Daarmee wordt de uitvoer van de code: hartje, 0, hartje, 1, uitroeptekens, 2, hartje, 3, hartje, 4</p>
--------------	--

Opdrachten

9) Introductieopdracht

Maak het voorbeeld uit de introductie na. Zorg dat je deze werkend uitvoert op een Micro:bit.

Variabelen

In het voorbeeld kwam je de variabele **index** al tegen. Maar wat is een variabele eigenlijk precies. Je kunt het vergelijken met een verhuisdoos. In de ene doos zitten boeken en in de andere zitten kleren en dan zijn er nog dozen waarin je persoonlijke spulletjes zitten. Elke doos heeft een andere inhoud. Een variabele is zo'n doos. Hoe die doos precies heet maakt niet uit. Dus dat mag *index* zijn, maar ook *getal* of gewoon een letter *x* of *y*. Het beste is dat de variabele vertelt wat erin zit. Net zoals je op een verhuisdoos ook zet wat er in de doos zit. Je mag daarbij een spaties of andere bijzondere tekens gebruiken.

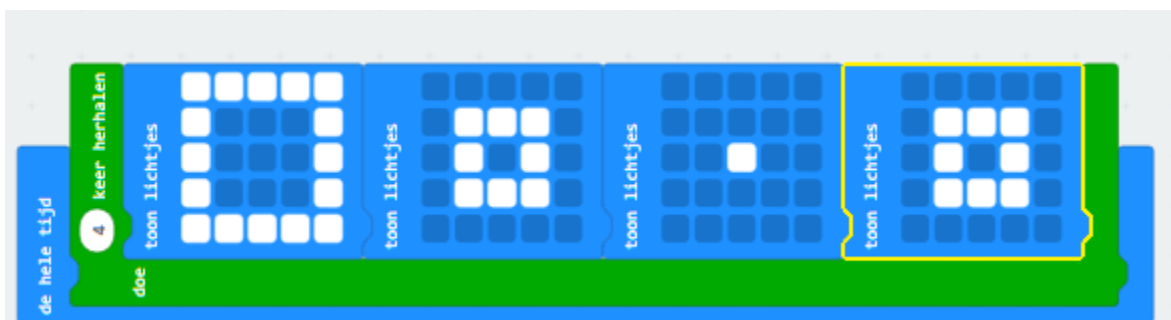


Tijdens het uitvoeren van het programma kan de inhoud van de variabele veranderen. Voor programmeurs is het heel handig dat ze gegevens tijdelijk kunnen bewaren in een variabele en later weer kunnen gebruiken of veranderen. Neem bijvoorbeeld de totaalscore in een game. Deze wil je het hele spel bewaren en als je weer iets goed hebt gedaan wil je deze verhogen.

Ook de waarden die worden doorgegeven door verschillende sensoren op de Micro:bit kan je zien als variabelen. Kijk maar de lichtsensor in opdracht 13.

10) Opdracht herhalen

In deze opdracht gaan we een lus maken die een mooi effect geeft op het led schermpje van de Micro:bit. Maak onderstaande code na.



Let op: deze afbeelding is in verband met de ruimte gedraaid

11) Opdracht herhalen verbeterd

Hieronder zie je nog een keer de herhaling, maar nu wordt gebruik gemaakt van een variabele **index** de variabele begint bij 0 en telt bij elke lus er eentje bij. Daarmee wordt door het paarse blokje elke keer een ledlampje op de Micro:bit aangezet. Dus eerst lampje 0,0 (= linksboven), daarna 1,0 tot en met 4,0. De ledjes die aanstaan blijven ook aan. Probeer dit voorbeeld maar eens uit.

	0	1	2	3	4
0	0,0	1,0	2,0	3,0	4,0
1	1,0	1,1	2,1	3,1	4,1
2	2,0	1,2	2,2	3,2	4,2
3	3,0	1,3	2,3	3,3	4,3
4	4,0	1,4	2,4	3,4	4,4



x,y van led scherm

- Probeer bovenstaand voorbeeld uit.
- Verander nu het voorbeeld zo dat de middelste kolom de ledjes één voor één aangaan en blijven.
- Verander het nu dat de ledjes diagonaal aangaan van linksboven tot rechtsonder.
- Kan dit ook andersom, dus van rechtsonder naar linksboven? Verander je code.

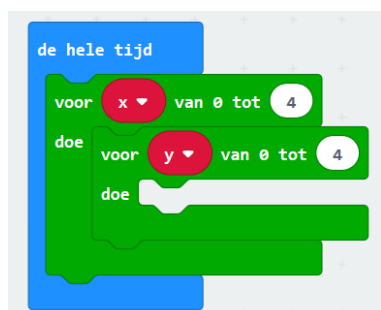
Hint:

- Tot nu toe bleven alle ledjes aan, kan je er ook voor zorgen dat er telkens maar één ledje brandt? Verander de code.

Hint:

12) Opdracht nesten

Je kan ook twee herhalingen combineren. Dit wordt ook wel 'nesten' genoemd. Zie het voorbeeld hieronder. Daarvoor maak je zelf twee nieuwe variabelen 'x' en 'y'. Let er goed op dat je niet dezelfde variabelenaam gebruikt. Dat gaat namelijk niet werken.



Maak de code waarin je alle ledjes van de Micro:bit één voor één laat branden.

13) Opdracht als licht dan ... anders donker ...

In deze opdracht gaan we bezig met een keuze code. Maar eerst gaan we ontdekken hoe de lichtsensoren op de Micro:bit werkt.

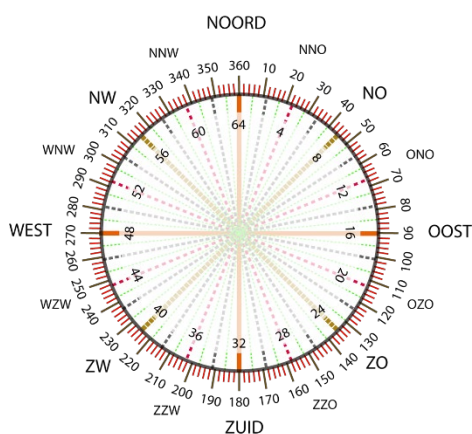
- a) Maak onderstaande code na en kijk welke waarden de lichtsensoren geeft. Maak het lichter en donkerder bij de Micro:bit.



- b) Maak nu een code die zorgt dat alle ledjes gaan branden als het schemerig wordt. Anders moeten de ledjes niet branden.

14) Opdracht kompas

In de volgende opdracht gaan we met een andere sensor aan het werk. We maken weer een programma met een keuzeblok. laat de Micro:bit een pijl zien als je deze richting het noorden houdt. Anders krijg je een verdrietig gezichtje. We houden een marge van 10 graden naar beide kanten.



15) Opdracht events

In de vorige lesbrief hebben we al kennis gemaakt met de knoppen op de Micro:bit. Een knop zorgt voor een zogenaamd event. Door het handelen van jou als gebruiker breek je in op de standaard loop van het programma. Hieronder zie je hoe je met een event twee nummers op het scherm kan zetten.



- a) Maak bovenstaande code na en voer deze uit.
- b) Je wilt nu met een variabele **score** tellen hoeveel doelpunten er zijn behaald. Programmeer knop A zo dat deze telkens de score met één verhoogd als er op de knop wordt gedrukt. De score moet daarna zichtbaar worden op het scherm.
- c) Je wilt ook de mogelijkheid om een punt weer af te keuren. Zorg dat je met knop B de mogelijkheid hebt om de score weer met één te verlagen. Zorg er wel voor dat de score niet onder de 0 kan komen.

Lesbrief 3 Leren programmeren

Doelen

- Ik kan een algoritme bedenken en omzetten in code(blokken).
- Ik kan nog analoge en digitale sensoren aansluiten en uitlezen.

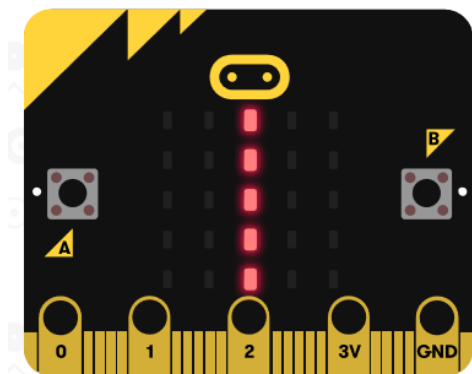
Introductie

Als je alle basisstructuren kent van programmeren kan je beginnen aan het schrijven van algoritmes. Als je een algoritme zelf kan maken dan bedenk je dus zelf de oplossing voor een programmeerprobleem. Je zult merken dat het zelf verzinnen van goede oplossingen nog best wel lastig is.

Opdrachten

16) Verplaats lichtbalk door kantelen Micro:bit

Als je de Micro:bit aanzet zie je het volgende:

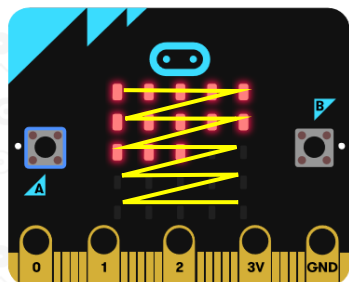


Door het kantelen van de Micro:bit verplaats je de lichtbalk naar links of naar rechts. Hieronder staan een aantal tips, maar deze mag je ook negeren.

<p>Tip</p>	<p>Maak een functie lichtjes</p>  <p>De pos is een variabele die je kan toevoegen als je de functie maakt (type number).</p>
<p>Tip</p>	<p>Dit zijn je events en opstart code:</p> 

17) Vul het scherm één voor één

Als je de Micro:bit aanzet is het scherm leeg. Je gaat nu alle ledjes één voor één aanzetten met de A knop. Zie gele lijn op onderstaande plaatje.



<p>Tip</p>	<p>Stel de variabelen X en Y in en zet ze standaard op 0.</p>
------------	---

18) Vervolg: opnieuw beginnen

Ga verder met de code uit de vorige opdracht. Zorg nu dat als alle ledjes aanstaan de code weer opnieuw begint. Dus alle lampjes weer uit en dan linksboven weer één voor één de ledjes aan.

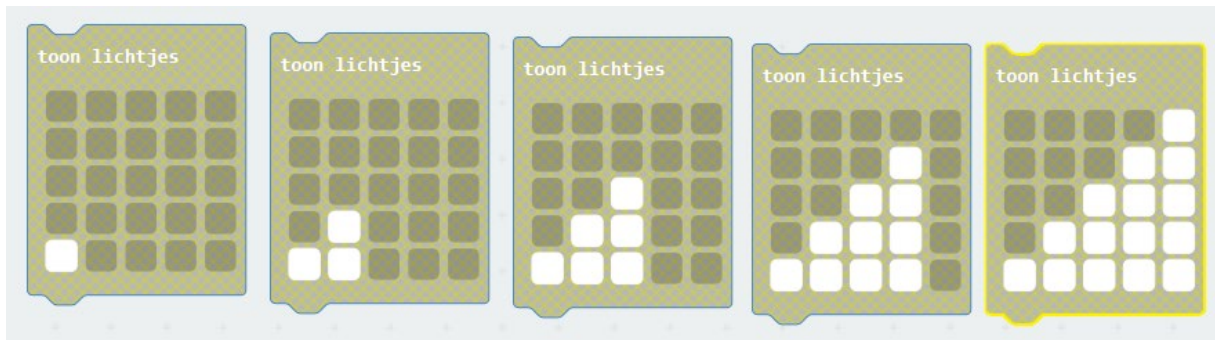
- a) Voer de opdracht uit zoals beschreven.
- b) Zorg dat je ook halverwege met de B knop kan resetten.

19) Uitlezen analoge pin

Maak onderstaande schakeling na. Gebruikt zijn een Potentiometer en een weerstand van 10kΩ.

20) Analooq naar digitale output

Werk de vorige opdracht nu zo uit dat het led scherm laat zien hoe hoog de pin waarde is. Zoals in het voorbeeld hieronder is aangegeven.



Tip	
-----	--

Lesbrief 4 Eindopdracht

Doelen

- Ik kan een complex algoritme omzetten in een werkend spel.
- Ik kan op basis van aanwijzingen (ontwerp) een werkend spel bouwen.

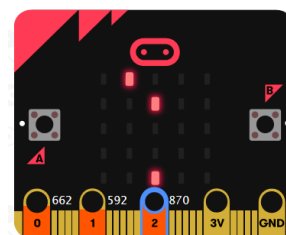
Introductie

In deze lesbrief combineren we de vaardigheden uit de vorige lesbrieven. In deze lesbrief laat je zien dat je met de opgedane kennis een werkend spel kan maken. Het is helemaal mooi als het je ook lukt om zelf een spel te verzinnen.

Opdrachten

21) Eenvoudig spel

Maak een spel waarbij je onderste rij ledjes het kanon zijn. Dit Kanon kan alleen recht naar boven schieten. De bovenste rij ledjes zijn vliegtuigen. Om de seconde verschijnt er een vliegtuig op een willekeurig ledje in de bovenste rij. Door op een knop te klikken kan je met het kanon het vliegtuigje uit de lucht schieten. Met twee andere knoppen kan je het kanon naar de juiste plaats verplaatsen.

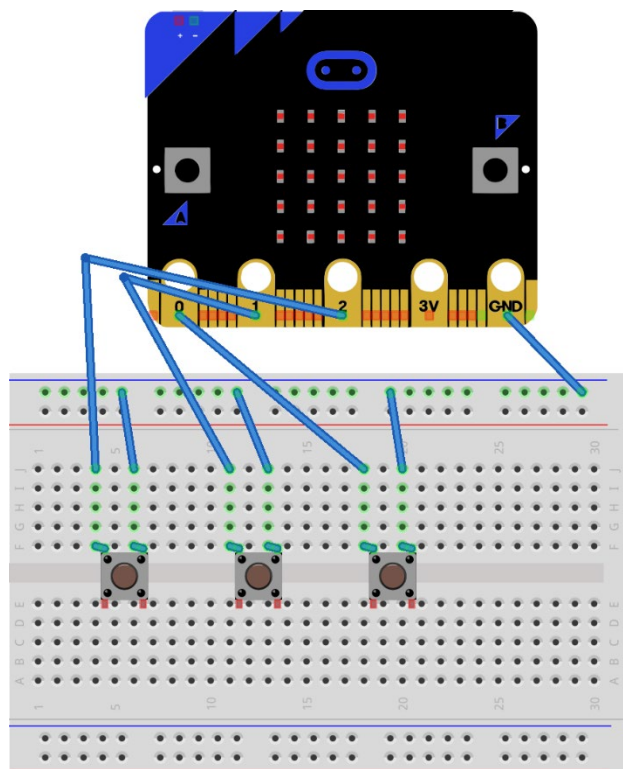


Als je schiet dan zie je de ledjes er boven één voor één oplichten totdat deze het bovenste ledje raakt en uit zet.

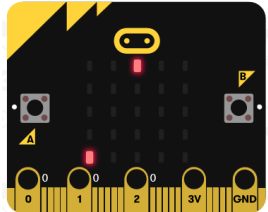
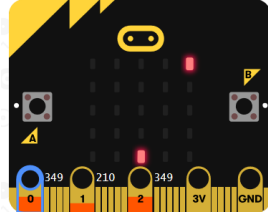
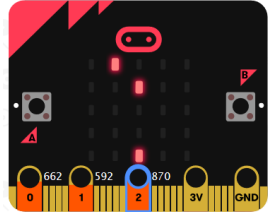
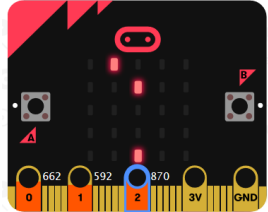
Hieronder zie je de bijbehorende schakeling.

Extra aanwijzingen

- Zodra je op knop A drukt zie je de huidige score.
- Als je op knop B drukt zie je ook de score en wordt deze weer op 0 gezet.



fritzing

Startscherm	Kanon verplaatst	Kanon schiet	Raak?
			
Bovenste ledje verspringt willekeurig elke seconde (de vlieg)	Door te klikken op een knop verbonden met pin 0 en 1 verplaats je de vlieg	Door te klikken op knop verbonden met pin 2 schiet je omhoog. Je ziet de kogel telkens een rij hoger gaan.	De score gaat met één omhoog.

Beoordeling

Omschrijving	Punten	Jouw score
Ik kan met de buttons een led van links naar rechts bewegen (onderste rij).	1	
Ik zorg ervoor dat er altijd een led brandt op de onderste rij.	1	
Ik kan een willekeurige led om de seconde laten oplichten in de bovenste rij.	1	
Ik kan een de led van de bovenste rij 'schieten'.	1	
Het schieten kan ik zichtbaar maken met de ledjes in rij 2,3 en 4.	1	
Als ik raak schiet krijg ik een scorepunt.	1	
Ik kan de huidige score opvragen met knop A.	1	
Ik kan het spel resetten met knop B.	1	
De code is netjes uitgewerkt.	1	
Je kan je code goed uitleggen.	1	
Cijfer		

22) Verzin zelf een spel

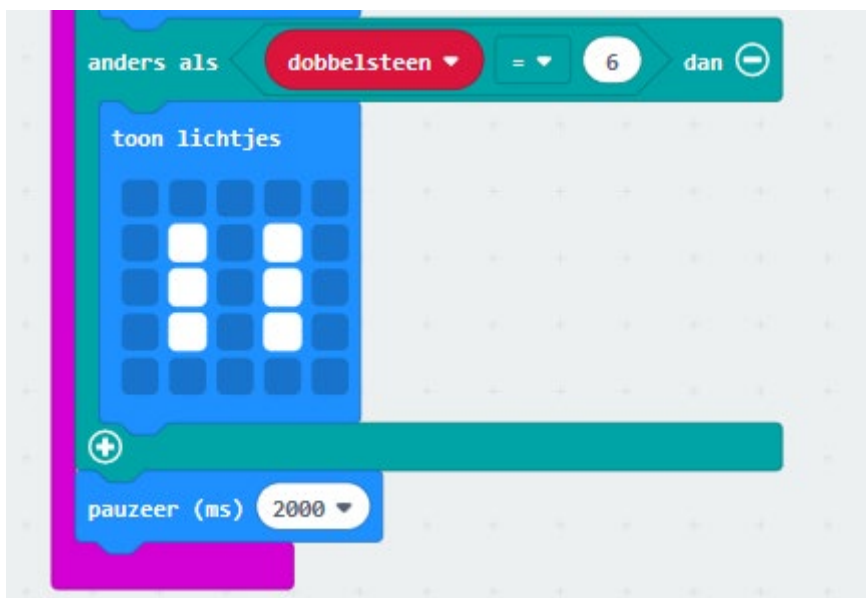
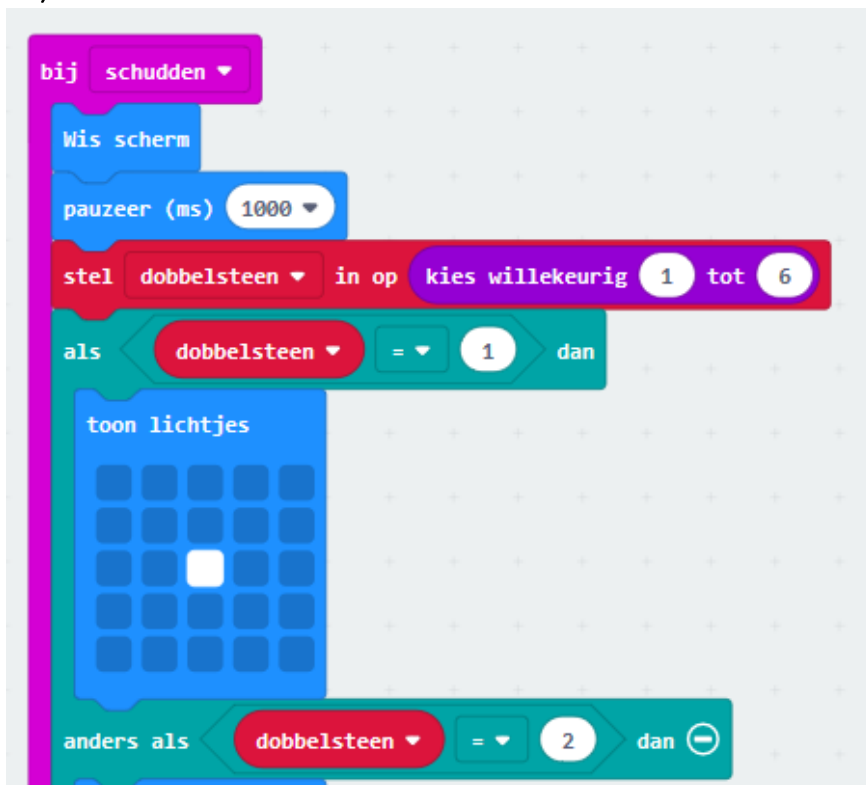
Dit is een vrije slotopdracht waarin je zelf een spel verzint voor op de Micro:bit. Kijk daarbij ook rustig naar andere mogelijkheden van sensoren en actuatoren. In overleg met de docent is er veel mogelijk. Eventueel kan je gebruik maken van de radio functie en dus met een ander groepje/persoon de slotopdracht samen maken.

Beantwoord voordat je begint met ontwikkelen de volgende vragenlijst:

Een korte omschrijving van het spel	
Maak een aantal tekening waarin je het spel uitlegt	
Welke sensoren gebruik je?	
Welke actuatoren gebruik je?	
Welke variabelen heb je nodig?	
Beschrijf de code.	
Teken de nodige schakelingen	

Extra opdracht ideeën

23) Dobbelsteen



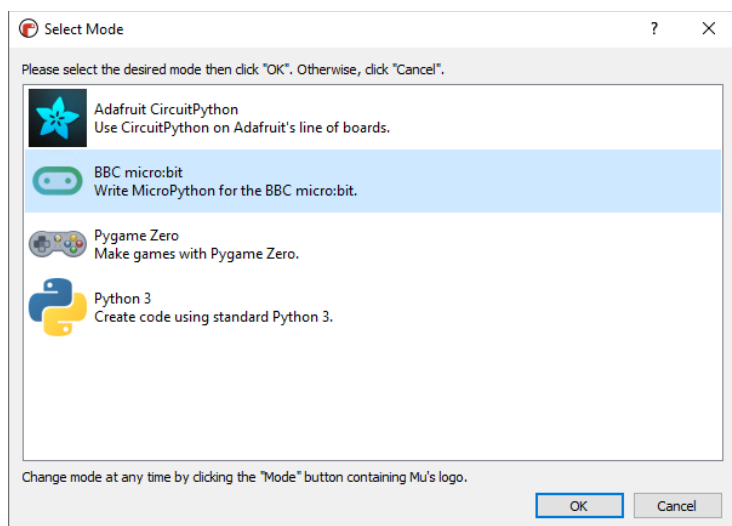
24) Escaperoom

Laat elk groepje een activiteit voor een Escaperoom uitwerken. Zo kan je met de hele klas een Escaperoom maken. Drie ideeën om alvast een begin te hebben (vergeet niet het spannende verhaal dat erbij hoort):

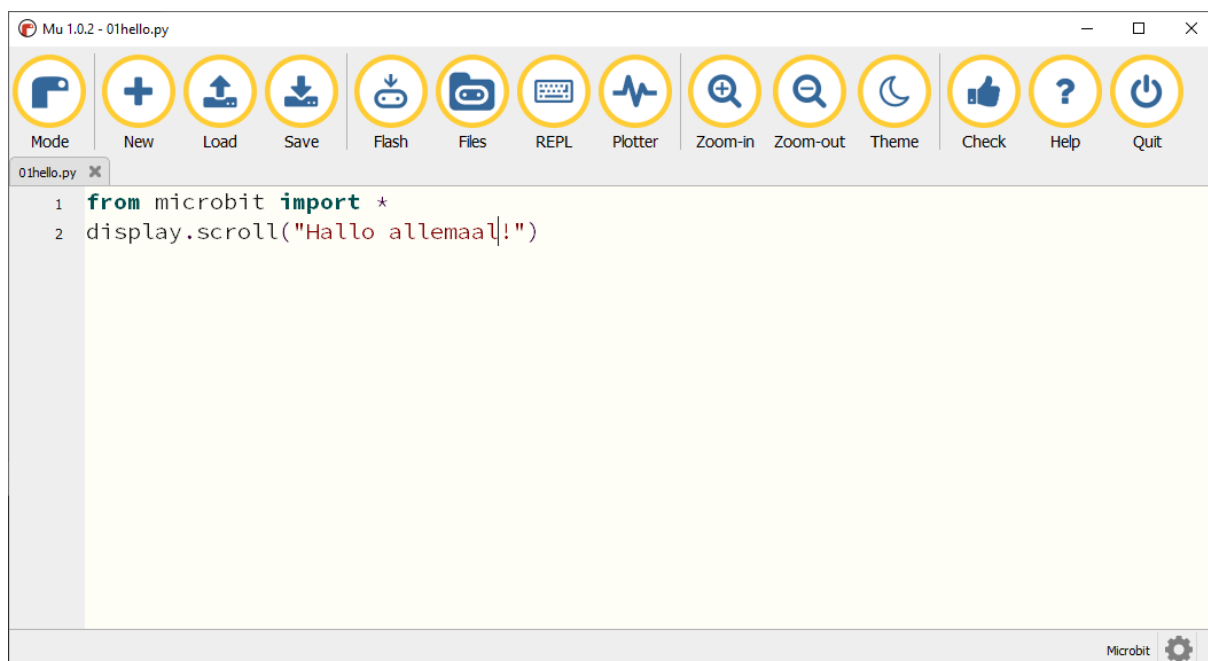
- Potentiometers raadsel
- Kistje met slotje (servo motor) en geluid of licht invoer om slotje te openen
- Knoppen raadsel of raadsel via display

Verdieping Mirco:bit met micropython

Je kan de Micro:bit ook met een echte programmeertaal programmeren. De Micro:bit ondersteund daarvoor Python. Je kan dit net als met de Scratch taal online doen (<https://python.microbit.org/v/2.0>), maar je kan ook een editor installeren op je computer. De editor heet Mu en daarmee kan je Python code programmeren (<https://codewith.mu/en/download>). Dit programma kan je ook op een usbstick zetten met PortaMu.



Kies hier voor BBC micro:bit



Hierboven zie je een programma die Hallo allemaal! op het ledschermje laat zien. Deze code is vergelijkbaar met onderstaande code in blokjes:



De onderstaande regel moet er altijd zijn:

```
1 from microbit import *
```

Deze regel zorgt ervoor dat de Micro:bit jou kan begrijpen. Je haalt daarmee code in jouw applicatie die die al door de programmeurs van de Micro:bit geschreven zijn zodat de code weet dat bij `display.scroll(" ")` het ledscherm wordt bedoeld.

Let op: elke typefout maakt dat de code NIET werkt. Dus ook het gebruik van kleine en hoofdletters moet je zeer precies doen. In Python is het ook heel belangrijk dat je alle code netjes in de linkerkantlijn onder elkaar staat. Dus geen extra spaties!

Uitleg van `display.scroll("Hallo allemaal!")`

De code `display` is een zogenaamd object. Dit object bedient het led scherm op de Micro:bit. Je kan via allerlei methodes dit object bewerken. In dit geval met de methode `scroll()`. Let op de open en sluit haak achter de methode. Dit geeft aan dat het een methode is. Het is een soort functie die aan de hand van attributen (die tussen de haakjes staat) een eigen stukje code weer uitvoert. In dit geval geef je een tekst mee. Alles wat een waarde is (zoals tekst) en geen Python commando moet tussen de hoge komma's staan. In dit geval geef je de tekst "Hallo allemaal!" mee. De functie `scroll` zorgt ervoor dat deze tekst vervolgens over het scherm wordt gerold.

Codevoorbeelden op een rij

<pre>boat = Image("05050:" "05050:" "05050:" "99999:" "09990") display.show(boat)</pre>	<p>Bedienen ledscherm Je kan elk ledje een helderde kleur geven. 0 is uit en 9 is het felste licht.</p> <p>Met het voorbeeld hiernaast kan je dit per led instellen.</p> <p><code>boat</code> is een variabele in Python.</p>
<pre>display.show(Image.HAPPY)</pre>	<p>Deze code laat een ingebouwde afbeelding zien.</p> <p>Er is een hele lijst met alle mogelijke afbeeldingen.</p>
<pre>sleep(10000) display.scroll(str(button_a.get_presses()))</pre>	<p>Dit script wacht 10000 miliseconden (= 10 seconden) en toont dan hoe vaak er is gedrukt op knop a.</p>
<pre>while running_time() < 10000: display.show(Image.ASLEEP) display.show(Image.SURPRISED)</pre>	<p>Dit script toont 10 seconden de ASLEEP afbeelding en daarna de SURPRISED afbeelding.</p>
<pre>while True: # Do stuff</pre>	<p>Deze code zorgt voor een continu loop (de hele tijd zoals we die kennen van het blokken programmeren).</p>

<pre>while True: if button_a.is_pressed(): display.show(Image.HAPPY) elif button_b.is_pressed(): break else: display.show(Image.SAD) display.clear()</pre>	<p>Hier zie je het afvangen van het drukken op knop a (HAPPY afbeelding) of knop b (stoppen van de continu loop). Anders wordt SAD afbeelding getoond.</p>
<pre>if something is True: # do one thing elif some other thing is True: # do another thing else: # do yet another thing.</pre>	<p>Dit is de als dan, anders dan uit de blokjes code. Let goed op het inspringen (een tab), anders werkt de code niet goed.</p>
<pre>while True: if pin0.is_touched(): display.show(Image.HAPPY) else: display.show(Image.SAD)</pre>	<p>Deze code staat ook een een continu loop (de hele tijd) en controleert of er op pin0 een knop (of iets anders) wordt ingedrukt.</p>
<pre>pin0.write_digital(1) sleep(20) pin0.write_digital(0) sleep(480)</pre>	<p>Dit is digitaal schrijven (zie opdracht 8 met het led lampje aan en uit).</p>
<pre>import random display.show(str(random.randint(1, 6)))</pre>	<p>Dit is de dobbelsteen, let op dat als je het object <code>random</code> gebruikt je deze wel moet importeren.</p>
<pre>while True: if button_a.was_pressed(): display.show(str(random.randint(1, 6)))</pre>	<p>De dobbelsteen actief onder knop a (gooien).</p>
<pre>while True: reading = accelerometer.get_x() if reading > 20: display.show("R") elif reading < -20: display.show("L") else: display.show("-")</pre>	<p>Hier zie je de accelerometer actief.</p>
<pre>while True: gesture = accelerometer.current_gesture() if gesture == "face up": display.show(Image.HAPPY) else: display.show(Image.ANGRY)</pre>	<p>Hier nog een voorbeeld met bewegingsdetectie.</p>
<pre>compass.calibrate() while True: needle = ((15 - compass.heading()) // 30) % 12 display.show(Image.ALL_CLOCKS[needle])</pre>	<p>Het kompas in werking.</p>



Meer informatie en online cursus?

Als je goed wordt in Python kan je met de Micro:bit meer dan je met de MakeCode kan programmeren. Maar dit is wel voor doorzetters. Het is ook goed voor je kennis van het Engels. De meeste handleidingen zijn in het engels geschreven. Hieronder staat een onlinebron waar je bijna alle informatie over Python code voor de Micro:bit kan vinden.

Onlinebron: <https://microbit.org/nl/guide/python/>