

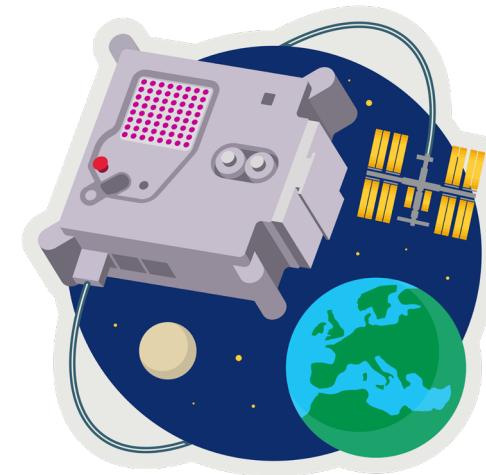
# De Astro Pi programmeren

## Mission Zero

De Astro Pi is een kleine computer aan boord van het internationale ruimtestation (ISS). En jij kunt deze vanaf jouw computer programmeren! In deze les leer je de belangrijkste commando's voor de Astro Pi. Na afloop van deze les kun jij meedoen met Mission Zero!

## Wat heb je nodig?

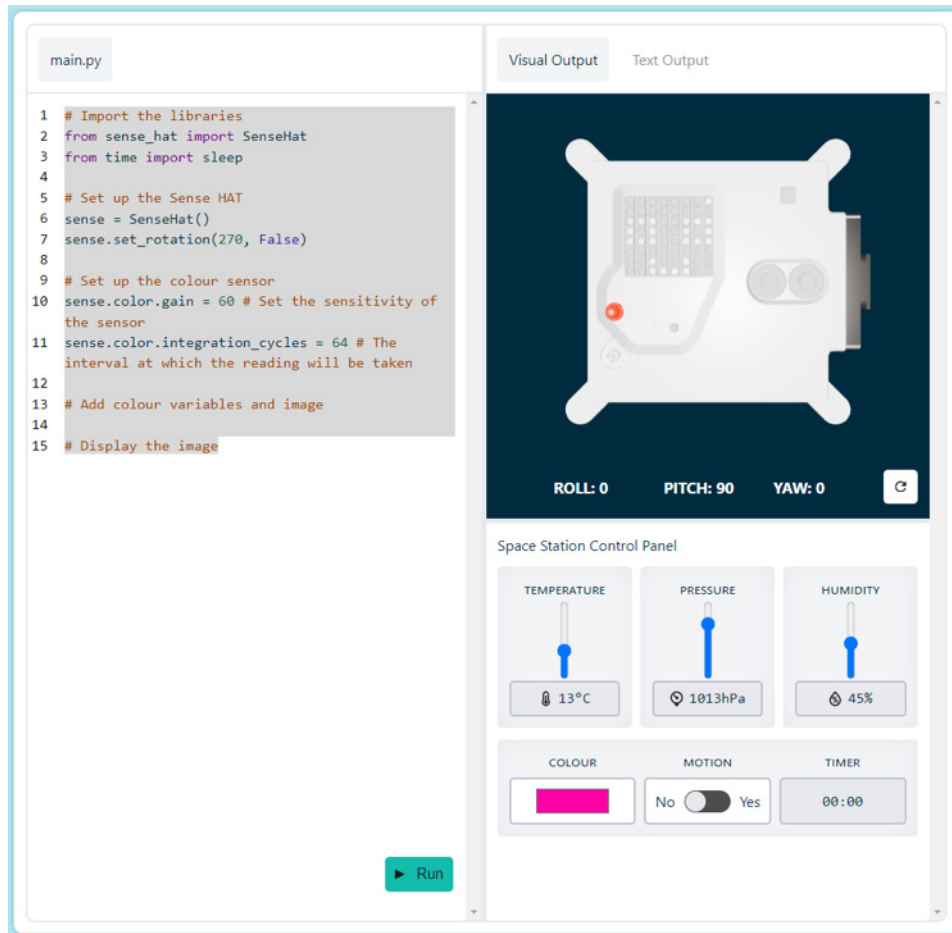
- een pc met internet



**ASTRO PI**

# Starten

Ga naar [https://missions.astro-pi.org/nl-NL/mz/code\\_submissions/new](https://missions.astro-pi.org/nl-NL/mz/code_submissions/new).  
Op het scherm zie je het onderstaande venster.



The screenshot displays the Astro Pi code editor interface. On the left, a code editor window titled 'main.py' contains the following Python code:

```
1 # Import the libraries
2 from sense_hat import SenseHat
3 from time import sleep
4
5 # Set up the Sense HAT
6 sense = SenseHat()
7 sense.set_rotation(270, False)
8
9 # Set up the colour sensor
10 sense.color.gain = 60 # Set the sensitivity of
11   the sensor
12 sense.color.integration_cycles = 64 # The
13   interval at which the reading will be taken
14
15 # Add colour variables and image
16
17 # Display the image
```

Below the code editor is a green 'Run' button. On the right, the 'Visual Output' panel shows a 3D model of the Astro Pi hardware. Below the model, the orientation is displayed as 'ROLL: 0 PITCH: 90 YAW: 0'. Underneath is the 'Space Station Control Panel' with several controls:

- TEMPERATURE: 13°C
- PRESSURE: 1013hPa
- HUMIDITY: 45%
- COLOUR: A pink color selection box.
- MOTION: A toggle switch currently set to 'No'.
- TIMER: A digital display showing '00:00'.

In het scherm links staat een code. Deze kun je wissen en vervangen door je eigen code. Rechts zie je hoe de Astro Pi reageert op jouw code. Met de stop-en-runknop onderaan kun je het programma stoppen of laten afspelen

# Wat je hebt geleerd over programmeren

Gefeliciteerd! Je weet nu alle codes die je nodig hebt om Mission Zero te voltooien. Maar behalve een aantal commando's heb je ook geleerd wat programmeren is en wat je nodig hebt om een goed programma te maken.

- Een computer doet precies wat je zegt, maar heeft geen idee wat hij doet. Dat bepaal jij helemaal.
- Computers gebruiken andere informatie dan wij. Voor ons is groen een kleur, voor de computer is 'groen' een getal of een stukje tekst.
- Je moet heel precies zijn. Een kleine spelfout kan betekenen dat je programma niet meer werkt.
- Probeer te denken als een computer: als ik dit doe dan gebeurt er dat.
- Als je een programma schrijft, ben je veel aan het uitproberen, fouten maken en verbeteren.
- Door opmerkingen bij de code te schrijven kan je beter volgen wat de code zou moeten doen en kan je vaak sneller ontdekken waar het mis gaat.
- Als je maar lang genoeg doorzet, kun je elk probleem oplossen!

Tot slot op de achterkant de allerlaatste uitdaging!

# Je eerste computerprogramma

Schrijf in de Sense HAT-emulator de volgende code:

```
#importeer de functies van de Sense Hat
from sense_hat import SenseHat

#maak een variabele waarmee je functies van de Sense Hat
#eenvoudig kunt oproepen
#dit is niet per se nodig, maar anders moet je overal
#waar sense staat SenseHat() invullen
sense = SenseHat()

#laat de volgende tekst op het display zien
sense.show_message("Groetjes van de aarde!")
```

`sense.show_message` laat tekst verschijnen op het ledscherm. Je kunt zelf bepalen wat voor tekst je invoert. Je kunt ook de scrollsnelheid veranderen. Probeer maar eens het volgende commando in te voeren:

```
sense.show_message("Groetjes van de aarde!", scroll_
speed=0.05)
```

## Meer experimenteren

Probeer ook eens het volgende programma uit:

```
from sense_hat import SenseHat
sense = SenseHat()

#importeer functies uit 'time' om de klok te gebruiken
#nodig voor time.sleep
from time import sleep

sense.show_letter("h")
sleep(0.3)
sense.show_letter("o")
sleep(0.3)
sense.show_letter("i")
```

**Stel jezelf de volgende vragen:**

- Wat zijn de verschillen tussen show\_message en show\_letter?
- Wat doet sleep?

## Meer experimenteren

```
r = (184, 134, 11) #Donkergeel
s = (139, 69, 19) #Bruin
t = (255, 140, 0) #Oranje
u = (178, 34, 34) #Donkerrood
v = (255, 0, 0) #Rood
w = (255, 192, 203) #Roze
y = (255, 20, 147) #Donkerroze
z = (153, 50, 204) #Paars
```

## Meer experimenteren

Je kan het scherm van de Astro Pi een kleur laten weergeven na je meting. Voeg twee regels toe waarbij het scherm een kleur laat zien na je meting. Maak gebruik van de verschillende waarden die je rood, groen en blauw kan geven. Het commando `sense.clear()` maakt het scherm leeg en laat een gekozen kleur zien. Als je geen kleur invult, wordt het scherm zwart.

```
#kies je eigen waarden voor rood, groen en blauw tussen
0 - 255
x = (178, 34, 34)
sense.clear(x)
```

Je kunt ook je eigen plaatje maken.

Kies de kleuren uit die je wilt gebruiken en voeg deze toe aan je code.

Om je plaatje te maken, verander je de letters tussen de blokhaken.

Kan je een slang maken? En een bloem?

```
a = (255, 255, 255) #Wit
b = (105, 105, 105) #Grijs
c = (0, 0, 0) #Zwart
d = (100, 149, 237) #Lichtblauw
e = (0, 0, 205) #Blauw
f = (25, 25, 112) #Donkerblauw
g = (0, 191, 255) #Middenblauw
h = (0, 255, 255) #Cyaan
j = (143, 188, 143) #Donker zeegroen
k = (46, 139, 87) #Zeegroen
l = (0, 255, 127) #Lichtgroen
m = (34, 139, 34) #Donkergroen
n = (154, 205, 50) #Geelgroen
o = (128, 128, 0) #Olijf
p = (240, 230, 140) #Khaki
q = (255, 255, 0) #Geel
```

## De kleuren van het scherm

Het scherm van de Sense HAT bevat 64 leds die verschillende kleuren kunnen weergeven. Elke led bestaat eigenlijk uit drie kleinere leds: een rode, een groene en een blauwe. Je kunt deze drie kleinere leds apart aansturen, zodat je alle kleuren van de regenboog kunt maken.

Schrijf het volgende programma:

```
from sense_hat import SenseHat
sense = SenseHat()

#draai het scherm een kwartslag zodat de tekst recht staat
sense.set_rotation(270)

#de drie basiskleuren van de ledlampjes
#voor elke kleur kun je een getal tussen 0 en 255 invullen
rood = (255,0,0)
groen = (0,255,0)
blauw = (0,0,255)

sense.show_message("rood", text_colour=rood)
sense.show_message("groen", text_colour=groen)
sense.show_message("blauw", text_colour=blauw)
```

Probeer nu zelf eens kleuren te maken, net als hierboven de kleuren rood, groen en blauw zijn gemaakt.

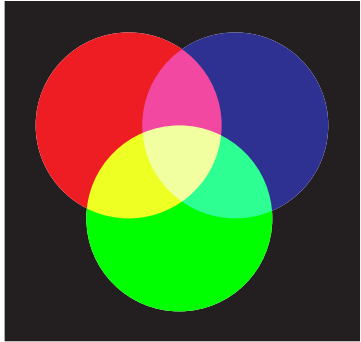
Verder kun je het getal dat hoort bij een kleur, rechtstreeks invoeren in `sense.show_message`. Hieronder een voorbeeld.

```
sense.show_message("welke kleuren zijn dit?", text_
colour=(255,255,0), back_colour=(255,0,255))

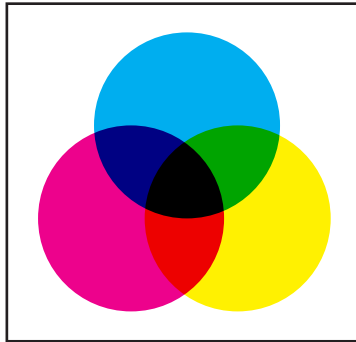
#maak het scherm weer zwart
sense.clear()
```

## Meer weten over kleuren

Er bestaan oneindig veel kleuren, maar jouw ogen kunnen er maar drie zien: rood, blauw en groen. Elke andere kleur die jij ziet, is een mengsel van deze drie kleuren. Zo is geel voor ons hetzelfde als rood plus groen. Wit is voor ons hetzelfde als alle drie de kleuren samen.



Kleuren mengen met licht



Kleuren mengen met verf

Let op: kleuren mengen met licht werkt anders dan kleuren mengen met verf. De drie basiskleuren van verf zijn magenta (rozerood), cyaan (soort blauw) en geel. Dit komt doordat je met lichtbundels kleuren toevoegt en bij verf juist kleuren filtert.

Probeer zelf nieuwe kleuren te maken door de getallen te veranderen. Welke kleur vind jij het mooist?

## De kleurensensor

De Sense HAT heeft een kleurensensor waar mee je de kleuren uit de omgeving kunt meten. Vul je code aan met onderstaande code en voer uit.

```
from sense_hat import SenseHat
from time import sleep
sense = SenseHat()
sense.clear()

sense.set_rotation(270)

w=(255,255,255)
z=(0,0,0)

for i in range(28):
    rgb = sense.colour
    z = (rgb.red,rgb.green,rgb.blue)

smiley = [
    w,w,z,z,z,z,w,w,
    w,z,w,w,w,w,z,w,
    z,w,z,w,w,z,w,z,
    z,w,w,w,w,w,w,z,
    z,z,w,w,w,w,z,z,
    z,w,z,z,z,z,w,z,
    w,z,w,w,w,w,z,w,
    w,w,z,z,z,z,w,w]

sense.set_pixels(smiley)
sleep(1)
```

Stel jezelf de volgende vragen:

- Wat doet sleep?
- Wat gebeurt er na 29 seconden?
- Hoe kan je dit veranderen?

## Maak een plaatje (2)

Via het commando `set_pixels` bestuur je het hele scherm in één keer.  
Zo hoef je niet voor elke pixel apart een commando te schrijven!

Schrijf onderstaand programma en voer uit.

```
from sense_hat import SenseHat
from time import sleep
sense = SenseHat()
sense.clear()

sense.set_rotation(270)

w=(255,255,255)
z=(0,0,0)

smiley = [
    W,W,Z,Z,Z,Z,W,W,
    W,Z,W,W,W,W,Z,W,
    Z,W,Z,W,W,Z,W,Z,
    Z,W,W,W,W,W,Z,
    Z,Z,W,W,W,Z,Z,
    Z,W,Z,Z,Z,W,Z,
    W,Z,W,W,W,Z,W,
    W,W,Z,Z,Z,W,W]

sense.set_pixels(smiley)
```

Wat gebeurt er als je de cijfers bij w verandert? En als je regel 9 weghaalt?  
Wat gebeurt er als je de '270' in regel 6 verandert in 90?

## Maak een plaatje (1)

Een scherm bestaat uit kleine puntjes, pixels. De Sense HAT heeft een LED-scherm van 8 bij 8 lampjes. Hieronder leer je hoe je de pixels kunt aansturen.

Schrijf het volgende programma

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()

#zet de x- en de y-coördinaat op 0
x=0
y=0

#zet de kleuren op maximale sterkte
r=255
g=255
b=255

sense.set_pixel(x,y,r,g,b)
```

Verander x in waarden 0 t/m 7. Wat gebeurt er op het scherm?  
Doe hetzelfde voor y.  
Waarom kun je voor x of y geen 8 of hoger invullen?