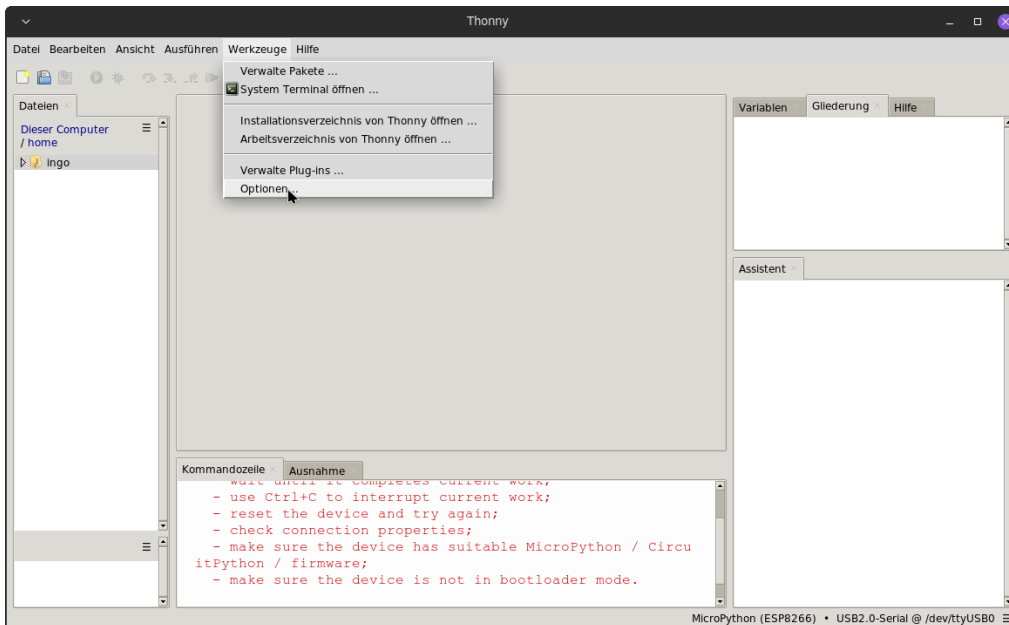


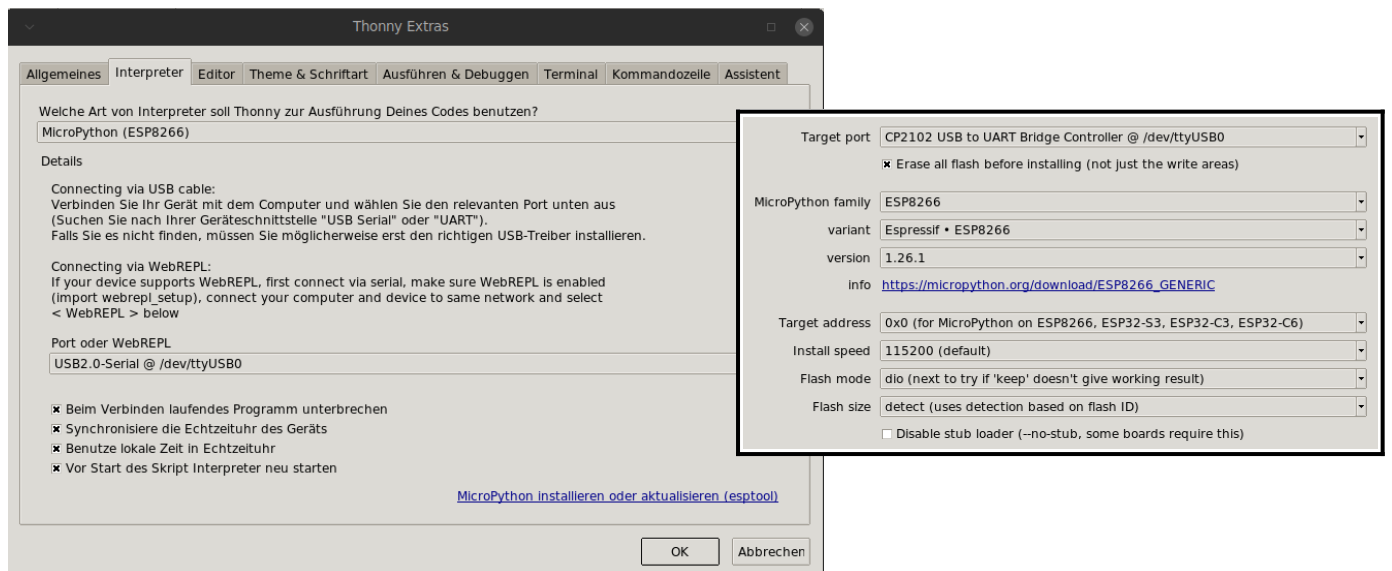
Aufgaben

Den ESP8266 einrichten

Öffne das Programm, schließe deinen ESP8266 über USB an und gehe direkt auf den Punkt Werkzeuge → Optionen:



Dort sollte unter dem Reiter „Interpreter“ euer ESP8266 automatisch erkannt werden. Falls ihr ihn nicht finden könnt, meldet ihr euch bei uns.

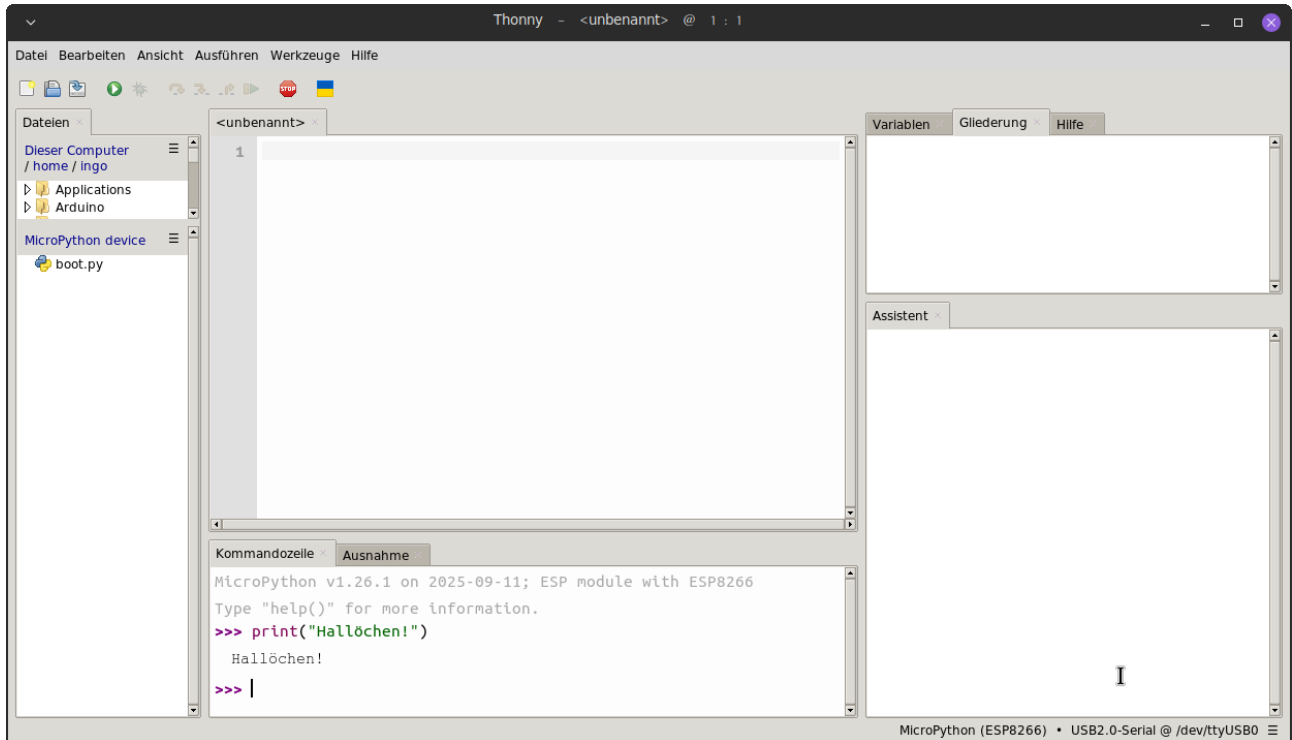


Hello World

Schreibe den Befehl

```
print („Hallöchen“)
```

in die Kommandozeile unten in Thonny hinein und drücke Enter. Erscheint dein Text in der nächsten Zeile?



Der Befehl `print()` ist mit dem „Zeige Text“-Block von Makecode vergleichbar, den wir vorher auch schon genutzt haben, um uns die Sensorwerte anzeigen zu lassen.



Aufgabe: Lege eine Variable *text* mit dem Inhalt „Hallöchen“ an:

```
text = „Hallöchen“
```

Gebe mit `print()` die Variable ausgeben und überprüfe den Inhalt.

Aufgabe: Lege die Variable *zahl* mit dem Inhalt 3 an:

```
zahl = 3
```

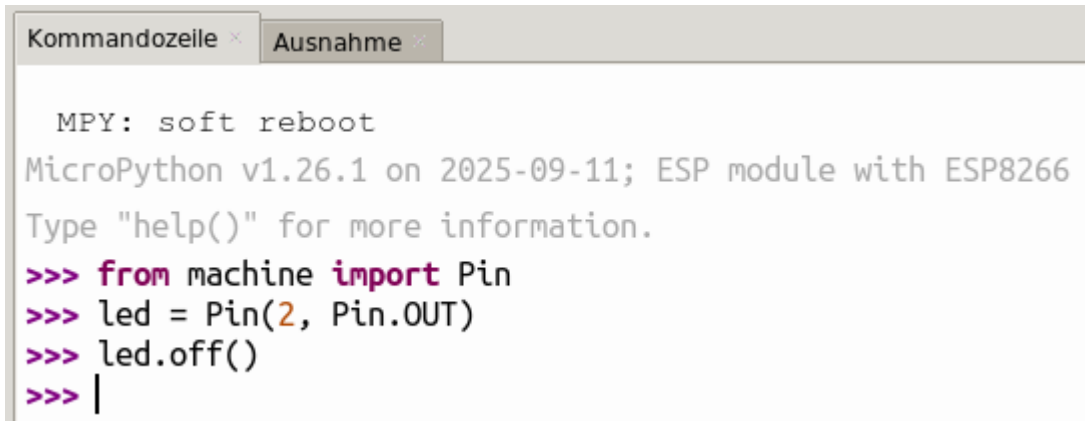
Erhöhe die Variable um 4 und gebe den neuen Wert mit `print()` aus.

Blinky LED

Füge die externe Bibliothek Pin mit der folgenden Zeile hinzu:

```
from machine import Pin
```

Bibliotheken sind so etwas wie Plugins. Ab jetzt kannst du den Befehl `Pin()` verwenden.



The screenshot shows a MicroPython REPL window with two tabs: 'Kommandozeile' (active) and 'Ausnahme'. The text in the window is as follows:

```
MPY: soft reboot
MicroPython v1.26.1 on 2025-09-11; ESP module with ESP8266
Type "help()" for more information.
>>> from machine import Pin
>>> led = Pin(2, Pin.OUT)
>>> led.off()
>>> |
```

Achtung! Die LED auf dem ESP geht mit `led.off()` an und mit `led.on()` wieder aus – es ist genau umgedreht.

Aufgabe: Probiere die anderen LEDs auf dem Pflanzensensor aus. Sie sind an die folgenden Pins angeschlossen:

grün – Pin 14

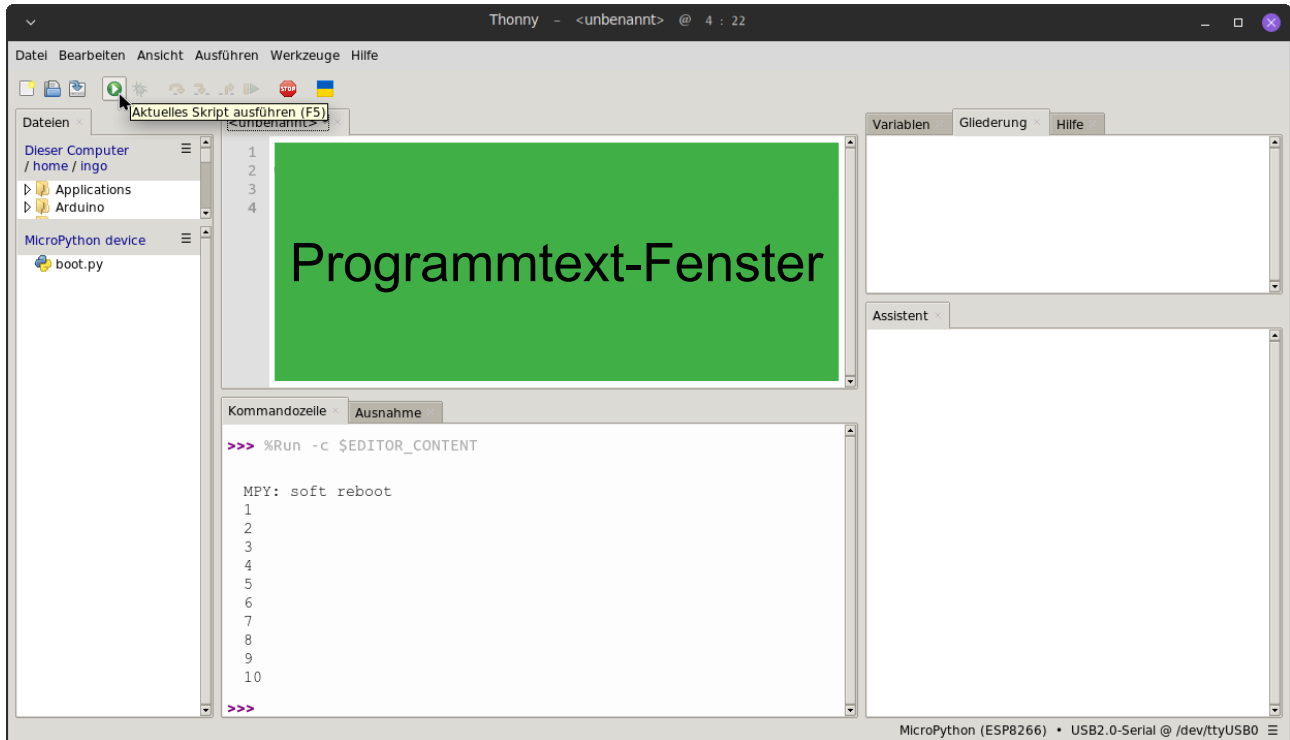
gelb – Pin 12

rot – Pin 13

Aufgabe: Was ist *led* genau? Tipp: Die Struktur mit dem = Zeichen haben wir in der letzten Aufgabe schon benutzt.

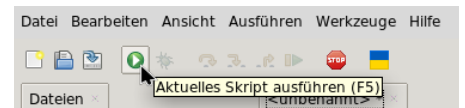
Poesie

In dem Programmtext-Fenster kannst du mehrere Befehle jeweils in eine Zeile untereinander schreiben. Diese werden der Reihe nach ausgeführt, wenn du auf den grünen Pfeil klickst.



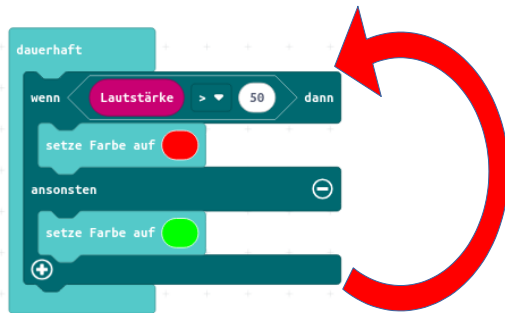
Aufgabe: Lasse dir ein selber ausgedachtes Gedicht mit 4-5 Zeilen untereinander ausgeben.

Aufgabe: Lasse nach jeder Zeile des Gedichts eine beliebige LED für jeweils eine Sekunde aufleuchten und anschließend wieder ausgehen. Dafür benötigst du den Befehl `sleep(sekunden)`, damit wartet dein ESP einfach für die in *sekunden* eingetragene Zeit, z.B. `sleep(3)` für 3s. Den Befehl `sleep()` kannst du erst benutzen, nachdem du ihn mit `from time import sleep` eingebunden hast.



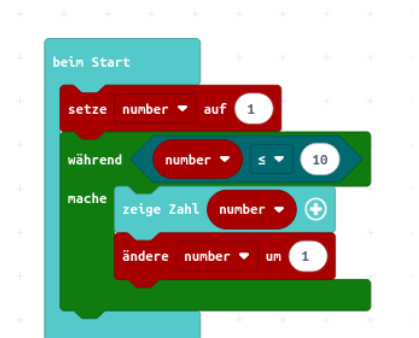
Schleifen

Eine Schleife wiederholt einen Block des Quellcodes. In Makecode habt ihr das z.B. für die Lärmampel benutzt:



In Python gibt es stattdessen die *while*-Schleife (englisch für „während“):

Aufgabe: Schreibe ein Programm, das Zahlen von 1 bis 10 ausgibt (so wie rechts). Achte in Python auf die gleichen Einrückungen für den Block!



```
1 number = 1
2 while number <= 10:
3     print(number)
4     number = number + 1
```

```
>>> %Run -c $EDITOR_CONTENT
MPY: soft reboot
1
2
3
4
5
6
7
8
9
10
>>>
```

Aufgabe: Wenn dir jetzt noch nicht klar ist, wie *while* funktioniert, Google nach weiteren Informationen oder lasse es dir von einem Chatbot erklären.

Aufgabe: Lasse jetzt die Zahlen endlos nach oben zählen, solange der ESP Strom hat (vergleichbar mit der Dauerhaft-Schleife von Makecode).

Bonus: Lasse jetzt noch bei jeder Zahl eine LED kurz aufblinken.

Den Sensor auslesen

Um den Sensor auszulesen benötigen wir die Bibliothek

```
from machine import ADC
```

Lege eine Variable `bodenfeuchte_sensor` mit dem Inhalt `ADC(0)` an.

Lese den aktuellen Messwert ein mit: `bodenfeuchte_sensor.read()`

Aufgabe: Gebe den aktuellen Messwert dauerhaft aus, solange der ESP Strom hat. Füge zwischen zwei Messungen eine kleine Pause ein, damit die Zahlen nicht zu schnell kommen.



```
1 from machine import Pin, ADC
2 from time import sleep
3
4 bodenfeuchte_sensor = ADC(0) # Sensor-Pin A0 am ESP verbinden
5 bodenfeuchte_wert = 0
6
7 while True:
8     bodenfeuchte_wert = bodenfeuchte_sensor.read()
9     print(bodenfeuchte_wert)
10    sleep(0.1)
11
```