



Lernziele:

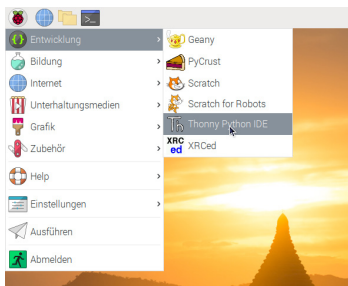
- Können die Entwicklungsumgebung (Thonny Python IDE) für Python starten.
- Können den Python Code zum Gebrauch der Kamera analysieren und nachbauen.

In der nächste Aufgabe sollt ihr nun Fotos mit Hilfe eines Programms machen. Zur Auswahl stehen die Programmiersprache Scratch oder Python. In diesem Aufgabenblatt wird Python verwendet, um die Aufgaben zu lösen.

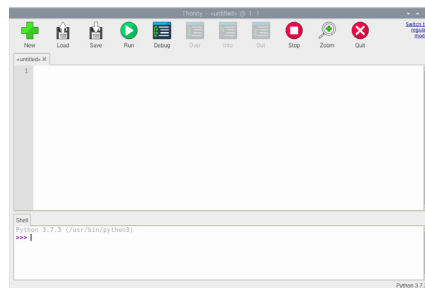
Starten der Entwicklungsumgebung. Es stehen viele Entwicklungsumgebungen für Python zur Verfügung. Hier wird die Umgebung «Thonny Python IDE» gezeigt und verwendet.

Aufgabe 1

Startet die Entwicklungsumgebung «Thonny Python IDE».



Start der Programmierungsumgebung für Python



So sieht Thonny Python IDE nach dem Start aus

Mit dem grünen «+» wird eine neue leere Programmdatei erzeugt
Mit «Load» kann ein bestehendes Programm geladen werden
Mit «Save» wird ein aktuelles Programm gespeichert
Mit «Run» wird das aktuelle Programm gestartet
«Debug» (darauf gehen wir später ein) dient der Fehlersuche
«Stop» hält das aktuelle Programm an
«Zoom» vergrößert die Programmierfläche
«Quit» beendet das Programm



Aufgabe 2 (Foto)

Analysiert zusammen folgenden Code. Nach der Analyse gebt ihr den Code ein und testet ihn. Die Kommentare, das ist der Text hinter dem «#», müsst ihr nicht eingeben. Das sind nur Erklärungen für euch.

The screenshot shows the Thonny Python IDE interface. The title bar reads 'Thonny - /home/pi/Documents/takePicture.py @ 1:99'. The toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The main editor window displays the following Python code:

```
1  #!/usr/bin/python3 # mit dieser Anweisung kann das Programm mit einem Doppelklick gestartet werden|
2
3  # An dieser Stelle werden die nötigen Programmteile geladen
4  # picamera - damit Python mit der Kamera das Raspberry Pi Kontakt aufnehmen kann
5  # time - damit eine Pause von 3 Sekunden eingebaut werden kann
6  from picamera import PiCamera
7  from time import sleep
8
9  # Python mit der Kamera verbinden. Die Kamera hat den Namen camera
10 camera = PiCamera()
11
12 # Ein Vorschaufenster starten, damit wir sehen, was aufgenommen wird
13 camera.start_preview()
14
15 # Die Vorschau soll 3 Sekunden gezeigt werden. Darum wird 3 Sekunden Pause
16 # gemacht, bis die nächste Anweisung ausgeführt wird
17 sleep(3)
18
19 # Ein Bild aufnehmen und es auf dem Desktop speichern
20 camera.capture('/home/pi/Desktop/bild1.jpg')
21
22 # Die Vorschau des Fotos beenden
23 camera.stop_preview()
```

Below the code editor is a 'Shell' window showing the Python 3.7.3 prompt and the command prompt:

```
Python 3.7.3 (/usr/bin/python3)
>>>
```

The bottom right corner of the IDE window indicates 'Python 3.7.3'.

Sollte das Programm nicht stoppen wollen, könnt ihr das über die Tastenkombination «alt-f4» tun. Das Problem ist dann sehr wahrscheinlich, dass das Foto nicht gespeichert werden kann. Überprüft, ob der Pfad für die Speicherung des Bildes existiert.



Aufgabe 3 (Video)

Analysiert zusammen folgenden Code. Nach der Analyse gebt ihr den Code ein und testet ihn. Die Kommentare, das ist der Text hinter dem «#», müsst ihr nicht eingeben. Das sind nur Erklärungen für euch.

```
Thonny - /home/pi/Documents/takeVideo.py @ 27:22

New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode

takeVideo.py x
1  #!/usr/bin/python3 # mit dieser Anweisung kann das Programm mit einem Doppelklick gestartet werden
2
3  # An dieser Stelle werden die nötigen Programmteile geladen
4  # picamera - damit Python mit der Kamera das Raspberry Pi Kontakt aufnehmen kann
5  # time - damit eine Pause von 3 Sekunden eingebaut werden kann
6  from picamera import PiCamera
7  from time import sleep
8
9  # Python mit der Kamera verbinden. Die Kamera hat den Namen camera
10 camera = PiCamera()
11
12 # Ein Vorschaufenster starten, damit wir sehen, was aufgenommen wird
13 camera.start_preview()
14
15 # Mit der Aufzeichnung beginnen und den Namen der Videodatei angeben
16 camera.start_recording("/home/pi/Desktop/video.h264")
17
18 # Die Vorschau soll 5 Sekunden gezeigt werden. Darum wird 5 Sekunden Pause
19 # gemacht, bis die nächste Anweisung ausgeführt wird
20 # Das Video wird nicht 5 Sekunden lang sein
21 sleep(5)
22
23 # Die Aufzeichnung anhalten
24 camera.stop_recording()
25
26 # Die Vorschau des Fotos beenden
27 camera.stop_preview()

Shell
Python 3.7.3 (/usr/bin/python3)
>>> |

Python 3.7.3
```

Das Video könnt ihr euch mit dem VNC-Viewer ansehen (einfach auf dem Video doppelklicken).

Wenn ihr mehrere Videos machen wollt, müsst ihr den Namen der Videodatei ändern. Sonst wird das Video jedesmal überschrieben.