

Examen VMBO-KB

2022

voorbeeldopdrachten MVI – programmeren in Python

opdrachten

Opdracht 1 Teken een complexe vorm

Programmeer een complexe vorm in Python met een Turtle module.

Werkwijze

- Open de Python shell.
- Importeer de Turtle module.
- Maak nu een scherm om op te tekenen.
- Maak de functie **jouwster** aan. Let op blok code, gebruik de TAB-toets.

Stap 1

Voer de volgende codes in:

```
>>> def jouwster(grootte, gevuld):  
    if gevuld == True:  
        t.begin_fill()  
        for x in range(1, 19):  
            t.forward(grootte)  
            if x % 2 == 0:  
                t.left(175)  
            else:  
                t.left(225)  
        if gevuld == True:  
            t.end_fill()
```

Stap 2

- De for lus ontbreekt. Maak de code compleet.
- De Turtle laten we vervolgens een **gouden ster** tekenen.
 - `t.color(0.9, 0.75, 0)`
 - `jouwster(120,True)`
- Teken een zwart lijntje om de ster.
- Sla het bestand op als `gouden_ster_jouwnaam.py` .

Opdracht 2 Alarmsysteem maken

Jij krijgt op je stageadres een opdracht: je gaat een simpel alarmsysteem maken. Op je stage gebruiken ze daarvoor een Raspberry Pi en het programma Python 3. Een collega is al begonnen. Jij gaat het programma aanpassen, omdat er nieuwe eisen zijn bij gekomen.

Werkwijze

Vorbereiding

- Sluit deze onderdelen aan op de Grove Base HAT:
 - op poort D5 de Red LED Button
 - op poort D16 de Mini PIR Motion Sensor
 - op poort D18 de Buzzer
- Sluit daarna de voeding aan op de Raspberry Pi.
- Bekijk deze code:

```
#!/bin/python
import RPi.GPIO as GPIO # importeer de GPIO bibliotheek.
from time import sleep # importeer de time bibliotheek voor tijdfuncties.

GPIO.setmode(GPIO.BCM) # Zet de pinmode op Broadcom SOC.
GPIO.setwarnings(False) # Zet waarschuwingen uit.

led=5 # GPIO pin als LED variabele declareren
pir=16 # GPIO pin als PIR variabele declareren
buzzer=18 # GPIO pin als BUZZER variabele declareren

GPIO.setup(buzzer, GPIO.OUT) # Zet de BUZZER pin als uitgang
GPIO.setup(led, GPIO.OUT) # Zet de LED pin als uitgang
GPIO.setup(pir, GPIO.IN) # Zet de PIR pin als ingang

def alarm(duur,pauze):
    GPIO.output(led, 1) # zet de GPIO pin van de LED poort hoog (1 = licht aan)
    GPIO.output(buzzer,1) # zet de GPIO pin van de BUZZER poort hoog (1 = piepen)
    print("alarm duur" ,duur, "pauze" ,pauze )
    sleep(duur) # wachten (duur)
    GPIO.output(led, 0) # zet de GPIO pin van de LED poort laag (0 = licht uit)
    GPIO.output(buzzer, 0) # zet de GPIO pin van de BUZZER poort laag (0 = piepen uit)
    sleep(pauze) # wachten (pauze)

while True:
    if GPIO.input(pir)== 0: # als GPIO pin van de PIR poort laag (0 = geen beweging) is
        alarm(0.01,2)
        print("PIR = 0, geen beweging...") # print tekst
    else:
        alarm(0.01,0.1)
        print("PIR <> 0, wel beweging...") # print tekst
```

Stap 1

- Kies in het menu: Programmeren, Thonny Python IDE
- Kies in Python: Load en laad het bestand KB_alarm.py
- Run het programma.
- Stop het programma als je de werking kent.

Het programma maakt gebruik van deze functie:

```
def alarm(duur,pauze):
    GPIO.output(led, 1) # zet de GPIO pin van de LED poort hoog (1 = licht aan)
    GPIO.output(buzzer,1) # zet de GPIO pin van de BUZZER poort hoog (1 = piepen)
    print("alarm duur" ,duur, "pauze" ,pauze )
    sleep(duur) # wachten (duur)
    GPIO.output(led, 0) # zet de GPIO pin van de LED poort laag (0 = licht uit)
    GPIO.output(buzzer, 0) # zet de GPIO pin van de BUZZER poort laag (0 = piepen uit)
    sleep(pauze) # wachten (pauze)
```

De functie gebruikt twee parameters: *duur* en *pauze*.

In de *while* lus wordt de functie aangeroepen:

- als er geen alarm is met *duur* 0.01 seconden en *pauze* 2 seconden.
- als er wel alarm is met *duur* 0.01 seconden en *pauze* 0.1 seconde.

Stap 2

- Verbeter de functie door een derde parameter toe te voegen: *tekst*
- Verander de regels in de *while* lus naar:
 - `alarm(0.01,2,"PIR = 0, geen beweging...")`
 - `alarm(0.01,0.1,"PIR <> 0, wel beweging...")`
- Test het programma.
- Sla het bestand op onder de naam `alarm_jouw naam.py`
- Laat aan je examiner zien dat je programma werkt.