

Examen VMBO-BB

# 2022

**voorbeeldopdrachten MVI – programmeren in Python**

opdrachten

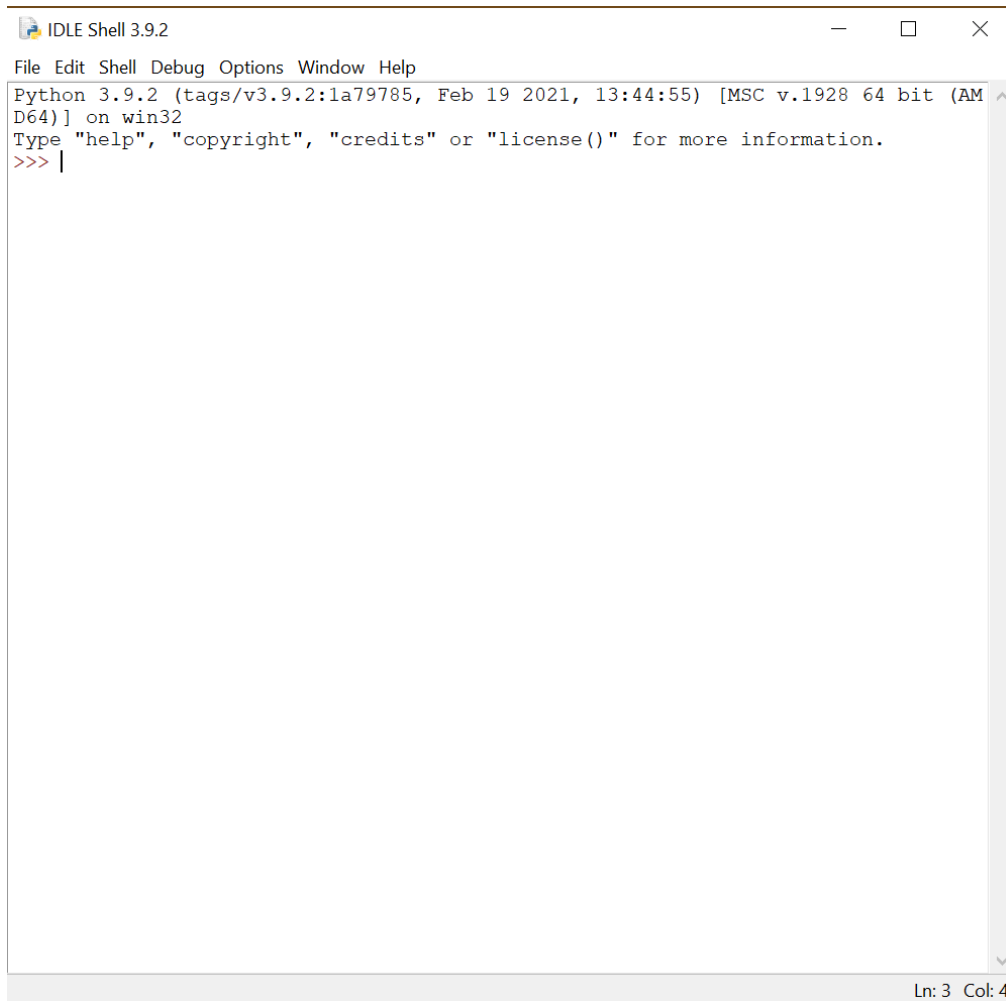
---

## Opdracht 1 Teken een vierkant

Programmeer een vierkant in Python met de Turtle module.

### Werkwijze

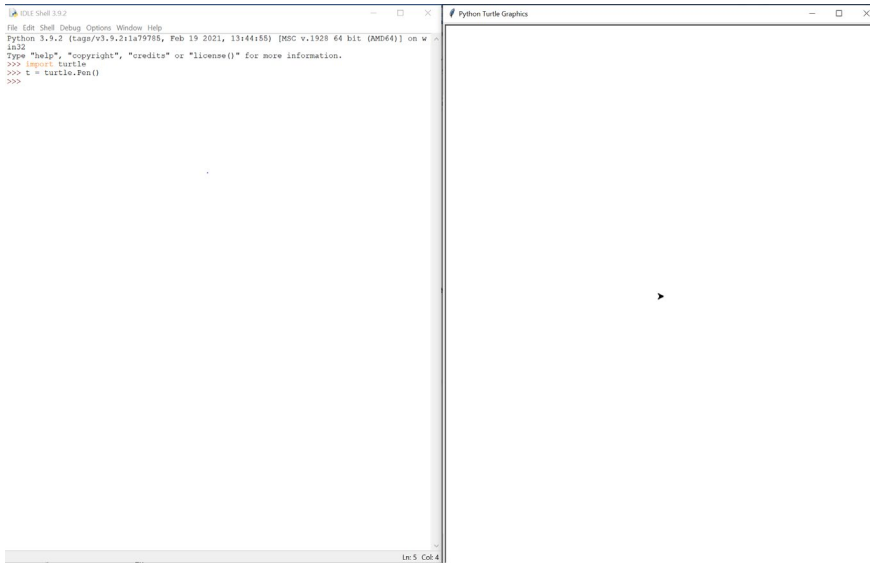
- Open de Python shell.



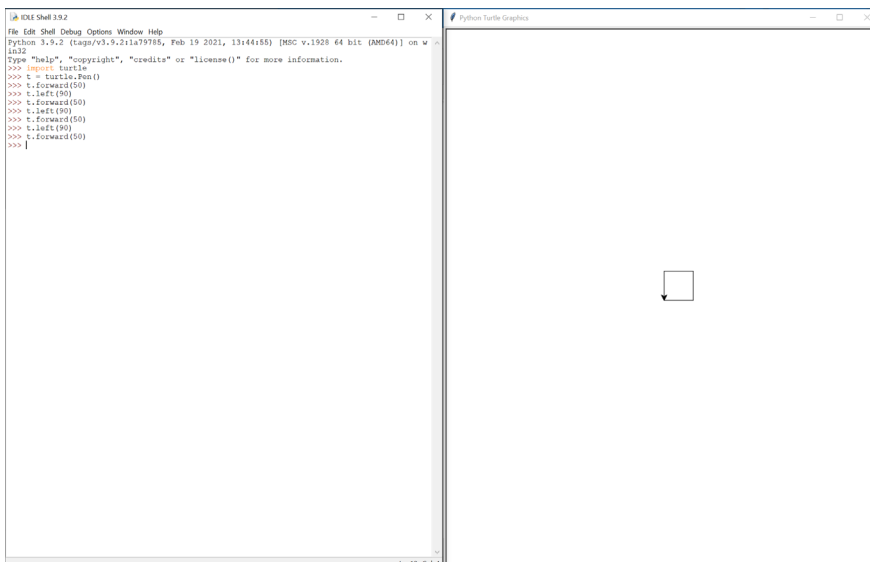
The screenshot shows the IDLE Shell 3.9.2 window. The title bar reads 'IDLE Shell 3.9.2'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following text: 'Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32', 'Type "help", "copyright", "credits" or "license()" for more information.', and a prompt '>>>' followed by a vertical bar cursor. The status bar at the bottom right indicates 'Ln: 3 Col: 4'.

- Importeer de Turtle module.
  - Typ >>>import turtle

- Maak een scherm waar op getekend kan worden.
  - Typ `>>>t = turtle.Pen()`



- Teken nu een vierkant.
- Voer de volgende codes in:
  - Typ `>>>t.forward(50)`
  - Typ `>>>t.left(90)`
  - Typ `>>> t.forward(50)`
  - Typ `>>>t.left(90)`
  - Typ `>>> t.forward(50)`
  - Typ `>>>t.left(90)`
- De Turtle heeft nu een vierkant getekend.



- Reset het scherm en laat de Turtle terugkeren naar de startpositie.
  - Typ `>>>t.reset()`

## 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 verder afmaken.

### 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 BB_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

while True:

    if GPIO.input(pir)== 0: # als GPIO pin van de PIR poort laag (0 = geen beweging) is
        print("PIR = 0, geen beweging...") # print tekst

        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)

        sleep(0.01) # 1/100 seconde wachten

        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(1) # 2 seconden wachten

    else:
        print("PIR <> 0, wel beweging...") # print tekst

        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)

        sleep(0.01) # 1/100 seconde wachten

        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(0.1) # 1/10 seconde wachten
```

#### Stap 1

- Kies in het menu: Programmeren, Thonny Python IDE
- Kies in Python: Load en laad het bestand BB\_alarm.py
- Bekijk het programma.
- Run het programma: druk op de groene knop.
- Controleer het resultaat: beweging wordt wel gezien, maar bij beweging hoor je nog geen snellere piepjes.
- Stop het programma als je de werking kent: druk op de rode knop.

## Stap 2

Het deel na `e/se:` is nog niet af.

- Gebruik het programma hierboven om het programma verder af te maken.
- Test het programma: controleer of de snellere piepjes het bij beweging nu wel doen.
- Sla het bestand op onder de naam `alarm_jouw naam.py`
- Laat aan je examiner zien dat je programma werkt.