

Python Functieopdrachten – Systeembeheer (MBO 4)

Opdracht 1: Schijfruimte controleren

Doel: Werken met functies en modules.

Schrijf een functie `controleer_schijfruimte()` die controleert hoeveel vrije ruimte er beschikbaar is.

Voorbeeldcode:

```
import shutil

def controleer_schijfruimte():
    totale, gebruikt, vrij = shutil.disk_usage("C:/")
    vrij_gb = vrij / (1024 ** 3)
    if vrij_gb < 5:
        print(f"Waarschuwing: minder dan 5 GB beschikbaar! ({vrij_gb:.1f} GB)")
    else:
        print(f"Vrije ruimte op C:/ is {vrij_gb:.1f} GB - voldoende.")
```

`controleer_schijfruimte()`

Opdracht 2: Gebruikers aanmaken (simulatie)

Doel: Functies met parameters schrijven.

Maak een functie `maak_gebruiker_aan(gebruikersnaam)` die een gebruiker aanmaakt (gesimuleerd).

Voorbeeldcode:

```
def maak_gebruiker_aan(gebruikersnaam, wachtwoord="Welkom123"):
    print(f"Gebruiker '{gebruikersnaam}' succesvol aangemaakt.")
    print(f"Wachtwoord ingesteld op: {wachtwoord}")
```

```
maak_gebruiker_aan("student123")
maak_gebruiker_aan("admin", "Beheerder2025")
```

Opdracht 3: Logbestanden opschonen

Doel: Werken met bestandsbeheer via functies.

Schrijf een functie `verwijder_oude_logs(pad)` die `.log`-bestanden ouder dan 30 dagen verwijdert.

Voorbeeldcode:

```
import os
import time

def verwijder_oude_logs(pad):
    nu = time.time()
    grensdatum = nu - (30 * 24 * 60 * 60)
    for bestand in os.listdir(pad):
        if bestand.endswith(".log"):
            volledig_pad = os.path.join(pad, bestand)
            aanmaak_tijd = os.path.getmtime(volledig_pad)
            if aanmaak_tijd < grensdatum:
                os.remove(volledig_pad)
                print(f"Verwijderd: {bestand}")
            else:
                print(f"Bewaard: {bestand} (recent)")
```

```
verwijder_oude_logs("C:/logs")
```

Opdracht 4: Systeemrapport genereren

Doel: Functies combineren en rapporteren.

Maak een functie `genereer_systeemrapport()` die RAM, CPU-info en IP-adres toont.

Voorbeeldcode:

```
import psutil
import socket

def genereer_systeemrapport():
    ram = psutil.virtual_memory().total / (1024 ** 3)
    cpu_info = psutil.cpu_freq()
    ip = socket.gethostbyname(socket.gethostname())

    print("Systeemrapport:")
    print(f"RAM: {ram:.1f} GB")
    print(f"CPU snelheid: {cpu_info.current:.2f} MHz")
    print(f"IP-adres: {ip}")

genereer_systeemrapport()
```