

# examen opdracht 3

- [Nieuwe pagina](#)
- [Oefenexamen met uitwerking](#)

# Nieuwe pagina

```
# =====
```

```
# Opdracht 4
```

```
# =====
```

```
# We starten een for-loop die loopt van 1 t/m 4  
for getal in range(1, 5):
```

```
    # Als het getal gelijk is aan 3
```

```
    if getal == 3:
```

```
        # Print de stopmelding
```

```
        print("Bij 3 stopt de loop")
```

```
        # Stop de loop met break
```

```
        break
```

```
    # Als het getal niet 3 is, print het getal
```

```
    print("Het getal is nu", getal)
```

# Oefenexamen met uitwerking

## ? Opdracht 1

**Maak één variabele aan voor het pad** en gebruik die variabele om de bestanden te openen.

“△ Let op: in Windows-paden gebruiken we **dubbele backslashes** `\\` of een **raw string** (`r"..."`), anders denkt Python dat `\P` of `\G` een speciale code is.

```
# We maken één variabele aan waarin het pad naar de map staat
# Dit voorkomt herhaling en maakt de code overzichtelijker
pad = r"C:\Python\Gegevens\Overzichten"
```

```
# We openen het bestand Docenten.csv
# Door het pad + bestandsnaam te combineren, weten we exact waar het bestand staat
DocentenInput = open(pad + "\\Docenten.csv", "r")
```

```
# We openen het bestand Studenten.csv
# Ook hier gebruiken we dezelfde pad-variabele
StudentenInput = open(pad + "\\Studenten.csv", "r")
```

---

## ? Opdracht 2

**Vraag aan de gebruiker of Studenten.csv leeg gemaakt moet worden**

Sla het antwoord (`j`) of (`n`) op in de variabele **wissen**.

```
# We vragen de gebruiker of het bestand Studenten.csv geleegd moet worden
# De input() functie wacht tot de gebruiker iets intypt en op Enter drukt
```

```
wissen = input("Moet het bestand Studenten.csv leeg gemaakt worden? (j/n): ")
```

```
# De variabele 'wissen' bevat nu de keuze van de gebruiker
```

```
# Bijvoorbeeld: 'j' of 'n'
```

Het bestand **Studenten.csv wordt echt leeg gemaakt** als de gebruiker  invoert.

---

## ? Opdracht 2 (uitgebreid en werkend)

```
# We maken opnieuw het pad aan naar de map waar de bestanden staan
```

```
# Dit pad wordt gebruikt om het juiste bestand te vinden
```

```
pad = r"C:\Python\Gegevens\Overzichten"
```

```
# We vragen de gebruiker of het bestand Studenten.csv leeg gemaakt moet worden
```

```
# De invoer wordt opgeslagen in de variabele 'wissen'
```

```
wissen = input("Moet het bestand Studenten.csv leeg gemaakt worden? (j/n): ")
```

```
# We controleren of de gebruiker 'j' heeft ingevoerd
```

```
if wissen == "j":
```

```
    # Als de gebruiker 'j' kiest, openen we het bestand in schrijfmodus ('w')
```

```
    # De 'w'-modus wist automatisch de inhoud van het bestand
```

```
    StudentenInput = open(pad + "\\Studenten.csv", "w")
```

```
    # We sluiten het bestand direct weer
```

```
    # Het bestand is nu leeg
```

```
    StudentenInput.close()
```

```
    # We geven een bevestiging aan de gebruiker
```

```
    print("Het bestand Studenten.csv is succesvol leeg gemaakt.")
```

```
# Als de gebruiker 'n' invoert, doen we niets met het bestand
```

```
elif wissen == "n":
```

```
    # We laten weten dat het bestand niet is aangepast
```

```
    print("Het bestand Studenten.csv is niet leeg gemaakt.")
```

```
# Als de gebruiker iets anders invoert dan 'j' of 'n'
```

```
else:
```

```
# We geven een foutmelding
print("Ongeldige invoer. Gebruik 'j' of 'n'.")
```

---

## ? Belangrijke uitleg (examenwaardig)

- `open(..., "w")`
    - wist automatisch alle inhoud
    - maakt het bestand leeg
    - maakt het bestand aan als het nog niet bestaat
  - **Waarom meteen sluiten?**
    - we hoeven niets te schrijven
    - alleen leegmaken is genoeg
  - **Waarom `if / elif / else`?**
    - duidelijke logica
    - voorkomt fouten
    - makkelijk te volgen voor een docent
- 

## ? Opdracht 3

### Sluit de bestanden die zijn geopend

```
# We sluiten het bestand Docenten.csv
# Dit is belangrijk om geheugen vrij te maken
DocentenInput.close()
```

```
# We sluiten het bestand Studenten.csv
# Bestanden altijd sluiten na gebruik
StudentenInput.close()
```

---

# ? Opdracht 4

**Pas de for-loop aan zodat deze stopt bij getal 3**

Originele loop:

```
for getal in range(1, 5):  
    print("Het getal is nu", getal)
```

Aangepaste versie met `break`:

```
# We starten een for-loop die telt van 1 t/m 4  
for getal in range(1, 5):  
  
    # We controleren of het getal gelijk is aan 3  
    if getal == 3:  
        # Als het getal 3 is, printen we een speciale melding  
        print("Bij 3 stopt de loop")  
  
        # Met break stoppen we de loop direct  
        break  
  
    # Zolang het getal geen 3 is, printen we dit  
    print("Het getal is nu", getal)
```

## ? Output:

```
Het getal is nu 1  
Het getal is nu 2  
Bij 3 stopt de loop
```

---

## ? Samenvatting (handig voor je examen)

- **Variabele voor paden** → overzichtelijk & herbruikbaar
- **input()** → invoer van gebruiker opslaan
- **.close()** → bestanden netjes afsluiten

- **break** → een loop voortijdig stoppen

Als je wilt, kan ik dit ook:

- in **één compleet script** zetten
- uitbreiden met **foutafhandeling**
- of herschrijven in **examen-spiekstijl** (superkort & duidelijk)