



# Python

---

Informatica

Renske Smetsers



# Hoofdstuk 4: variabelen

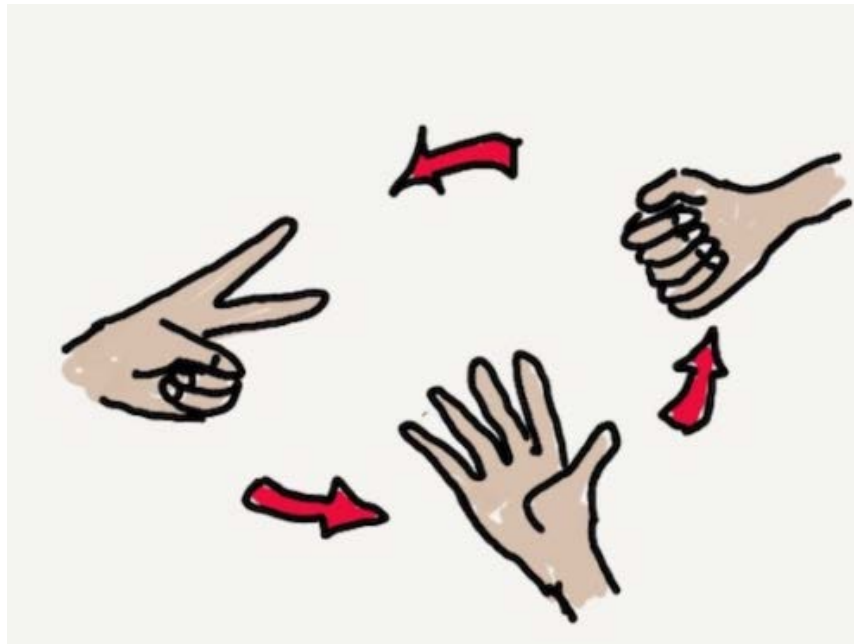
---

Leerdoelen:

- ❑ Wat variabelen zijn
- ❑ Het toekennen van een waarde aan een variabele
- ❑ Correcte namen voor variabelen
- ❑ Conventies met betrekking tot variabele namen
- ❑ Soft typing
- ❑ Het debuggen van code waarin variabelen onverwachte waarden hebben
- ❑ Verkorte operatoren
- ❑ Commentaar

# Schaar-papier-steen

- In drietallen: Speel het spel (1 minuut)
- Houd (op papier) bij hoe vaak ieder wint en hoe vaak gelijk gespeeld



# Variabelen en constanten

- ▣ **Variabelen** veranderen gedurende het spel

*Score/Aantal keren gewonnen*

```
speler1_aantal_gewonnen = 6
```

```
speler2_aantal_gewonnen = 4
```

```
speler1_aantal_gewonnen += 1
```

- ▣ **Constanten** blijven gedurende het spel hetzelfde

*Namen*

```
SPELER1 = "Mary"
```

```
SPELER2 = "Doug"
```

	Game 1		
		Win	Tie
○	Mary		
	Doug		
○	Game 2		
		Win	Tie
	Mary		
○	Doug		



# Variabelen

- ❑ Variabelen: om informatie in op te slaan.
- ❑ Geef variabelen een zinnige naam
  - `mijn_salaris_per_uur = 4.50`
  - `aantal_uren_gewerkt = 4`
- ❑ Constanten: hoofdletters voor wat tijdens programma niet verandert:
  - `DAGEN_PER_JAAR = 365`
  - `PI = 3.14`
  - `MINIMUM_SALARIS_PER_UUR = 3.04`
- ❑ Gebruik variabelen in een berekening in plaats van de getallen
  - Makkelijker om later aan te passen
  - Makkelijker om te begrijpen wat er gebeurt

```
a = 3.14159265
b = 7.5
c = 8.25
d = (a * b * b * c) / 3

print( d )
```

```
PI = 3.14159265
straal = 7.5
hoogte = 8.25
volume_van_kegel = (PI * straal * straal * hoogte) / 3
print( volume_van_kegel )
```



# Opdracht op papier: Toewijzing

1) Geef de variabele `aantal_autos` de waarde 99

```
aantal_autos = 99
```

2) Geef de variabele `aantal_brommers` de waarde 10

```
aantal_brommers = 10
```

3) Bereken `aantal_wielen` (zonder getallen 99 en 10 te gebruiken)

```
aantal_wielen = 4*aantal_autos + 2*aantal_brommers
```

4) Wat is de waarde van `aantal_boeken` na het uitvoeren van deze code:

```
aantal_boeken = 3  
aantal_boeken = aantal_boeken + 4
```

```
aantal_boeken heeft de waarde 7
```

5) `aantal_honden` is 5. Dan voeren we deze code uit:

```
aantal_dieren = aantal_honden + 3
```

Wat is nu de waarde van `aantal_honden`? En `aantal_dieren`?

```
aantal_honden is nog steeds 5, aantal_honden veranderd niet,  
aantal_dieren heeft de waarde 8
```



# Demo: door een programma heen stappen

---

□ <DEMO>

Voorbeeld code om doorheen te stappen:

```
snelheid_kmu = 14
aantal_uren = 2
afgelegde_afstand = snelheid_kmu * aantal_uren
print(afgelegde_afstand )
```



# Verkorte rekenoperatoren

---

Operator	code	Wat er gebeurt
Optellen	<code>x += 1</code>	x krijgt de waarde van x+ 1
Aftrekken	<code>x -= 2</code>	x krijgt de waarde van x-2
Vermenigvuldigen	<code>x *= 2</code>	x krijgt de waarde van x*2





# Vragen: Toewijzingen en berekeningen

1) `aantal_appels` is 5. Wat is `aantal_appels` na:

```
aantal_appels = aantal_appels + 3
```

```
aantal_appels heeft de waarde 8
```

2) `aantal_appels` is 5.

Wat is de waarde van `aantal_appels` en `aantal_appels` na:

```
aantal_fruit = aantal_appels
```

```
aantal_fruit += 1
```

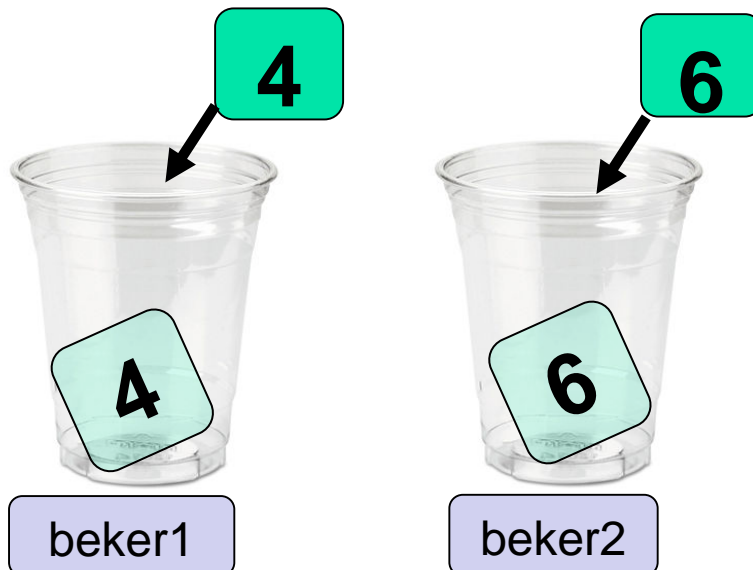
```
aantal_appels heeft de waarde 5  
aantal_fruit heeft de waarde 6
```

3) Schrijf code om het `aantal_vliegen` met 1 te verlagen

```
aantal_vliegen -= 1  
OF  
aantal_vliegen = aantal_vliegen - 1
```

# Taak: verwissel de waarde van twee dingen

- Je begint met:
  - beker1 krijgt de waarde 4, beker2 krijgt de waarde 6
- Regels:
  - 1 getal per beker
  - een nieuw waarde schopt de ander eruit
  - Waarden worden gekopieerd (niet verplaatst)
  - Zeg: “beker ... krijgt dezelfde waarde als beker ...”



beker1 = 4  
beker2 = 6

# Wisseltruc (unplugged)

- Regels:
  - 1 getal per beker
  - Een nieuw waarde schopt de ander eruit
  - Waarden worden gekopieerd (niet verplaatst)
  - Zeg: “beker ... krijgt dezelfde waarde als beker ...”
- Tip: Je mag een extra beker gebruiken.



beker1



beker2



beker\_tijdelijk

# Wisseltruc (unplugged)

- Er zijn meerdere manieren om dit te doen.
- Probeer het eerst met bekertjes
- Schrijf dan de code.

```
beker1 = 4
beker2 = 6
print( "beker1 heeft", beker1, "en beker2 heeft", beker2 )
# hier komen een aantal regels code
print( "nu heeft beker1:", beker1, "en beker2:", beker2 )
```



beker1



beker2



beker\_tijdelijk

# Wisseltruc (unplugged)

- Dit kan op meerdere manieren.
- Voorbeeld:
  - **beker\_tijdelijk** krijgt dezelfde waarde als **beker1**
  - **beker1** krijgt dezelfde waarde als **beker2**
  - **beker2** krijgt dezelfde waarde als **beker\_tijdelijk**
- Resultaat:




beker1



beker2



beker\_tijdelijk



# Wisseltruc (unplugged)

- Voorbeeld oplossing:
  - **beker\_tijdelijk** krijgt dezelfde waarde als **beker1**
  - **beker1** krijgt dezelfde waarde als **beker2**
  - **beker2** krijgt dezelfde waarde als **beker\_tijdelijk**

```
beker1 = 4
beker2 = 6
print( "beker1 heeft", beker1, "en beker2 heeft", beker2 )
beker_tijdelijk = beker1
beker1 = beker2
beker2 = beker_tijdelijk
print( "nu heeft beker1:", beker1, "en beker2:", beker2 )
```