




# Hoofdstuk 3: Expressies

---

## Leerdoelen:

- Data types string, integer en float
- Berekeningen
- Basale string expressies
- Type casting naar
  - tekst met `str()`
  - geheel getal met `int()`
  - kommagetal met `float()`



# Types

---

- Wat hoort bij wat? Maak groepjes:

|       |       |      |
|-------|-------|------|
| fiets | 12.1  | True |
| 42    | False | 5    |
| bus   | 0.334 | auto |

- Types:
  - integer: gehele getallen (2, -4, 0)
  - float: kommagetallen (1.0, -0.2) *let op de punt*
  - boolean: True / False
  - string: een reeks letters (woorden, zinnen)



# Dingen afdrukken

- Met `print` druk je af in de console (venster onderaan scherm)
- Tekst gebruik je met aanhalingstekens: **“fiets”** of **‘fiets’**
- Getallen gebruik je gewoon: **4**
- Om teksten aan elkaar te plakken gebruik je: **+** of **,**
- Getallen combineren met tekst?

Zet getal eerst om in tekst met: **str(..)**

```
print( “Ik heb een fiets.” )
print( ‘Ik heb een fiets.’ )
print( 4 )
print( “Ik heb een fiets” + “of twee” )
print( “Ik heb een fiets of” + str(2) )
```



# Met getallen kun je rekenen

```
print( "optelling: ", 2 + 3 )  
print( "aftrekking: ", 2 - 3 )  
print( "vermenigvuldiging: ", 2 * 3 )  
print( "deling: ", 2 / 3 )  
print( "machtsverheffing: ", 2 ** 3 )
```



# Met getallen kun je rekenen

```
print( "integer deling: ", 2 // 3 )  
print( "modulo: ", 2 % 3 )
```

- Integer deling // : deel en gooi de rest weg
  - $12 / 5 = 2.4$  maar  $12 // 5 = 2$
- Modulo: klok rekenen
  - 21 uur = 9 uur (steeds 12 aftrekken)
  - Dus  $21 \% 12 = 9$



# Ingewikkeldere berekeningen

- Net als met je rekenmachine: gebruik haakjes
- Wat doet deze code?

```
print( 1 + 2 / 3 )
```

- Drukt 1.6666... af, want  $1 + (2/3) = 1.6666$
- Wil je eerst dat 1 en 2 opgeteld worden? Gebruik dan haakjes:

```
print( (1 + 2) / 3 )
```

- Dit drukt 1.0... af, want  $(1+2)/3 = 3/3 = 1$



# Dingen onthouden: variabelen!

---

- Een programma kan dingen **onthouden**
  - Voorbeelden: gebruikersinvoer, een berekende waarde
- Data kan ook veranderen
- Een **variabele** is dat wat een programma moet onthouden.

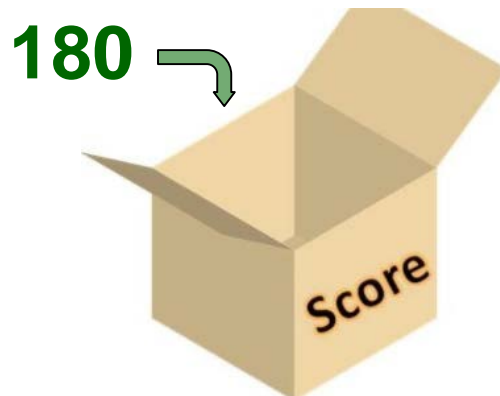
# Variabelen: naam, type en waarde

**Naam:** label / wat voor iets is het

**Type:** wat voor iets past er in

**integer** (getal), **float** (kommagetal), **string** (woord), **boolean** (True/False)

**Waarde:** wat er nu echt in zit



naam: **score**

type: **int**

waarde: **180**





# Toewijzing

---

- = teken lees je als **wordt**
- Onthoud dat je score 180 is:

**score = 180**

- Lees: “score **wordt** 10”



# Type casting

---

Python kan niet **raden** wat de type van iets is. Soms moet je dat echt aangeven.

Als de **gebruiker iets intypt**, dan gaat Python uit van een **tekst**. Maar, met een tekst kun je niet rekenen, met getallen wel.

Je kunt een tekst **omzetten** naar een getal: **casting**

```
#vraag gebruiker hoeveel fietsen er zijn
aantal_fietsen = input ("Hoeveel fietsen zijn er?")

#bereken 2 wielen per fiets
aantal_wielen = 2 * int(aantal_fietsen)

#Druk aantal wielen af
print ("Dan zijn er ", aantal_wielen, " wielen.")
```



# Tekst en getallen combineren

- Er zijn twee manieren om `print` te gebruiken: `+` of `,`
- Met `,` krijg je er meteen een spatie bij
- Met `+` plak je twee teksten aan elkaar zonder spatie.

```
print("boeken", "kast")
```

Drukt af: **boeken kast**

```
print("boeken" + "kast")
```

Drukt af: **boekenkast**

- Wil je `+` gebruiken met getallen?

Dan eerst getal omzetten naar tekst met `str(...)`

```
print(str(4) + "euro")
```

Drukt af: **4euro**



# Check je kennis.

---

Wat wordt er op het scherm afgedrukt?

```
boodschap = "aardappel"  
fruit = "aardbei"  
print(boodschap)
```

aardappel

```
boodschap = "aardappel"  
fruit = "aardbei"  
print("boodschap")
```

boodschap