

1 Gegevens

Doorheen een computerprogrammaatje gaan we bijna altijd met gegevens zitten goochemen. Maar wat zijn gegevens nu juist? En kunnen we zomaar van het ene type naar het andere gaan?

1.1 Verschillende soorten

Laten we beginnen met een eerste kennismaking met de meest voorkomende soorten gegevens.

1.1.1 Text: `str`

De eerste soort gegevens die we tegenkomen zijn teksten. Een tekst is een reeks van karakters. Een karakter is een letter, cijfer, leesteken, spatie, We beginnen en eindigen tekst altijd met aanhalingstekens (").

```
"Dit is een tekst"
"47" # ook dat is tekst
```

1.1.2 Getallen: `int` en `float`

Een volgende soort gegevens zijn getallen. We maken een onderscheid tussen getallen zonder en met komma:

- Getallen zonder komma noemen we integers (het Engelse woord voor gehele getallen): `int`.
- Getallen met komma noemen we floats (het Engelse woord voor 'vlottende' komma): `float`.

```
47 # integer
47.0 # opgelet, een float!
3.14 # float
```



In python kan je van iedere waarde opvragen tot welke soort gegevens het hoort. Hiervoor gebruik je de functie `type()`. Weet jij het verschil tussen `"47"`, `47` en `47.0`? Probeer het eens in de interactive shell!

```
>>> type("47")
>>> type(47)
>>> type(47.0)
```

1.1.3 Logische waarde: `bool`

Een andere veelvoorkomende soort gegevens zijn logische waarden. Er bestaan exact twee logische waarden:

- `True`: de uitkomst is juist.
- `False`: de uitkomst is fout.

```
True
False
# dit zijn de enige twee logische waarden die bestaan
```

1.2 Transformaties

Het is heel belangrijk dat je je realiseert dat python de verschillende soorten gegevens anders behandelt. Ook al lijken ze voor jou op elkaar, zo python ziet een groot verschil tussen `"47"` en `47`. Gelukkig kunnen we meestal heel eenvoudig een waarde van de ene soort omzetten naar een waarde van een andere soort. Dit doen we met functies die dezelfde naam hebben als de verschillende soorten gegevens. Namelijk:

- `str()`: een waarde omzetten naar tekst
- `int()`: een waarde omzetten naar getal (zonder komma)
- `float()`: een waarde omzetten naar kommagetal
- `bool()`: een waarde omzetten naar logische waarde



We hebben deze twee waarden verkregen van de gebruiker via de functie `input()`: `"12"` en `"36"`.
We willen de gebruiker de som (+) van deze twee getallen tonen.
We kunnen dat rechtsreeks met de gekregen tekst doen. Of we kunnen de waarden eerst omvormen naar getallen.
Is er een verschil, denk je?
Had je het juist? Kan je uitleggen waarom?

```
>>> "12" + "36"
>>> int("12") + int("36")
```



Laten we nog eens iets anders proberen. Het getal `3.14` is een `float`, akkoord? Kunnen we dit omvormen naar een waarde van het type `int` denk je?
Zullen we dat eens samen proberen?

```
>>> int(3.14)
```

1.3 Kort overzicht

- `str`
Tekst, begint en eindigt met aanhalingstekens
vb. `"Dit is een tekst"`
- `int`
Gehele getallen (getallen zonder komma)
vb. `47`

- `float`
Getallen met komma
vb. `3.14`
- `bool`
Logische waarden
`True` of `False`
- Gegevens omvormen van het de ene soort in de andere:
`str()`, `int()`, `float()`, `bool()`