Python: For/While

D. Leeuw

10 juni 2025
1.0.0

1 Loops

De reden van automatisering is om herhaalde handelingen automatisch uit te laten voeren. Ook bij het schrijven van code willen we herhaalde handelingen minimaliseren. Als iets 5x uitgevoerd moet worden in en programma, dan willen we niet 5x dezelfde code hoeven te schrijven. We willen de code 1x schrijven en daarna 5x uitvoeren.

Hier komen in programmeertermen "loops" aan te pas. Een loop doorloopt een bepaald stuk code één of meerdere keren. In Python zijn er twee soorten loops. Er bestaat de for-loop en de while-loop. Daar gaan we het in dit stuk verder over hebben.

2 Wanneer een for- en wanneer een while-loop?

Zowel de for-loop als de while-loop zijn manieren om bepaalde handelingen in een programma te herhalen onder bepaalde voorwaarden. Waarom zijn er twee manieren om dat te doen?

2.1 De verschillen

Er zijn een aantal redenen waarom we in de ene situatie een for-loop gebruiken en in de andere een while-loop. Bij een for-loop geef je op welke waarden doorlopen moeten worden, bij de while-loop geef je een conditie op die bereikt moet worden. For loops doorlopen een reeks en while loops testen of iets waar (`True`) is.

Een while-loop zal eerder gebruikt worden in test omgevingen waar een bepaalde waarde bereikt moet worden. Bijvoorbeeld meten of de watertemperatuur 100°C bereikt heeft.

2.2 For kenmerken

- Als je van te voren weet hoe vaak je door een loop moet lopen, dan gebruik je de for-loop
- De for-loop volgt een sequentie (list, tuple, string, range) en voert het code block uit voor elk item in de sequentie
- De loop variable is beschikbaar in de loop
- Een for-loop zal doorgaan totdat alle elementen doorlopen zijn.
- Een for-loop stopt als het laatste item is bereikt.

2.3 While kenmerken

- De while-loop wordt gebruikt als je niet van te voren weet hoe vaak je een block code moet doorlopen, maar je weet aan welke conditie voldaan moet worden om de loop te laten eindigen.
- Bij een while-loop is het van belang dat de gegeven eind conditie ooit bereikt wordt, anders heb je een oneindige loop.
- Een while-loop zal doorgaan tot een bepaalde conditie is bereikt.
- Een while-loop stopt zodra de opgegeven conditie niet meer geldig is.

2.4 Samenvatting

Het belangrijkste verschil tussen een for- en while-loop is dat de for-loop kan worden gebruikt als het aantal iteraties bekend is en de while-loop kan worden gebruikt als het aantal iteraties niet bekend is.

	for-loop	while-loop
Loop variabele	Gedefinieerd in de loop aan het begin	Gedefinieerd buiten de loop, moet expliciet gedaan worden
Conditie	Controle voor elke iteratie	Controle voor elke iteratie
Update	Gedaan na elke iteratie	Gedaan in de loop, moet expliciet gedaan worden
Scope	Wordt bepaald door de loop body (voor gedefinieerd)	Moet expliciet gedaan worden
Gebruik	Als het aantal iteraties bekend is	Als het aantal iteraties niet bekend is, of als er aan een bepaalde conditie voldaan moet worden

3 Break: ontsnappen aan de loop

De for- en de while-loop kunnen voortijdig afgebroken worden door het break-statement. Een for-loop hoeft dan bijvoorbeeld niet alle items te doorlopen en een while-loop kan gestopt worden als bijvoorbeeld de loop een aantal keer doorlopen is.

Voorbeeld

```
for i in range(10):
    if i == 5:
        break
    print(i)
```

4 Continue: stappen overslaan

Het kan ook voorkomen dat je uit een reeks 1 bepaalde waarde niet mee wil nemen in je for-loop. Om door te gaan naar de volgende waarde in de reeks gebruiken we continue.

Voorbeeld

```
for i in range(10):
    if i == 5:
        continue
    print(i)
```