

Python: 5. Iteraties

Dr. Cornelis Stal

April 27, 2022

1 Iteraties

1.1 While-lus

Iteraties zijn programmeerconstructies die herhalingen mogelijk maken. De syntax van een **while**-statement is:

```
while <voorwaarde>:    <actie>
```

Net als bij een **if**-statement, test een **while**-statement een booleaanse expressie en wanneer deze expressie **True** oplevert, wordt het blok code in de **while**-lus uitgevoerd. Wanneer het blok code volledig uitgevoerd is, gaat Python terug naar de booleaanse expressie in de **while**-lus en wordt de test opnieuw uitgevoerd. Als de expressie nog steeds **True** oplevert, dan wordt het blok code nogmaals uitgevoerd. Nadat de actie opnieuw uitgevoerd is, keert Python wederom terug naar de booleaanse expressie. Deze procedure wordt steeds herhaald, totdat de booleaanse expressie **False** oplevert. Pas op dat moment slaat Python het blok code binnen de **while**-lus over en gaat eronder het blok verder.

Merk op dat als de booleaanse expressie meteen **False** oplevert, het blok code onder de **while** onmiddellijk overgeslagen wordt. Anderzijds zal het blok code oneindig vaak worden uitgevoerd als de booleaanse expressie op geen enkele manier zal resulteren in een **False**.

```
[ ]: num = 1
    while num <= 5:
        print(num)
        num += 1
    print("Klaar met de while-lus")
```

In bovenstaande code gebeurt het volgende: De eerste regel initialiseert een variabele **num** op 1. Vervolgens start de **while**-lus. De booleaanse expressie controleert dat **num <= 5**. Aangezien nu de variabele **num** nog gelijk is aan 1, en dus kleiner dan 5, levert de expressie **True**. Dus wordt het blok code onder de **while** uitgevoerd.

De eerste regel van het blok code print de waarde van **num** en zal dus een 1 afdrukken. De tweede regel telt 1 op bij de waarde van **num**, zodat **num** nu gelijk is aan 2. Python gaat daarna terug naar de booleaanse expressie. Omdat **num** nu gelijk is aan 2 is, geeft de booleaanse expressie nog een **True** terug. Het blok code wordt dus nogmaals uitgevoerd. 2 wordt afgedrukt, **num** krijgt een waarde 3, en de code keert terug bij de booleaanse expressie, ...

De laatste regel van het programma, het printen van de tekst **Klaar**, wordt dus pas uitgevoerd de code dusdanig vaak is uitgevoerd dat de waarde van **num** gelijk is aan 6.

1.2 For-lus

For-lussen worden voor andere doeleinden gebruikt dan **while**-lussen. Bij een **for**-lus ligt namelijk op voorhand vast hoe vaak men door de lus gaat. De syntax voor de **for**-lus is:

```
for <variabele> in <collectie>:    actie
```

Om hetzelfde resultaat als in de zojuist gegeven **while**-lus te verkrijgen, gebruiken we de volgende code. Hierbij wordt de ondergrens meegenomen bij de evaluatie, maar de bovengrens niet.

```
[ ]: for num in range(1, 6):
      print(num)
      print("Klaar met de for-lus")
```

Een **for**-lus wordt ook vaak gebruikt om een collectie van items één voor één te verwerken in de volgorde waarin ze worden aangeboden (zie verder).

```
[ ]: toestel = 'Topcon ES-105'
      for letter in toestel:
          print(letter)
      print("Klaar met de for-lus")
```

1.3 Break-statement

Het **break**-statement maakt het mogelijk een lus voortijdig af te breken. Als Python een **break** tegenkomt, stopt het met het verwerken van de lus en keert niet terug naar de eerste regel van de loop. In plaats daarvan gaat de verwerking verder met de eerste regel code na de lus.

In onderstaand voorbeeld presenteren we een verwerkingsprocedure dat stopt wanneer een gemeten afstand kleiner is dan 50 cm:

```
[ ]: resultaat = "succesvol"
      for meting in (12.5, 3.6, 1.5, 6.9, 2.0, 1.8, 4.6, 0.3, 6.4, 4.4, 5.0):
          if meting < 0.5:
              resultaat = "niet succesvol"
              break
      print("De metingen zijn %s" % resultaat)
```

1.4 Continue-statement

Als het “continue”-statement in een blok code bij een loop wordt aangetroffen, wordt onmiddellijk het uitvoeren van de huidige cyclus in de lus beëindigd en wordt teruggekeerd naar de eerste regel van de lus:

- **while**-lus: booleaanse expressie wordt opnieuw geëvalueerd;
- **for**-lus: het volgende item van de collectie wordt genomen en verwerkt.

In het volgende voorbeeld worden oneven getallen kleiner dan 10 uitgeprint:

```
[ ]: i = 0
      while i < 10:
```

```
if i % 2 == 0:
    i += 1
    continue
print(i)
i += 1
```

1.5 Oefeningen

Opdracht: stemgerechtigden

Men schat dat het aantal stemgerechtigden in een bepaalde stad de volgende jaren als volgt zal verlopen:

$$N(t) = 30 + 12t^2 - t^3$$

met $0 \leq t \leq 8$ waarbij t de tijd in jaren is en $N(t)$ het aantal stemgerechtigden (in duizenden) na t jaar.

Bepaal voor elk jaar het aantal stemgerechtigden.

[]: `## UW CODE HIER ##`

Opdracht: Opdracht: vat

Een cilindervormig vat met hoogte van 32 dm heeft een inhoud van 8000 liter en is geheel gevuld met water. Als men de kraan opendraait stroomt het vat leeg. Tijdens het leegstromen geldt voor de hoogte h van de waterspiegel op tijdstip t bij benadering de formule:

$$h(t) = 0.0008 * t^2 - 0.32 * t + 32$$

Hierin is t de tijd in minuten vanaf het moment waarop de kraan opengedraaid wordt en h de hoogte van de waterspiegel in decimeter.

Schrijf een programma waarmee de volgende vragen beantwoord kunnen worden:

- Wat is de hoogte na 2, 4, ... 10 minuten?
- Na hoeveel minuten is de hoogte gezakt onder 5 dm?

[]: `## UW CODE HIER ##`

Opdracht: vuurpijl

Van een hoge brug wordt op tijdstip $t = 0$ een vuurpijl verticaal omhoog geschoten. De hoogte van de vuurpijl (uitgedrukt in meter) wordt beschreven door de functie met het voorschrift:

$$h(t) = -5t^2 + 30t + 80$$

waarbij t de tijd is (uitgedrukt in seconden). Na hoeveel seconden landt de vuurpijl op de grond?

[]: `## UW CODE HIER ##`

Opdracht: fietscomputers

Een bedrijf produceert draadloze fietscomputers. De bedrijfsleiding wil weten hoeveel computers er per uur moeten geproduceerd worden om winst te maken. Het verband tussen de winst W (in euro) en de productie x (aantal geproduceerde computers) wordt gegeven door:

$$W(x) = 0.5x^3 + 8x^2 - 24x$$

Schrijf een programma om dit vraagstuk op te lossen.

[]: `## UW CODE HIER ##`

Opdracht: firma

Een firma verkoopt maandelijks 5000 stuks van een bepaald type. De vraagprijs per stuk bedraagt 30 euro. Marktonderzoek heeft uitgewezen dat de verkoop met 500 stuks zal stijgen telkens de prijs met 2 euro wordt verlaagd. Welke prijs moet de firma aanrekenen wil men een maximale omzet realiseren?

[]: `## UW CODE HIER ##`

Opdracht: vermoeden van Collatz

We beschouwen de rij met onderstaand voorschrift:

$a_{i+1} = \frac{a_i}{2}$ indien a_i is even

$a_{i+1} = 3a_i + 1$ indien a_i is oneven

waarbij we de rij starten met $a_0 = n$. Het vermoeden van Collatz bestaat erin dat deze rij, ongeacht $n > 0$ vroeg of laat op 1 uitkomt.

Schrijf een programma dat voor een ingelezen $n > 0$ bepaalt na hoeveel stappen de rij op het getal 1 botst.

[]: `## UW CODE HIER ##`