Opdrachten Python 102

# While-statement

Zoals je gezien hebt lijkt een while-statement op een if-statement. Het verschil is dat bij een while-statement het stuk code dat ingesprongen staat herhaald wordt zolang wat er achter het woord while staat waar is. Het is dan ook handig om te zorgen dat wat je hier controleert verandert, want als je elke keer precies hetzelfde controleert blijft je computer eeuwig doorgaan met de lus!

**Opdracht:** Schrijf een stukje Python-code waarin je je leeftijd in een variabele leeftijd zet, en dan met een while-lus alle getallen van 1 tot en met je leeftijd bij elkaar optelt, en daarna uitprint. Je kan hierbij een extra variabele gebruiken die je in het begin op 0 zet.

We hebben gezien dat je met het break-statement een lus eerder kan stoppen.

**Opdracht:** Schrijf een stukje Python-code waarin je in een while-lus de getallen van 1 tot en met je leeftijd bij elkaar optelt. Laat de while-lus stoppen bij het getal 10; dat betekent dus dat je eigenlijk alleen de getallen van 1 tot en met 10 bij elkaar optelt, maar dan door een break-statement te gebruiken.

# If-statement

In het vorige keuzecollege hebben we het if-statement gezien. We hebben daarbij gezien dat je met een if-statement een keuze kan maken door bijvoorbeeld twee getallen te vergelijken. Vandaag gaan we hier nog wat op verder.

**Opdracht:** Probeer een stukje Python-code te schrijven dat je leeftijd in een variabele leeftijd zet. Print dan je leeftijd alleen als deze groter dan 10 en kleiner dan 17 is.

Het lijkt lastig om twee dingen te vergelijken, zoals hierboven. Daar heeft Python echter een oplossing voor: je kan twee vergelijkingen allebei gebruiken door er het woord and tussen te zetten.

**if** leeftijd > 10 **and** leeftijd < 17:

**print**(leeftijd)

Je kan ook testen of minstens één van twee vergelijkingen waar is door er het woord or tussen te zetten.

**if** leeftijd < 11 **or** leeftijd > 16:

**print**(leeftijd)

Bovenstaande code print de leeftijd alleen als deze kleiner dan 11 of groter dan 16 is. Merk op dat dit nooit tegelijk waar kan zijn!

**Opdracht:** Schrijf een stukje Python-code dat je leeftijd print, maar alleen als deze deelbaar is door 2 of deelbaar door 3. Vorige keer hebben we gezien dat we “x is deelbaar door y” in Python geschreven kan worden als x % y == 0.

# Lijsten

We hebben gezien dat we getallen kunnen toevoegen aan en verwijderen uit lijsten.

**Opdracht:** Bekijk de code die hieronder staat. Wat denk je dat er geprint zal worden? Voer de code nu uit. Klopt dit met wat je dacht?

list = [2, 4, 6, 8, 10]

**del** list[1]

**print**(list[2])

list.append(12)

**print**(list[4])

**Opdracht:** Maak een lijst met de getallen van 0 tot en met 9. Haal hieruit het getal 5 weg, en gebruik een for-lus om het kwadraat van de getallen te printen.

In de vorige opdrachtenset stond deze opdracht als bonusopdracht, maar we hebben nu genoeg extra Pythonkennis om deze opdracht te kunnen maken. FizzBuzz is een heel bekende opdracht in de ICT, die wel gebruikt wordt bij sollicitatiegesprekken om te kijken of iemand een beetje kan programmeren.

**Opdracht:** Schrijf een stuk Python-code die alle getallen van 1 tot 100 onder elkaar zet. Echter, als het getal deelbaar is door 3, moet je niet het getal printen, maar het woord Fizz. Als het getal deelbaar is door 5, moet je in plaats van het getal het woord Buzz printen. Als het getal deelbaar is door 5 en door 3, dan moet je in plaats van het getal het woord FizzBuzz printen.

# Recursie

De *faculteit* van een getal is het product van 1 tot en met het getal zelf. Zo is bijvoorbeeld de faculteit van 5 (dat schrijf je ook wel als 5!) gelijk aan 1 × 2 × 3 × 4 × 5 = 120.

Je kan hieraan ook zien dat de faculteit van 5 gelijk is aan 4! × 5, immers 4! = 1 × 2 × 3 × 4. Dit is een voorbeeld van een recursieve functie: je kan de waarde berekenen aan de hand van de vorige waarde. We moeten echter ook een begingeval kiezen, anders gaan we oneindig door. We kunnen bijvoorbeeld zien dat 1! = 1, dus als de invoer van de functie 1 is kunnen we meteen als uitvoer 1 geven.

**Opdracht:** Maak een functie faculteit in Python die de faculteit van een getal berekent.