

Python: 6. Functies

Dr. Cornelis Stal

April 27, 2022

1 Functies

1.1 Ingebouwde functies

Een functie is een blok herbruikbare code dat een bepaalde actie uitvoert. Om een functie aan het werk te zetten, roepen we hem aan met de parameters die de functie nodig heeft. We hoeven niet te weten hoe de functie precies werkt. Er dienen minimaal drie zaken geweten te zijn:

- De naam van de functie;
- De parameters die de functie nodig heeft (als die er zijn);
- De waarde die de functie teruggeeft (als er zo'n waarde is).

In onderstaande voorbeeld worden 3 variabelen gedefinieerd, waarop we enkele ingebouwde functies of methodes uit zullen voeren.

```
[ ]: x = -2
      y = 3
      z = 1.27
      print(max(x, y, z))
      print(min(x, y, z))
      print(pow(x, y))
      print(round(z, 1))
      print(len("Dit is een tekst met 33 karakters"))
```

In veel programma's willen we dat de gebruiker van het programma data verstrekkt. We kunnen de gebruiker vragen een string in te typen met behulp van de `input`-functie:

```
[ ]: tekst = input("Type de tekst in: ")
      print("De opgegeven tekst is: ", tekst)
```

Deze functie zal altijd een string als resultaat geven. Indien we echter een getal wensen als input, zullen we deze string eerst moeten converteren:

```
[ ]: getal = input("Type een getal in: ")
      print("Het kwadraat van het opgegeven getal is " + str(pow(float(getal),2)))
```

1.2 Eigen functies

Het is vaak zeer nuttig om na te denken over het schrijven van eigen functies. Eigen functies hebben namelijk een aantal interessante voordelen:

- Een bepaalde functionaliteit binnen de code kan onafhankelijk van de rest van het programma ontwikkeld worden. Als we een dergelijke functionaliteit in een functie stoppen, kunnen we deze functie vaak onafhankelijk van de rest van de code testen en verder verfijnen. Uiteindelijk hoeven we na afronden van de code niet verder na te denken over het gebruik ervan. Uiteraard is goede documentatie van de code dan wel vereist;
- Bepaalde functionaliteiten kunnen we door middel van eigen functies op verschillende plaatsen in het programma oproepen. Hierdoor hoeft de uit te voeren code niet telkens gekopieerd te worden, wat de code beduidend korter en overzichtelijker zal maken;
- Het laatste element in het vorige puntje, zijnde het korter en overzichtelijker maken van de code, resulteert in betere controle over de inhoud en de werking van de code zelf.

Het is dus sterk aan te raden om code op te delen in meerdere eigen functies. We kunnen eigen functies maken zonder variabelen:

```
[ ]: # Een eigen functie definiëren
def welcome():
    print("Welkom bij het OLOD 'Project GeoICT'")

# Roep de functie op
welcome()
```

En we kunnen functies maken met variabelen:

```
[ ]: # Een functie met parameters definiëren
def convertDeg2Gon(deg):
    gon = deg / 360 * 400
    return gon

# Roep de functie op
degAngle = 10.235
gonAngle = convertDeg2Gon(10.235)
print("%s graden is %s gon" % (str(degAngle), str(gonAngle)))
```

1.3 Oefeningen

Opdracht: wandelen en calorieenverbranden

Iemand die wandelt aan 5km/uur verbruikt het volgende aantal calorieën per kilometer (Bron: [calorieenverbranden](#)):

- tot 55 kg: 44 kcal;
- tot 65 kg: 51 kcal;
- tot 75 kg: 59 kcal;
- tot 85 kg: 67 kcal;
- tot 95 kg: 75 kcal;
- anders: 104 kcal;

Een persoon met een gewicht van 85 kg zal dus bij een wandeling van 25 km een totaal van 1875 kcal verbranden.

Schrijf een functie die op basis van het gewicht en het aantal kilometer het aantal verbruikte kcal geeft.

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

Opdracht: levensverwachting

Veronderstel dat de levensverwachting van een doorsnee persoon 70 jaar is, en dat die levensverwachting kan worden aangepast op basis van de volgende criteria:

- Bij vrouwen wordt er vier jaar bijgeteld;
- Bij rokers wordt er vijf jaar afgetrokken en bij niet-rokers wordt er vijf jaar bijgeteld;
- Trek drie jaar af voor personen die nooit sporten en tel één jaar bij voor elk uur dat iemand wekelijks aan sport doet;
- Trek een half jaar af voor elk geconsumeerd glas alcohol boven de zeven dat iemand per week drinkt (bij iemand die elf glazen alcohol per week drinkt worden dus twee jaren afgetrokken); bij geheelonthouders wordt er twee jaar bijgeteld;
- Bij personen die niet vaak fast food eten wordt er drie jaar bijgeteld.

Op basis van deze formule kan de levensverwachting van een mannelijke roker die slechts twee uur per week aan sport doet, tien glazen alcohol per week drinkt en vaak fast food eet worden berekend op 65,5 jaar. Commentaar: Dit is GEEN wetenschappelijke voorspelling.

Schrijf een functie levensverwachting waaraan de volgende vijf parameters moeten doorgegeven worden:

- **geslacht**: een string die het geslacht van de persoon aangeeft (man of vrouw);
- **roker**: een Booleaanse waarde die aangeeft of de persoon rookt;
- **sport**: een natuurlijk getal dat aangeeft hoeveel uren per week de persoon aan sport doet;
- **alcohol**: een natuurlijk getal dat aangeeft hoeveel glazen alcohol de persoon per week drinkt;
- **fastfood**: een Booleaanse waarde die aangeeft of de persoon vaak fastfood eet.

De functie moet op basis van de doorgegeven waarden de levensverwachting voorspellen, en moet deze prognose als resultaat teruggeven. Het resultaat van de functie moet als een reëel getal teruggegeven worden. Bijvoorbeeld:

```
print(levensverwachting("man", True, 2, 10, True)) geeft 65.5  
print(levensverwachting("man", True, 5, 5, True)) geeft 70.0  
print(levensverwachting("vrouw", False, 5, 0, False)) geeft 89.0  
print(levensverwachting("vrouw", False, 3, 14, True)) geeft 78.5  
print(levensverwachting("man", False, 4, 4, False)) geeft 82.0
```

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