Lesbrief 6 Strings knippen en functies

Slicing

Buiten string concatenation, hebben we ook slicing. Dit is wanneer je een string hebt en hier vervolgens in gaat knippen. Zo heb je bijvoorbeeld het woord:

"supercalifragilisticexpialidocious"

Als je hier uit dit woord bepaalde delen wilt halen, kan je hier dus slicing voor gebruiken. Zo kan je uit deze string het woord "super" halen! Dit kan bijvoorbeeld met indexing en gaat als volgt:

```
x = "supercalifragilisticexpialidocious"
print (x[0:5])
```

Dit zal dan “super” printen! Het eerste getal staat voor waar je begint in de string, en het tweede waar het eindigt (hij print dus alle karakters op de posities 0, 1, 2, 3 en 4 (en niet op positie 5). Bij programmeren begin je eigenlijk altijd met tellen bij de 0, dus het nulde element zal de "s" zijn. Als je niks invult voor of achter de dubbele punt wordt er gewoon tot het einde van de string aan die kant genomen. Dus:

```
print (x[9:])
```

Dit zal dan “fragilisticexpialidocious” printen! Probeer het zelf maar, en speel er een beetje mee:

String slicing voorbeeld

Opdracht 1
Neem het woord:

"Zandzeepsodemineraalwatersteenstralen"

haal de volgende woorden eruit:

zeep
mineraal
steen
stralen

en concatenate vervolgens deze woorden aan elkaar:

zeepstralen
Zandsteen

Opdracht 2
Haal elke pokemon uit de volgende string (elke naam begint met een hoofdletter) en print deze uit:

“BulbasaurIvysaurCharmeleonWartortlePikachuJigglypuffKadabraSlowpoke”

Opdracht 3
Vraag steeds aan de gebruiker om een woord, en slice hier de laatste twee letters van af. Als de gebruiker “stop” geeft, dan stopt het programma.

Opdracht 4
Vraag vijf keer om een zin, en print de eerste en laatste twee letters van deze zin.

***** Opdracht 5 (ster opdrachten sowieso doen!)**
Print nu de volgende string achterstevoren:

".naag et prewredno edneglov teh raan mo raalk ej neb uN !deog leeH"

Functies

Functies heb je inmiddels al een paar keer gezien, maar nooit echt zelf gemaakt! Een functie is een stuk code die je kan aanroepen en zo steeds dezelfde acties kan uitvoeren door de naam ervan aan te roepen. Je hebt al heel vaak inmiddels functies als `print()`, `len()`, `input()` etc. Maar nu gaan we kijken hoe we dit soort dingen zelf kunnen maken! Het gaat als volgt, stel je wilt van drie verschillende getallen steeds weten welke de grootste is, dan kan je daar een functie voor schrijven: Hoogste van de drie functie

```
def max_van_drie ():
  a = int(input("Geef me een getal!"))
  b = int(input("Geef me nog een getal!"))
  c = int(input("Geef me nog een laatste getal!"))
```

```
  if a < b:
    if b < c:
      print (str(c) + " is het grootste getal!")
    else:
      print (str(b) + " is het grootste getal!")
  else:
    print (str(a) + " is het grootste getal!")
```

Hier stopt de functie. Hieronder kan je dan vervolgens de functie aanroepen en dan wordt het bovenstaande uitgevoerd, zonder het aanroepen van de functie wordt deze dus ook niet uitgevoerd.

```
max_van_drie()
```

Hier zie je dat alles wat onder de `def` staat en geïndenteerd is (een tab ervoor heeft) bij de functie hoort. Zo kan je voor vrijwel alles een functie schrijven, die je later simpelweg alleen maar hoeft aan te roepen om hem meerdere keren te kunnen gebruiken! Handuuuuuuuugh!

Verder kan je ook nog een variabele aan een functie mee geven, zo kunnen we bijvoorbeeld de max_van_drie functie ook omschrijven zodat er geen input van de gebruiker gevraagd wordt. Dit werkt als volgt:

```
def max_van_drie (a, b, c):
  if a < b:
    if b < c:
      print (str(c) + " is het grootste getal!")
    else:
      print (str(b) + " is het grootste getal!")
  else:
    print (str(a) + " is het grootste getal!")
```

Als we de functie nu aanroepen, kunnen we de waarden van a, b en c meer geven tussen de haakjes. Zoals dit:

max_van_drie(4,9,23)

Hier wordt de 4 als het eerste argument gezien dat meegegeven wordt aan de functie, en zal dus als “a” in de functie zelf gebruikt kunnen worden. Kijk zelf maar!

Als je nog een beetje meer informatie wilt over zelf functies maken, kan je ook het komende filmpje bekijken: Uitleg functies maken

Opdracht 6
Schrijf een functie die een zin vraagt aan de gebruiker en deze vervolgens achterstevoren print. Dit kan met de slicing techniek!

Opdracht 7
Schrijf een functie die steeds om een aantal getallen vraagt, en deze allemaal bij elkaar optelt en de som ervan uiteindelijk uit print.

Opdracht 8
Schrijf een functie die checkt of de input van een gebruiker binnen een bepaalde range valt. Dus of deze tussen twee verschillende getallen past.

Voorbeeld:

```
>>> Hallo, wilt u mij een getal geven?

>>> 38

>>> 38 valt niet tussen de 3 en de 9
```

Opdracht 9
Schrijf een functie die drie verschillende strings vraagt aan de gebruiker, de lengte ervan berekend, deze lengte vervolgens bij elkaar op telt, dit getal dan deelt door 2.5, en het dat uiteindelijke getal afgerond naar boven en naar beneden print.

Opdracht 10
Schrijf een functie die een lange string vraagt aan de gebruiker en vervolgens deze string in drie gelijke delen slicet (als de lengte van de string niet door drie te delen valt, rond dit dan af). En in een andere volgorde print.

Voorbeeld:

```
>>> Hallo, mag ik een zin?

>>> Dit is een voorbeeldzin

>>> rbeeldzineen vooDit is
```

Opdracht 11
Schrijf een functie die de laagste van vijf cijfers terug print.

***** Opdracht 12**
Schrijf een functie die drie variabelen mee krijgt, en vervolgens deze allemaal met elkaar vermenigvuldigd en het totaal print.