Lesbrief 2: Input en datatypes

Terug naar strings

Nu we al die lastige informatie hebben verwerkt gaan we even weer verder met berekeningen met variabelen en deze straks combineren met strings!

We kunnen de variabelen bijvoorbeeld ook weer gebruiken om de turtle, die je hebt leren kennen in Python 0, vooruit te laten bewegen. In plaats van dat je een getal meegeeft aan de `forward()` functie, kun je bijvoorbeeld ook een variabelenaam meegeven. En ook een berekening met een variabele! Dat ziet er dan zo uit:

```
turtle.forward(d*10)
```

Het bovenstaande kun je ook via de volgende link in actie zien: Turtle variabele

Opgave 1

Wat wordt met de onderstaande code op het scherm geprint? Bedenk het eerst voor jezelf en probeer het daarna uit.

```
a = 10
b = 15
c = a + b
d = a - b
a = d - c
a = a - 3
print (a)
```

Opgave 2

Schrijf een programma waarmee je twee variabelen instelt met een beginwaarde. De eerste met waarde 3 en de tweede variabele met waarde 8. De namen van de variabelen mag je zelf weten. Tel de twee getallen bij elkaar op en sla het resultaat op in een nieuwe variabele. Print uiteindelijk het resultaat naar het scherm.

Interactieve programma's met de input functie

Met de print functie kun je gegevens op het scherm printen. Maar wat als je nou wilt dat de gebruiker iets kan invoeren? Heb je daar ook iets voor? Zie bijvoorbeeld het volgende programma. Het tekent een cirkel op basis van het antwoord van de gebruiker: Cirkel met invoer
Als je het programma uitvoert kun je rechtsonder je kleur invoeren zodat het programma het kan gebruiken.

Bij het voorgaande voorbeeld zie je dat er een nieuwe functie wordt gebruikt om informatie van de gebruiker te vragen. Met deze invoer van informatie (ook wel input genoemd) kunnen vervolgens weer allerlei dingen gedaan worden. Maar je moet het resultaat van de `input` functie wel eerst in een variabele zetten, zodat dit later gebruikt kan worden. Op deze manier kan je dus een programma schrijven dat reageert op de antwoorden van zijn gebruiker!

Opgave 3

Schrijf een programma dat de voornaam van de gebruiker opvraagt met de functie input. Vervolgens wordt de gebruiker gegroet met 'Hallo, [en hier komt de voornaam van de gebruiker]!'. Hint: denk aan concatenation!

Opgave 4

Schrijf een programma dat de gebruiker vraagt om een zelfstandig naamwoord (bv. boom, huis, kast), een bijvoeglijk naamwoord (bv. groot, vies, vervelend) en een infinitief van een werkwoord (bv. lopen, schrijven, eten). Vul de ingevoerde vervolgens in, in de onderstaande zin en zet het resultaat op het beeld.

Vond je ooit een <*zelfstandig naamwoord*> zo ontzettend <*bijvoeglijk naamwoord*> dat je het voortdurend opnieuw wilde <*werkwoord*>?

Bijvoorbeeld: Vond je ooit een boom zo ontzettend mooi dat je het voortdurend opnieuw wilde knuffelen?

Opgave 5

Schrijf een programma dat om drie woorden vraagt en vervolgens die drie woorden in alle zes mogelijke volgordes weer naar het scherm schrijft. Je moet zes keer een print commando geven! Bekijk het volgende voorbeeld: hier.

Van strings naar getallen

Bekijk eens het onderstaande programma:

```
getal1 = input("Geef een getal")
getal2 = input("Geef nog een getal")
som = getal1 + getal2
print ("De som van getal1 en getal2 is")
print (som)
```

Je zou denken dat je hiermee een optelprogramma hebt gemaakt, maar dat valt vies tegen! Probeer maar eens een paar getallen op te tellen en kijk goed naar het resultaat. Je kunt het programma hier testen:

Van string naar int

Wat gaat hier nou mis? Als je voor het eerste getal 22 kiest en voor het tweede getal 56, dan is het resultaat 2256. Wat is hier misgegaan?

Python heeft de twee 'getallen' aan elkaar vastgeplakt. Dit doet hij, omdat hij denkt dat je twee strings aan elkaar wilt vastplakken (concatenation!). Net alsof je `"programmeren is "` en `"fun"` aan elkaar wilt vastplakken. Maar wij willen niet dat Python die twee getallen aan elkaar vastplakt, maar dat hij ze bij elkaar optelt! Eigenlijk willen we dat Python de twee getallen ook *echt* als twee getallen gaat beschouwen. De vraag is: Hoe doen we dat?

De input-functie is hier van belang. Die zegt altijd dat de invoer van de gebruiker als tekst moet worden beschouwd. Wij kunnen expliciet aangeven dat de invoer *toch* moet worden opgevat als een echt getal. Dit doen we door de functie int() te gebruiken. Dit ziet er als volgt uit:

```
getal1 = int(input("Geef een getal"))
getal2 = int(input("Geef nog een getal"))
som = getal1 + getal2
print ("De som van getal1 en getal2 is")
print (som)
```

Let vooral op het gebruik van de functie `int()`! `int` is een afkorting voor integer. Het is de Engelstalige benaming voor een **geheel getal**. Je zegt hier eigenlijk: Hetgeen dat de gebruiker invoert met de input functie moet worden opgevat als een getal.

Het juiste optelprogramma kan via de volgende link worden getest: Werkende omzetting van string naar int

Opgave 6

Schrijf een programma dat vraagt om vier getallen. Tel de eerste twee getallen bij elkaar op, deel dat dan door het derde getal, en vermenigvuldig het met het vierde getal. Print vervolgens het antwoord op het scherm.

Opgave 7

Schrijf een programma dat de gebruiker vraagt om vijf getallen en schrijf daarna het gemiddelde van die vijf getallen naar het scherm (niet afronden).

Opgave 8

Schrijf een programma dat de gebruiker vraagt om het huidige jaartal en om het jaar waarin hij geboren is. Reken uit hoe oud de gebruiker is aan het eind van het huidige jaar (in hele jaren) en schrijf het antwoord naar het scherm.

Types en type-casting

Inmiddels ben je al bekend met twee soorten informatietypes (datatypes), namelijk `strings` en `int's`. `Strings` zijn een verzameling karakters die als het ware als een zin gebruikt worden door de computer. Een `string` is een type (soort) informatie waarvan de computer alles tussen de "" als een geheel behandeld. En net zoals dat een `string` een type informatie is, zijn er ook nog andere types. Een van de belangrijkste hiervan ken je nu ook al. De `int` wat dus staat voor integer, was een heel getal. Maar wat nou als je met decimale getallen wilt werken (getallen met een komma erin)? Hiervoor bestaat er nog een ander type informatie. Namelijk de `float` wat een getal is met nog cijfers achter de komma.

In het Engels gebruik je een '.' (punt) in plaats van een ',' (komma). Daarom schrijf je het getal 2,63 bij programmeren als 2.63.

Deze laatste twee datatypes zijn heel erg belangrijk aangezien je er complexe berekeningen mee kan doen. Eerder deze les heb je al meerdere berekeningen gedaan met gehele getallen. Als je daarentegen met decimale getallen wilt werken heb je dus de `floats` nodig. Een paar voorbeelden van `floats` zijn: Voorbeelden van floats!

Hierboven zie je dat er verscheidene berekeningen gedaan worden met `floats`. Als je met de `input()` functie werkt, wordt het belangrijk om hetzelfde trucje toe te kunnen passen als we eerder met de `ints` hebben gedaan. Als je namelijk niet expliciet het type informatie geeft aan de `input()` functie, gaat dit altijd als string behandeld worden. Maar als je het type `int` aan een `float` mee geeft krijg je hier een error van. Geef bij het volgende voorbeeld maar een `float` als input mee en kijk wat er gebeurt!

Voorbeeld int/float error

Om dit op te lossen kan je dus simpelweg in plaats van de functie `int()` de `float()` functie gebruiken. Als je dit doet, wordt de input behandeld als `float` en kan je hier dus weer berekeningen mee maken met getallen achter de komma.

Oplossing int/float error

Het bovenstaande voorbeeld laat dus heel goed het verschil tussen `float` en `int` waardes zien. Bij berekeningen moet je hier dus goed op letten hoe je het in gaat voeren en wat voor antwoord je wilt krijgen.
Verder kan het natuurlijk ook zo zijn dat je wilt dat een `int` of `float` als `string` behandeld wordt. Dit is namelijk nodig als je een variabele die een `int` of `float` bevat wilt printen. Een voorbeeld hiervan is het volgende:

Zo print je variabele waardes!

Opgave 9
Geef de types van de outputs van de volgende tien printjes.

```
print (19)
print (“3”)
print (21.234)
print (“Hoela hoepen”)
print (24 * 21.4)
print (29 / 3)
print (“293.3”)
print (25.1 + 5)
print (2 / 3 * 9)
```

```
print (8 + 3 / 10.4)
```

Zoals je hierboven kan zien, maakt het erg veel uit hoe je de berekening invoert. Voor de volgende opdracht ga je hier zelf een programma voor schrijven:

Opdracht 10

Schrijf een programma dat 2 of meer inputs vraagt, maak er een berekening mee en print het antwoord uit. In het programma moeten alle drie de datatypes voorkomen. Een voorbeeld hiervan is het volgende:

Voorbeeldprogramma: Types

Als je even niet meer weet wat voor type iets heeft, bestaat er een functie die je hiervoor kan gebruiken. Als je namelijk `type(variabele)` gebruikt, zegt de computer wat het type is. Een voorbeeld hiervan is het volgende: Types checken