

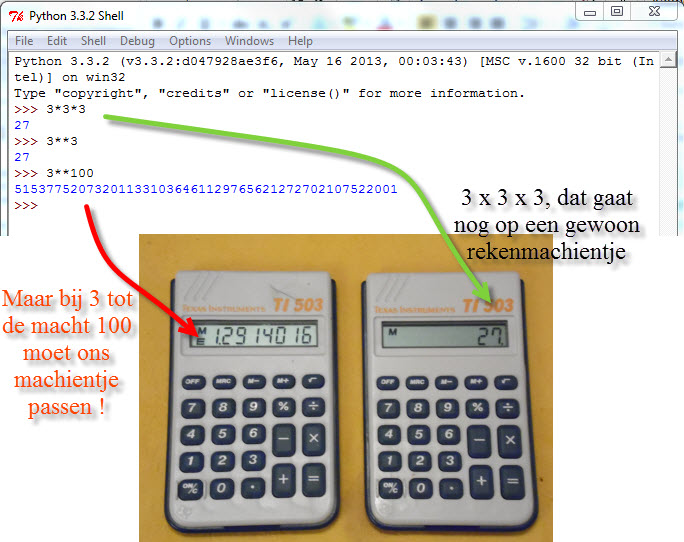


Bert Smits voor

Basisschool “De Esdoorn”  
Bankstraat 29  
2811 Hombeek (Mechelen)

Programmeren in Python – Deel 2

1. We hebben in de vorige sessie geleerd dat we Python konden gebruiken als een knap rekenmachientje:



Er is nog een verschil : in een programmeertaal gebruiken we haakjes om controle te houden over te volgorde van de bewerkingen. Als we geen haakjes plaatsen, dan hebben delen en vermenigvuldigen voorrang op optellen en aftrekken :

7 + 30 \* 20 in Python betekent hetzelfde als 7 + (30\*20). Dus eerst wordt 30\*20 uitgevoerd, en dan wordt er 7 bij opgeteld : het resultaat is dus 607



Als we eerst 7 bij 30 hadden willen optellen, en het eindresultaat daarna met 20 vermenigvuldigen, dan hadden we moeten schrijven:

(7 + 30) \* 20.

Dan voert hij eerst de berekening tussen de haakjes uit, en vermenigvuldigt het resultaat dan met 20.



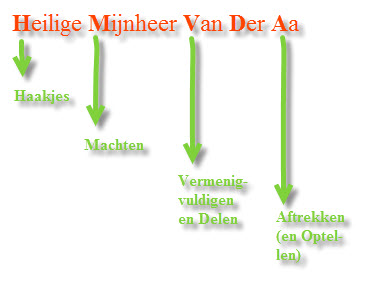
Tenslotte heeft Machtsverheffen voorrang op vermenigvuldigen en delen.

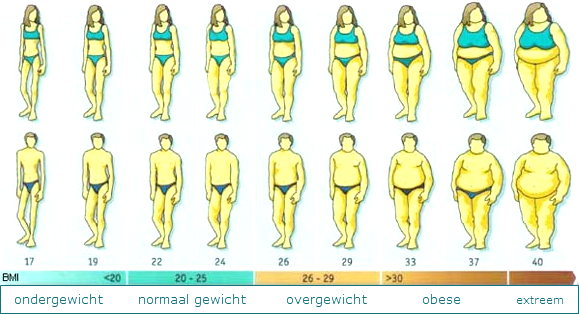
Dus als je 23x4 hebt, hetgeen we in Python noteren als 2\*\*3\*4, dan gaat hij eerst 23 (= 8) uitrekenen, en het resultaat met 4 vermenigvuldigen.



Als je wil dat er eerst 3 x 4 uitgerekend wordt, dan moet je haakjes zetten (en dan hebben we het over 2(3x4) = 212).

Je kan de volgorde van berekeningen onthouden met een “ezelsbruggetje” : **H**eilige **M**ijnheer **V**an **D**er **A**a



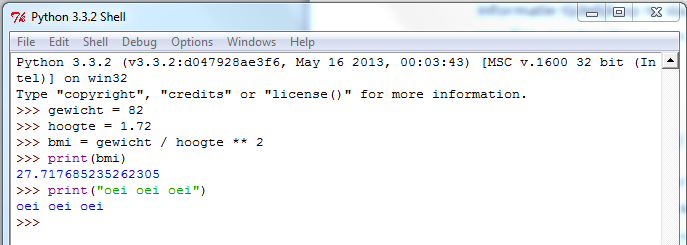
1. Variabelen – Naamkaartjes. Bij het programmeren gebruiken we een variabele als een plaats om informatie tijdelijk op te slaan. Dat kan tekst zijn, maar ook een getal, een lijst,   
   een berekening en noem maar op.

gewicht = 82

hoogte = 1.73

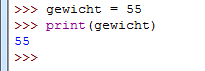
bmi = gewicht /hoogte\*\*2

Dit is de formule voor de “body mass index”. Ze zeggen dan mensen met een BMI van meer dan 25 “zwaarlijvig” zijn of “overgewicht” hebben.

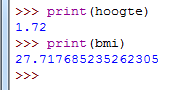


Nu kunnen we aan Python zeggen om de waarde van een variabele te veranderen :

Gewicht = 55. Als we nadien de variabele “gewicht” opnieuw afdrukken, dan zien we dat deze nu de waarde 55 heeft (was voorheen 82)



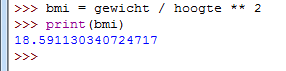
Aan de hoogte en de bmi is niets veranderd.



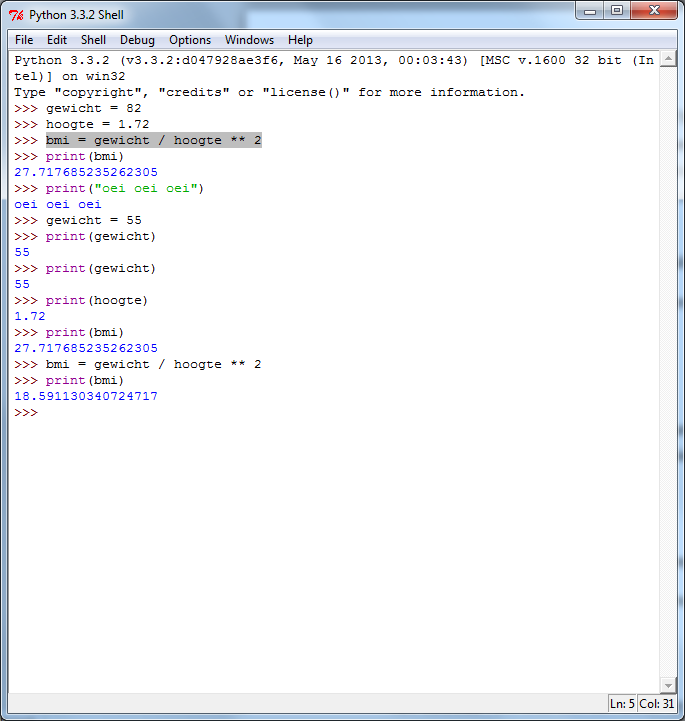
Tiens, maar was de bmi niet **gewicht / hoogte2** ? Ja, maar zoals je weet zijn computers…DOM. De computer heeft **bmi** een waarde toegekend toen gewicht nog 82 en hoogte 1.72 was. Die werd toen 27.717685… en daar is sindsdien niets aan veranderd, ook al is gewicht nu in 55 veranderd.

Wat je wel kan doen is nog eens **bmi** als een formule uitrekenen :

bmi = gewicht /hoogte\*\*2 en dan krijgen we



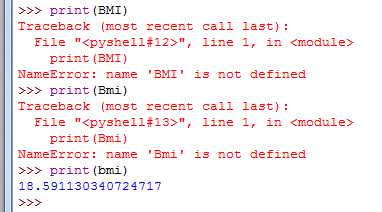
Nu is de waarde voor bmi natuurlijk aangepast aan de nieuwe waarden. Overigens hoef je de formule niet opnieuw in te typen. Je kan gewoon de formule **benadrukken**:



(dat doe je door met je muis links van  “bmi” te gaan staan, op de linkermuisknop te drukken en dan over de hele formule te “wrijven”). De benadrukte formule kun je dan kopiëren door

**Edit…Copy** en dan, bij de nieuwe inputstreepjes “>>>” te klikken en dan met **Edit…Paste** de formule te “plakken”. Dat spaart typwerk.

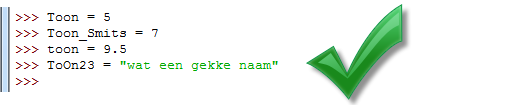
Overigens ziet Python het verschil tussen kleine en grote lettertjes : de variabele “bmi” kent hij, maar van “BMI” of “Bmi” heeft hij nog niet gehoord:



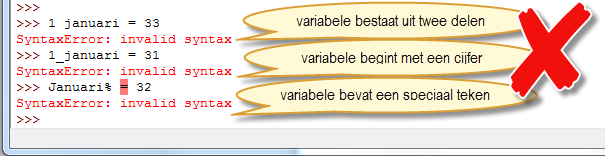
Alleen als we print(“bmi”) typen, weet hij wat ermee aan te vangen.

Namen van variabelen moeten aan elkaar geschreven worden en mogen hoofdletters, kleine letters, cijfers en een verbindingsstreepje “\_” bevatten, maar ze moeten met een hoofdletter of een kleine letter beginnen.

DUS :



zijn allemaal goede keuzes voor variabelen, maar…

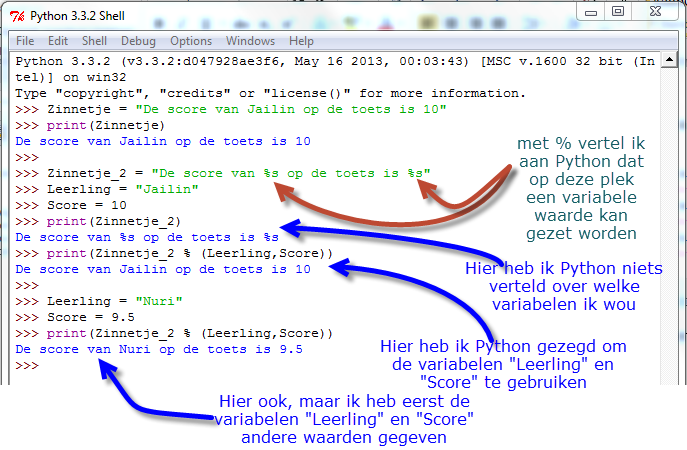


…zijn allemaal verkeerd.

1. Je kan stukken tekst ook variabel maken:

Zinnetje = “De score van Jailin op de toets is 10”

Zinnetje\_2 = “De score van %s op de toets is %s”

.

Dit is heel erg nuttig als je de computer wil programmeren om in te spelen op wat een gebruiker ingeeft. We laten dat straks zien, maar eerst moeten we het nog over “lijsten” hebben.

1. Je zou kunnen een lijst bewaren als een lange tekst.

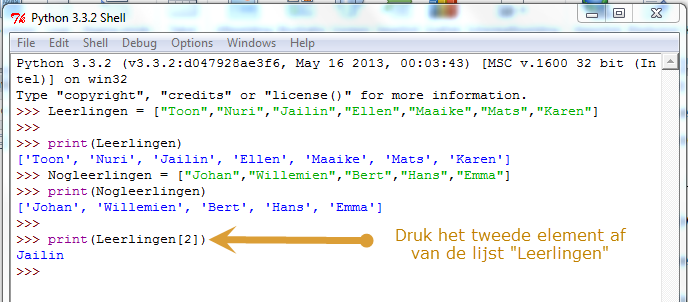
Bijvoorbeeld :

Leerlingen ="Toon,Nuri,Jailin,Ellen,Maaike,Mats,Karen"

Maar we kunnen ook een “lijst” maken, waarmee Python dan veel meer kan doen:

Leerlingen = ["Toon","Nuri","Jailin","Ellen","Maaike","Mats","Karen"]

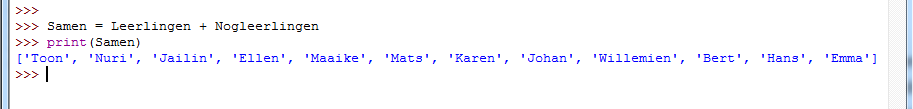
Het is een beetje meer typ-werk, maar daarna kun je zowel met de hele lijst als met de afzonderlijke elementen van je lijst werken.



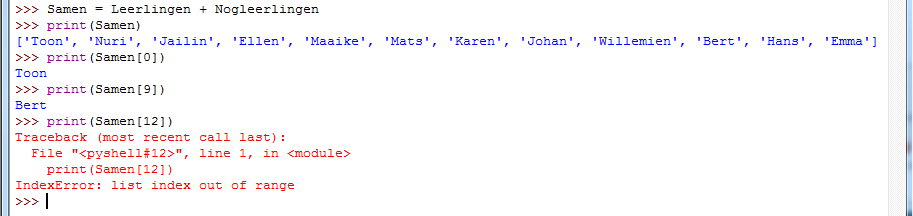
Heh ? Jailin is toch het *derde* element van de lijst. Ja, maar lijsten beginnen hun nummering bij 0, dus het eerste element is Leerlingen[0], het tweede element is Leerlingen[1], het derde element is dan Leerlingen[2], enzovoort…

Dat is niet zo logisch voor mensen, maar computers vinden dat dat juist wel logisch is.

Je kan ook lijsten bij elkaar “optellen” :



En zo hebben we dan



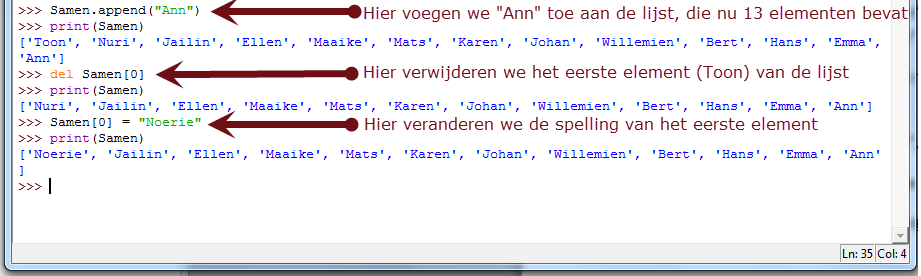
Het eerste element, Samen[0], is “Toon”.  
Het tiende element, Samen[9], is “Bert”.

De lijst heeft 12 elementen, dus als we naar Samen[12], het dertiende element, vragen, krijgen we een foutmelding.

**“IndexError : list index is out of range”,** hetgeen zoveel wil zeggen als

**“Nummeringsfout : de nummering van de lijst is buiten het bereik”**

Je kan ook makkelijk elementen toevoegen, vervangen en verwijderen uit de lijst :



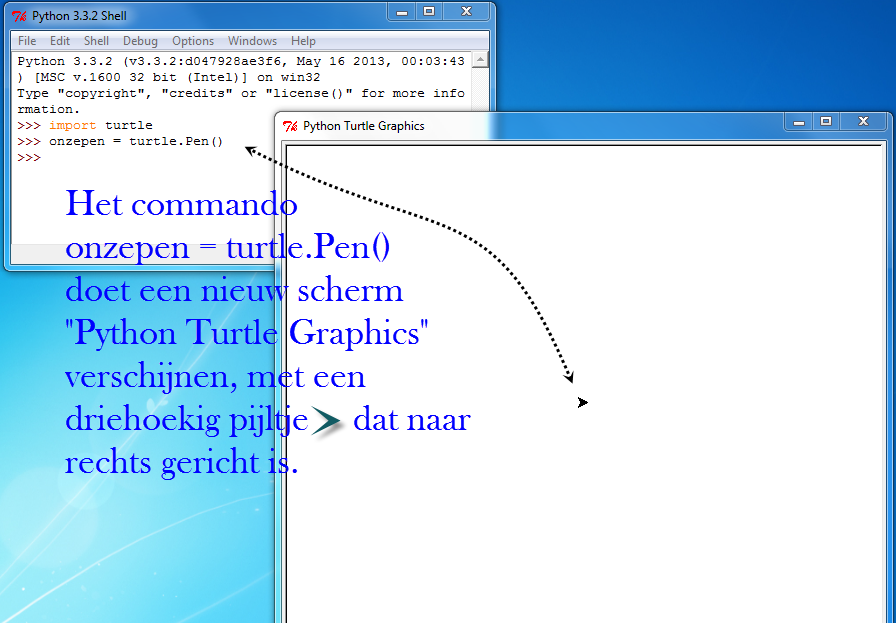
1. Tekenen met Python

Er is in Python een eenvoudige module waarmee we kunnen tekenen. Herinner je : een module een stukje programma dat eigenlijk op zichzelf werkt en waar we nieuwe dingen mee kunnen programmeren.

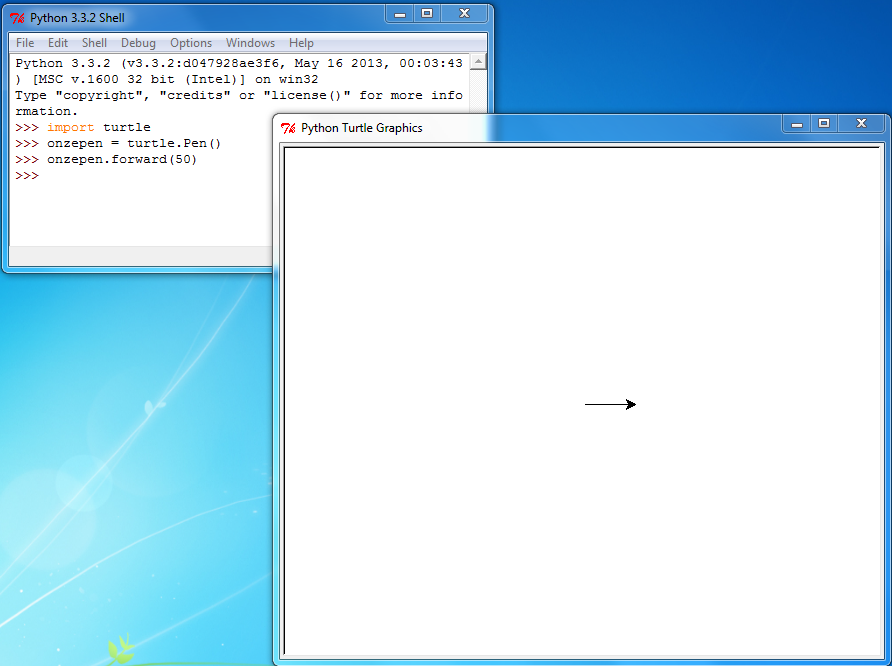
De tekenmodule in Python heet “turtle”. Dat is Engels voor “schildpad”. Net zoals de schildpad over de grond kruipt, zo kruipt de schildpad (met een “pen”) over het scherm.

De manier om een module te laden is met de instructie “import”. Denk aan “importeren”.

We typen dus : **import turtle**



Nu kunnen we het pijltje over het scherm laten bewegen met de instructie

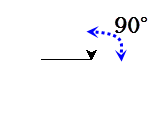
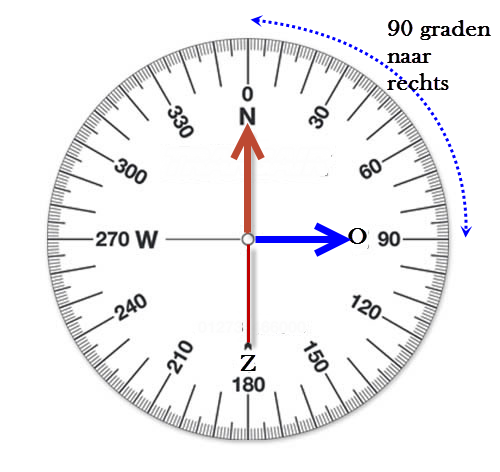
**onzepen.forward(50)**

Dit is python-taal dat wil zeggen : “neem de pen genaamd **onzepen**, en ga 50 stapjes vooruit”.

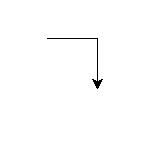
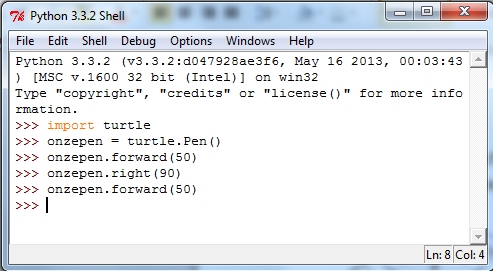
Dit is het resultaat : we zien dat het pijltje nu een steeltje gekregen heeft, en dat de pijlpunt 50 puntjes naar rechts is opgeschoven.

Om een lijn in een andere richting te trekken, moeten we eerst de pijlpunt van richting doen veranderen. Dat doen we met de instructie : **onzepen.right(90)**

“Right” is Engels voor “rechts”. En het cijfer dat tussen de haakjes staat, geeft aan hoeveel de pijl naar rechts moet draaien. Dat is ingegeven in “graden”.



Wat gaat er nu gebeuren als we opnieuw het commando **onzepen.forward(50)** geven ?



Juist : de schildpad gaat 50 stapjes vooruit, maar omdat het pijltje nu naar beneden wijst, wordt de lijn ook naar beneden getrokken.

Je kunt ook het pijltje naar links draaien. Dat doe je met de instructie

**onzepen.left(45)**

**En wat hebben we vandaag geleerd ?**

**Ten eerste** : je kan variabelen gebruiken in een programma. Het zijn “naamkaartjes” waar je mee kan rekenen. Bijvoorbeeld

leeftijdvanNuri = 11

En je kan stukken tekst ook variabel maken, met het “%” teken in een tekst en dan de waarde achteraan:

print(“De leeftijd van Nuri is %s” % leeftijdvanNuri)

De leeftijd van Nuri is 11

**Ten tweede** : je kan allerlei waarden in “lijsten” rangschikken. Deze lijsten kun je dan samenvoegen, en je kunt individuele elementen van een lijst opvragen.

lijst =[“Willemien”,”Jan”,”Piet”]

print(lijst[2])

Piet

**Ten derde** : in Python kan je tekenen met de “turtle”-module. Je doet dat door de module te importeren met “import turtle”, en vervolgens een “pen”-object aan te maken

onzepen = turtle.Pen()

En dan kan je **onzepen** allerlei opdrachten geven, zoals

onzepen.right(45), onzepen.forward(100), onzepen.up(), onzepen.down(), enzovoort.

Woordenlijst

|  |  |  |
| --- | --- | --- |
| Engels | Nederlands | Commentaar |
| Error  Syntax Error | Fout  Spellingsfout | Computers doen niets, tenzij jij het hen heel precies uitlegt wat ze moeten doen. |
| Python | Python | De Slang, maar ook, het komische televisieprogramma uit de jaren ‘70 |
| Shell | Schelp | Een programmaom-geving waar je recht-streeks instructies kan ingeven |
| File | Dossier | Een stukje van het geheugen waar we tekst, programma’s of andere informatie in opslaan. |
| Run | Letterlijk : Loop! Hier: Voer uit ! | Een programma uit-voeren heet ook wel eens een programma “runnen”. |
| Print | Druk Af ! | Een printer is eigenlijk een “drukker” |
| Software | Letterlijk : Zachte waren | Programma’s die op een computer draaien |
| Hardware | Letterlijk : Harde waren. Hardware betekent ook “ijzerwaren”. | Alle onderdelen en randapparatuur van een computer. |
| Module | Module | Een onderdeel van een programma dat op zichzelf een geheel vormt. |

1. Programmeerpuzzels

Probeer thuis de Schelp op te starten en maak de volgende berekeningen in Python :

1. Als er 29 kindjes in de kangoeroeklas zitten en die moeten allen een lespakket van 15 bladzijden krijgen, hoeveel vellen papier verbruiken we dan ? ………………………………………….
2. Wat is het laatste cijfer van

3 x 3 x 3 x 3 x 3 x 3 x….x 3 ? ………………

100 keer

1. Wat is de rest als je 9257 door 7 deelt ? ……………………………………..