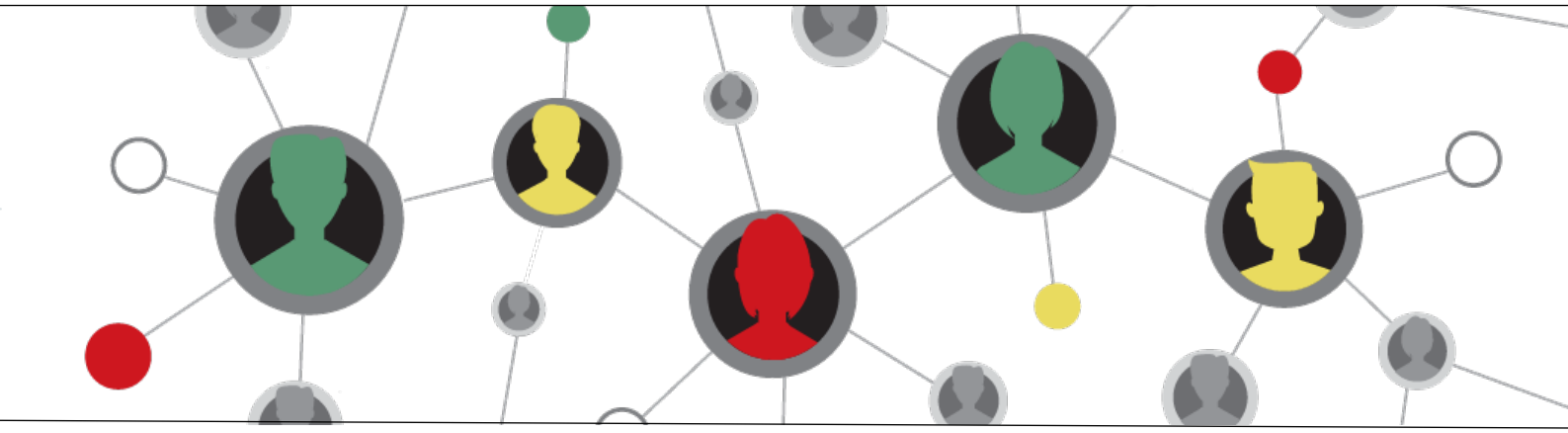
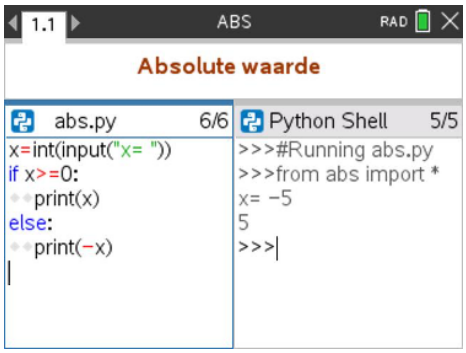
TI Python BootCamp

Deel 1 – TI Python Basics





T³ NEDERLAND

Teachers Teaching with Technology™

T³ VLAANDEREN

1. Wat is een variabele?

van Dale

varia'bel in staat om van getal, afmeting, vorm, plaats, enz. te wisselen, resp. veranderd te worden

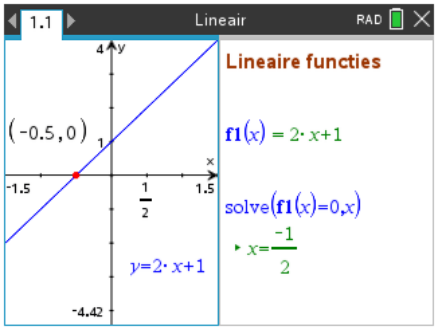
varia'bele (wisk) grootheid die in waarde enz. kan wisselen

WIKIPEDIA
De vrije encyclopedie

nl.wikipedia.org/wiki/Variabele (wiskunde)
Een variabele is de aanduiding van een willekeurig element van een verzameling. De variabele neemt waarden aan in die verzameling. Een variabele wordt meestal voorgesteld door een letter.

nl.wikipedia.org/wiki/Variabele (informatica)
Een variabele is een term die gebruikt wordt in verband met programmeren. In de code van een computerprogramma associeert een variabele een naam met een of meer geheugenadressen.

Voor reële functies $f:\mathbb{R} \longrightarrow \mathbb{R}: x \longmapsto 2x+1$ noemen we x de variabele en $f(x)=2x+1$ het beeld van x. $f(x)$ wordt ook genoteerd d.m.v. y: $y=f(x)$ of $y=2x+1$ of $y(x)=2x+1$.



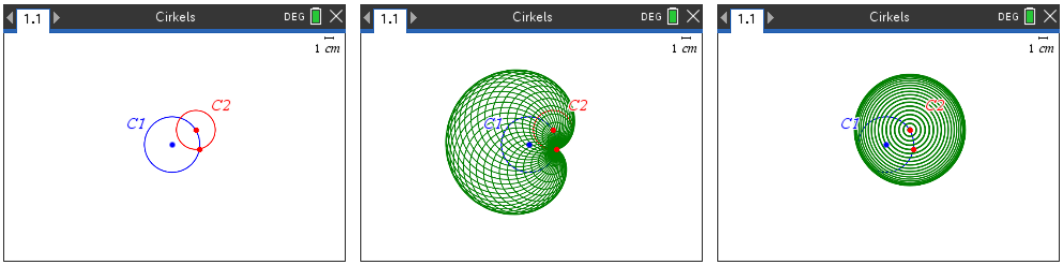
Hierbij wordt x de onafhankelijke variabele genoemd en $y=2x+1$ de afhankelijke variabele, waarbij het lineaire verband $y=2x+1$ aangeeft hoe y verandert i.f.v. x. $y=2x+1$ is de vergelijking van de grafiek van de functie f.

Het bepalen van de nulpunten van $f(x)=2x+1$ (snijpunten x-as) komt neer op het oplossen van de vergelijking $y=0 \Leftrightarrow 2x+1=0$ en omgekeerd.

Wat als we de rol van x en y omdraaien, $x(y)$?

$$y(x)=2x+1 \Leftrightarrow x(y)=\frac{1}{2}y-\frac{1}{2}$$

Onafhankelijke en afhankelijke variabelen kunnen goed geïllustreerd worden a.h.v meetkundige construties. Bijvoorbeeld, teken een cirkel *C1* (onafhankelijk) en een cirkel *C2* (afhankelijk) met middelpunt en radiuspunt op de cirkel *C1*. Voor de cirkel *C1* kan het middelpunt willekeuring verplaats worden als ook de straal verandert. Het veranderen van de cirkel *C2* is beperkt tot het verplaatsen van middel- en radiuspunt op de cirkel *C1*. Hieronder de meetkundige plaatsen bepaalt door de cirkel *C2* met als eerst het variabele middelpunt en dan het radiuspunt.



En nog een andere visualisatie van onafhankelijke en afhankelijke variabelen.

Onafhankelijke variabele Hoeveelheid water	Afhankelijke variabele Hoogte van de plant
	

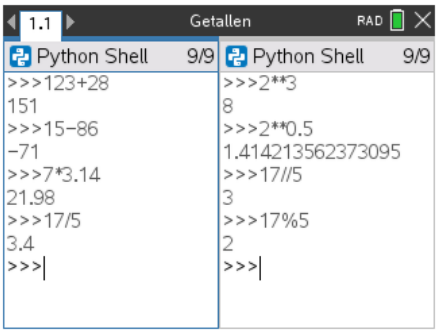
2. Getallen in Python

Python kent tal van verschillende getal-types. We beperken ons hier tot gehele, Integers (int), en decimale getallen, Floating Point (float).

2.1. Eenvoudig rekenwerk

De meest gebruikte operaties voor rekenwerk in Python zijn de onderstaande. We voeren enkele berekeningen uit in de Python Shell

Operator	Bewerking
+	Optelling
-	Verschil
*	Vermenigvuldiging
/	Deling
**	Machtsverheffing
//	Vloerdeling
%	Modulo (rest)



Bij het invoeren van een bewerking in de Shell of het gebruik in code wordt de operator rood weergegeven.

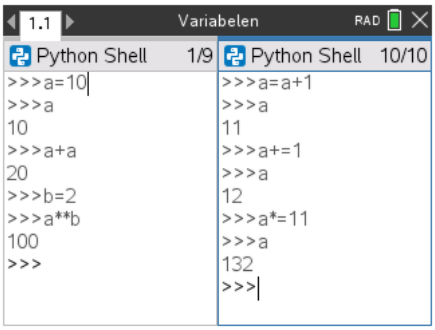
2.2. Declaratie van variabelen

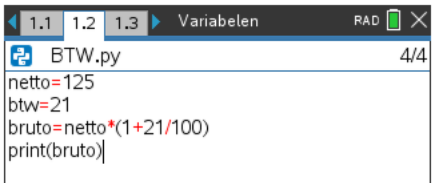
In Python wijzen we een waarde (data) toe aan een variabele met het gelijkheidsteken, =.

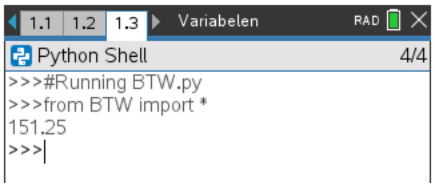
Bij een herdeclaratie van de variabele a kan de uitdrukking a = a + 1 verkort noteren met a += 1. Hetzelfde geldt voor -=, *=, /=, ...

Merk op dat na het ingeven van a = 10, de waarde niet getoond wordt.

Indien we het volgende programma dat de nettoprijs (exclusief BTW) omzet in de brutoprijs (inclusief BTW) runnen, krijgen we het volgende resultaat in de Python Shell. Om ook effectief de brutoprijs als output in de Shell te tonen, gebruiken we het print()-statement.





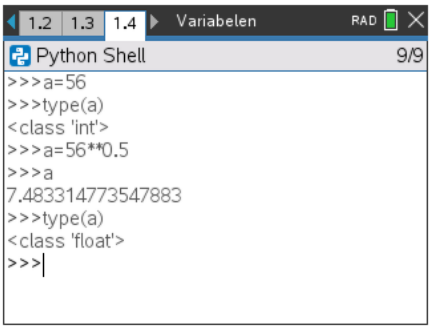


De naam van een variabele moet aan de volgende regels voldoen:

- niet starten met een getal
- geen spaties
- geen gebruik van de volgende symbolen:'",<>/?\()@#$%^&*~-+
- onder Python-gebruikers is kleine letters een algemene afspraak
- vermijd het gebruik van worden met een speciale betekenis in Python: float, int, ...

Merk op dat Python dynamisch omgaat met datatypes voor variabelen. M.a.w., je kan een variabele een herdeclaratie geven van een ander data type.

Het type van een variabele verifieer je met de type()-statement.



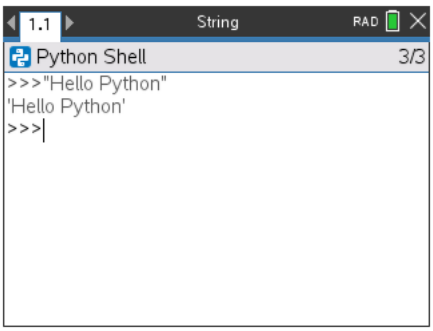
3. Woorden in Python

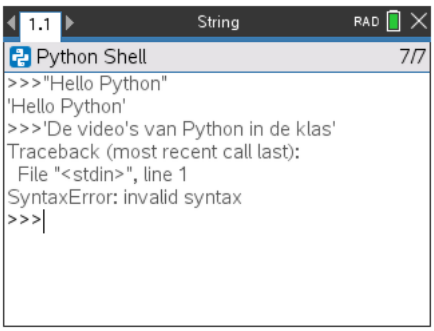
Woorden of Strings worden in Python gebruikt om tekst te bewaren. Een string is in feite niets anders dan een rij van karakters in een specifieke orde. Hetgeen betekent dat we m.b.v. indexering iedere letter van een woord kunnen aanspreken.

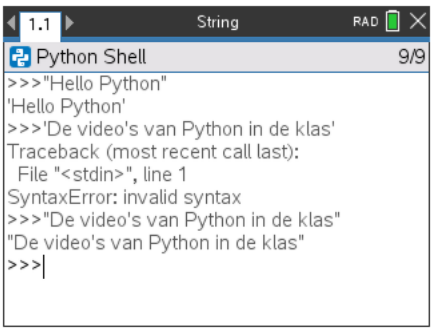
3.1. Definiëren van een string

Een string of woord plaats je tussen aanhalingstekens, b.v. “Hello Python”. Je kan ook gebruik maken van enkele aanhalingstekens maar dat geeft niet altijd het gewenste resultaat: ‘De video’s van Python in de klas’.

De error komt omdat het aanhalingsteken van video’s de string afsluit. Het gebruik van een enkel aanhalingsteken kan perfect binnen dubbele aanhalingstekens: “De video’s van Python in de klas”.

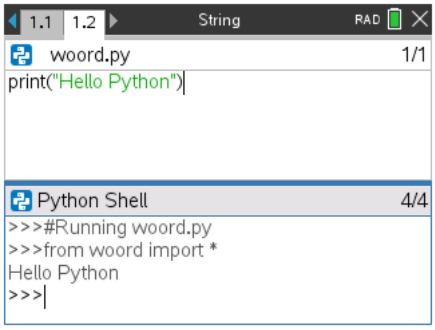






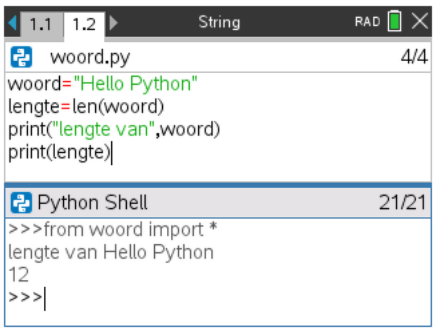
3.2. Printen van een string

Het print()-statement kan ook hier gebruikt worden voor het tonen van een string van uit een programma. Onderstaande print()-statements kunnen samengevoegd worden door gebruik te maken met de \n-operator die staat voor een nieuwe lijn.



3.3. Lengte van een string

Met de functie len() kan je het aantal karakters van een string controleren.

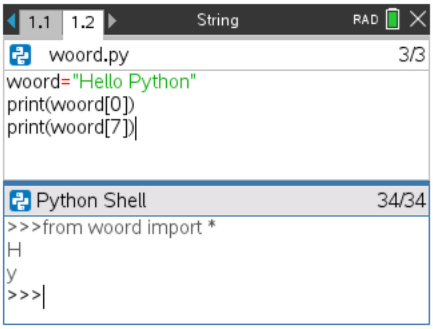


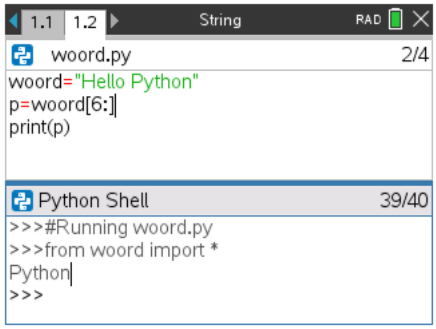
3.4. Indexering

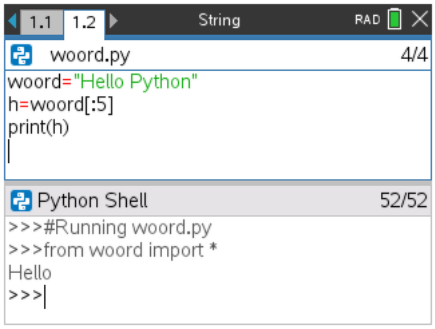
Zoals eerder aangeven is de volgorde van de karakters in een woord vast. Ieder karakter kan aangesproken worden door de index die de positie weergeeft van het karakter in de string.

Python gebruikt vierkante haakjes voor het aangeven van een index. Merk op dat het eerste karakter van een woord overeenkomt met index 0.

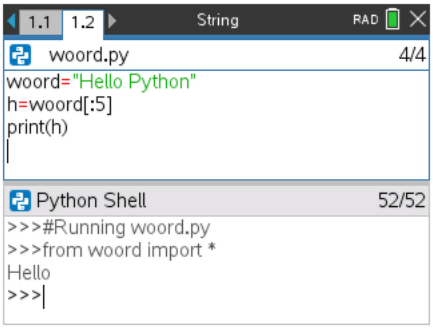
Men kan ook vanaf of tot een bepaalde index alles weergeven en/of bewaren in een andere variabele. In Python noemt men dit slicing. In Python betekent woord[:3], neem alle karakters (elementen) van index 0 tot 3, index 3 niet inbegrepen.

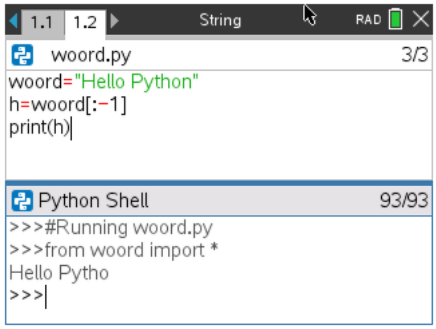


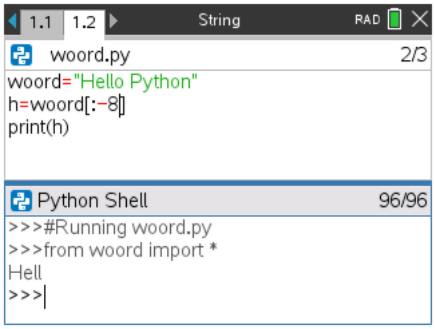




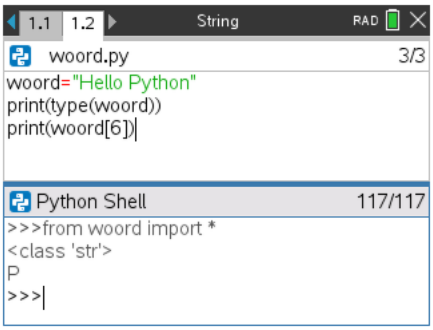
Met negatieve slicing kunnen we tellen vanaf het einde van een woord.

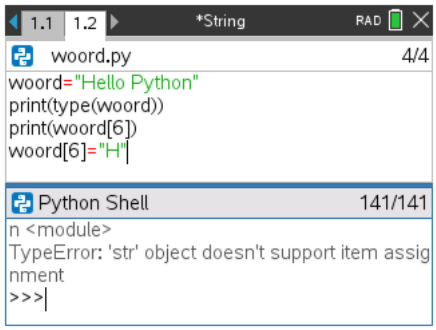


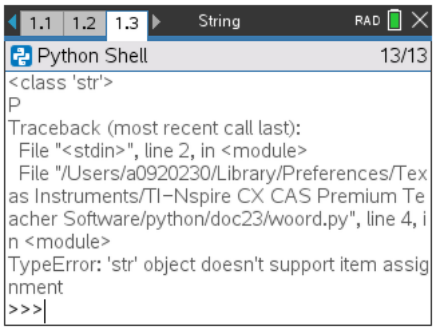




Strings in Python hebben de eigenschap onveranderlijkheid: als een variable gedeclareerd is als een string, kunnen de karakters van de varaiable (string) niet worden veranderd.







3.5. String-methodes

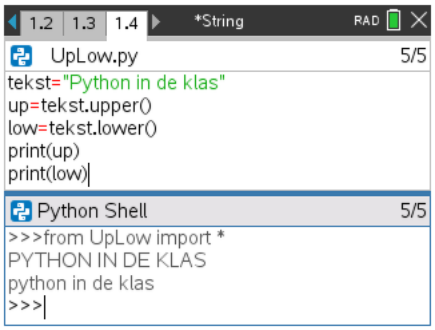
In Python is alles een object: een specifiek getal is een object van b.v. de klasse int (geheel getal) of float (decimaal getal) en “Hello Python” een object uit de klasse str (string). Wat dit exact betekent, verduidelijkingen we meer in details in het gedeelte Object georiënteerd programmeren.

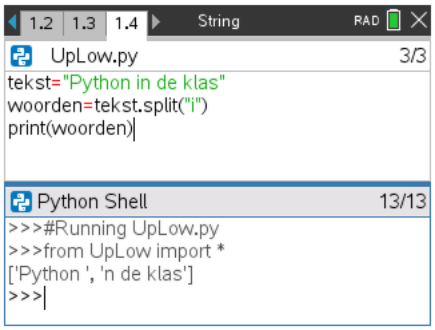
Voor iedere klasse van objecten, zijn er ingebouwde methodes (of functies) die op de objecten kunnen uitgevoerd worden. Voor het uitvoeren van een methode op een object gebruiken we een punt gevolgd door de methodenaam: object.methode() of object.methode(parameters).

Enkele voorbeelden voor de string tekst = ”Python in de klas”.

upper()	alle karakters in hoofdletters
lower()	alle karakters in kleine letters
split()	splitsen bij een spatie
split(“i”)	splitsen bij een specifiek element (element niet inclusief)

“Python in de klas”.split() geeft als resultaat [‘Python’,’in’,’de’,’klas’], een lijst met alles element de woorden van de tekst. Hetgeen ons brengt tot het volgende data type Lijsten.



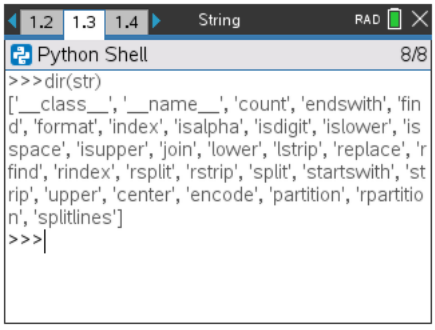


Voor een overzicht van alle ingebouwde methodes voor een bepaalde klasse, voer de dir()-statement uit in de Python Shell.

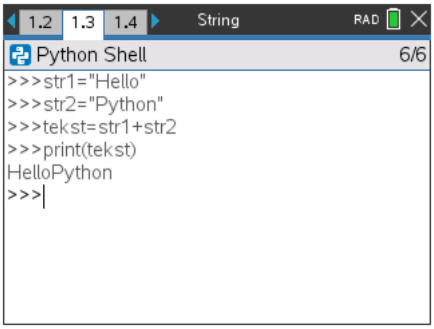
Wat en hoe over deze methodes kunt u vinden @ www.python.org of www.micropython.org.

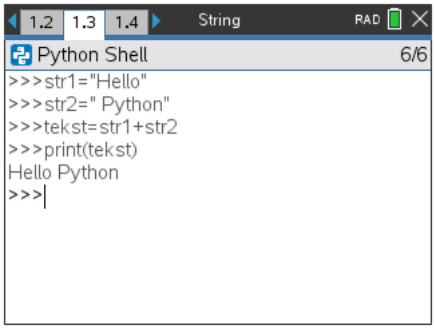
Of natuurlijk uitproberen is ook een zeer zinvolle leerschool.

De implementatie van Python in TI-technologie is gebaseerd op MicroPython, versie 3.4.



Het samenvoegen van twee strings kan heel eenvoudig door gebruik te maken van de +-operator.





4. Lijsten in Python

Lijsten kan je beschouwen als de meest algemene versie van een rij in Python. Zoals voor strings is de volgorde van belang maar lijsten zijn veranderlijk, m.a.w. ieder element van een lijst kan, na definitie, veranderd worden gebruikmakend van de index van een element.

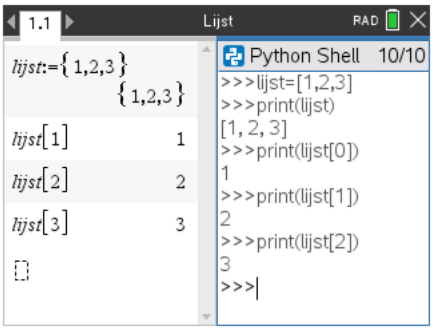
4.1. Definiëren van een lijst

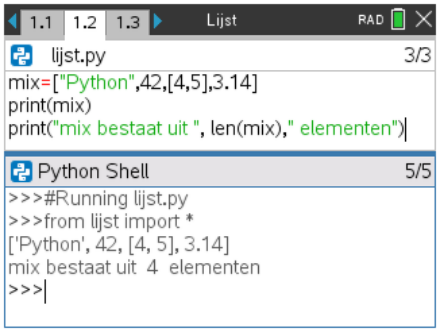
Het grote verschil met lijsten voor TI-technologie is dat de indexering van lijsten in Python start vanaf 0 i.p.v. vanaf 1 zoals voor TI-technologie.

Bovendien is de syntax van een lijst verschillend:

- Python lijst = [1,2,3,4,5]
- TI lijst := {1,2,3,4,5}

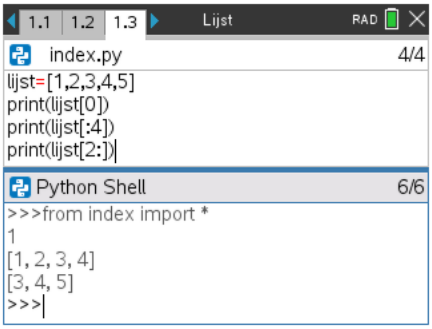
Python lijsten kunnen verschillende soorten van data types bevatten. Voor lijsten bepaalt het len()-statement ook het aantal elementen van een lijst.

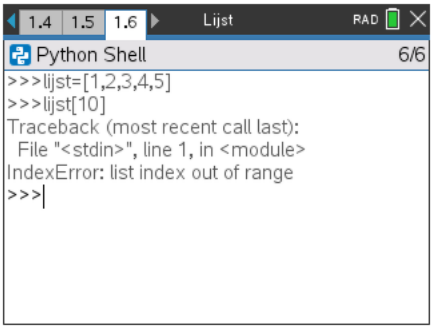




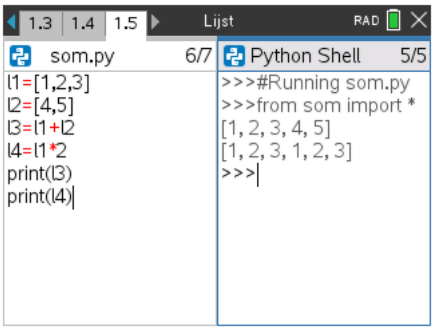
4.2. Indexering & Slicing

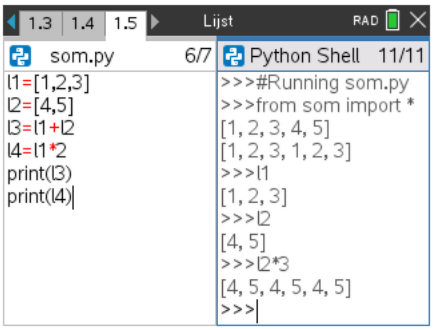
Indexering en slicing voor lijsten werkt hetzelfde als voor strings, hieronder enkele voorbeelden. Indien er geen element is voor een bepaalde index geeft Python de volgende error-boodschap.





Lijsten kunnen in Python eenvoudig samengevoegd worden m.b.v. +. En met * kun je een lijst verdubbelen. Merk op dat deze operatie op lijsten de originele lijsten niet verandert, en om het resultaat te bewaren er een nieuwe variabele moet gedeclareerd worden.

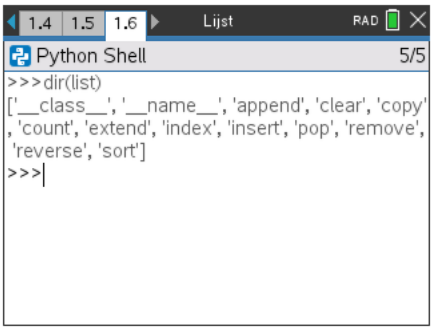


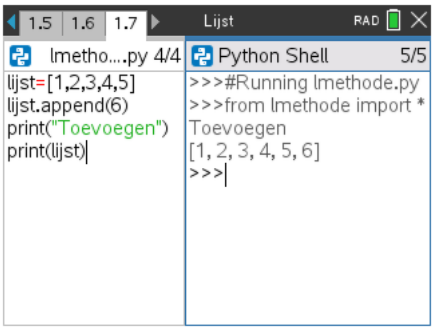


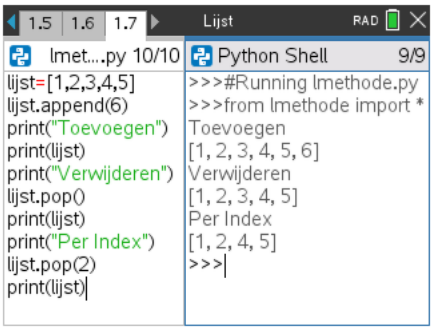
4.3. Lijst-Methodes

Hieronder enkele methodes die uitgevoerd kunnen worden op lijsten.

- append() voeg een element permanent toe aan het einde van de lijst
- pop() verwijder permanent het laatste element, of specifiek pop(1)
- pop(i) verwijder permanent het element met index i

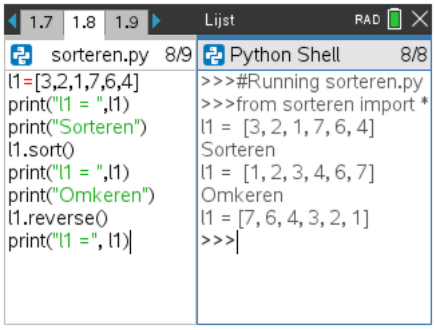




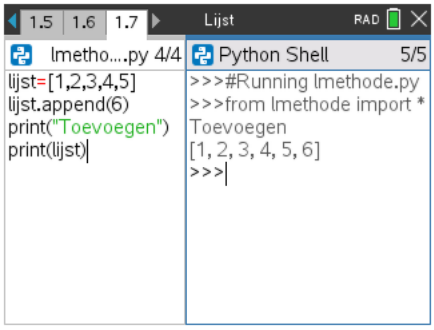


We bekijken ook even de methodes sort() en reverse(). Het manipuleren van een lijst verandert de lijst ook hier permanent.

l1=[3,2,1,7,6,4]

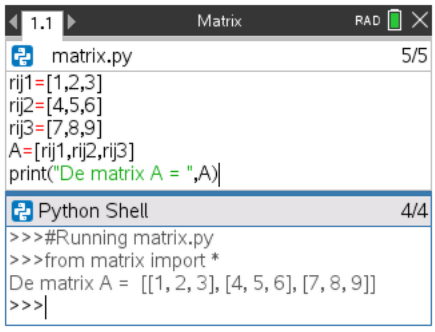


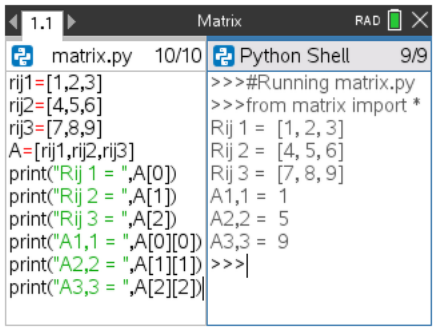
l1=["y","x","z","t","u"]



4.4. Matrices

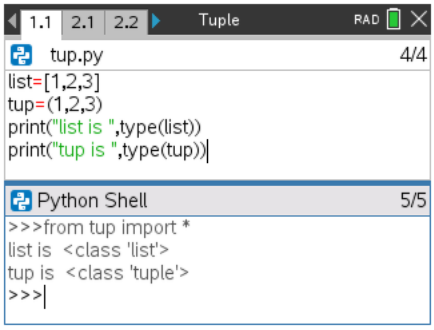
Matrices kunnen voorgesteld worden gebruikmakend van een lijst bestaande uit lijsten.

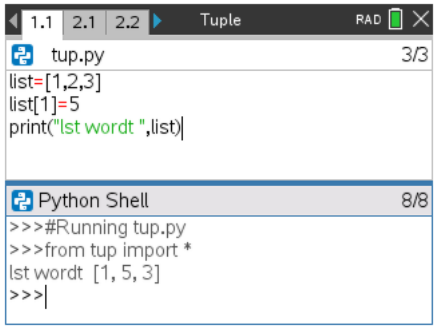


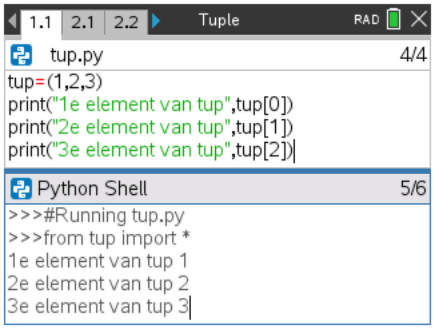


5. Tuples

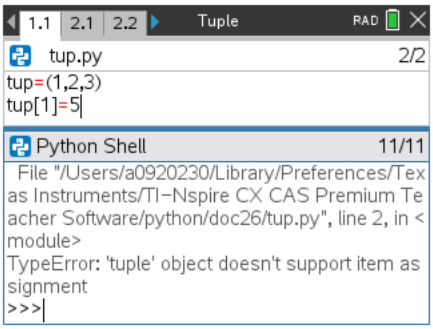
Het data type Tuple is vrij gelijkaardig als een lijst alleen dat, zoals voor strings, de elementen van een tuple niet kunnen veranderd worden.

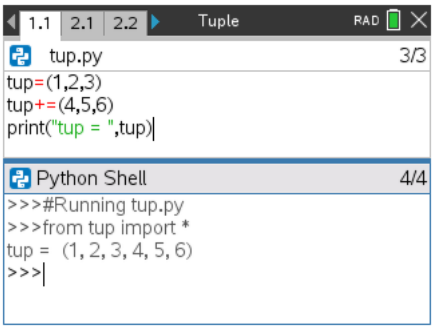




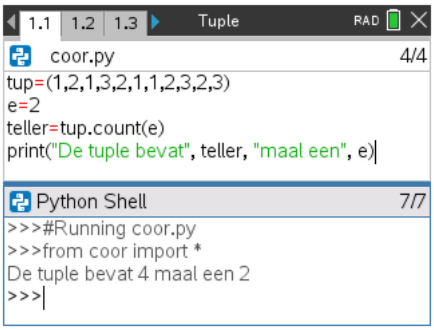


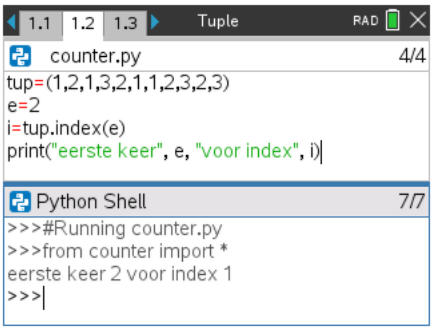
Indien we een element van een tuple proberen te veranderen, b.v. tup[1]=5, krijgen we de onderstaande error-boodschap. Wel kunnen elementen toegevoegd worden met de + operator.





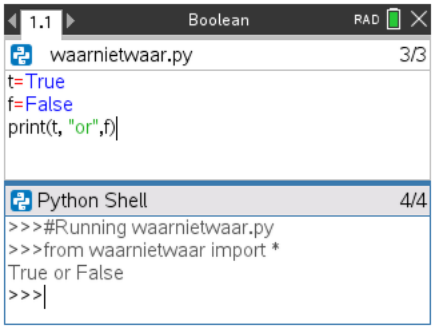
Voor tuples hebben we slechts twee methodes beschikbaar: count en index.

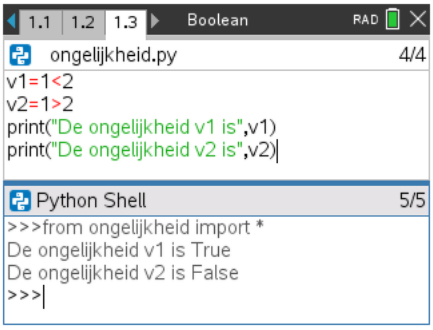




6. Boolean - True or False

Python heeft twee voorgedefinieerde booleaanse objecten: True en False. True en False zijn in feite niet anders dan de getallen 1 en 0.

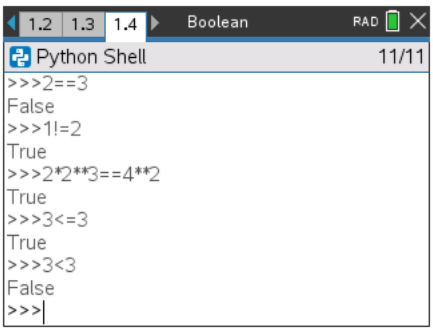




6.1. Vergelijkingsoperatoren

Hieronder de lijst met vergelijkingsoperatoren.

Operator	Bewerking
==	gelijk aan
!=	niet gelijk aan
>	groter dan
<	kleiner dan
>=	groter of gelijk
<=	kleiner of gelijk



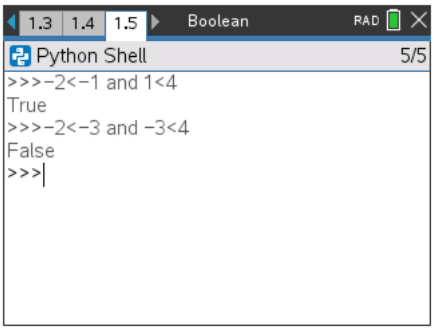
6.2. And en/of Or

Met de statements and en or kunnen verschillende logische tests met mekaar gecombineerd worden om meer complexe testen te construeren.

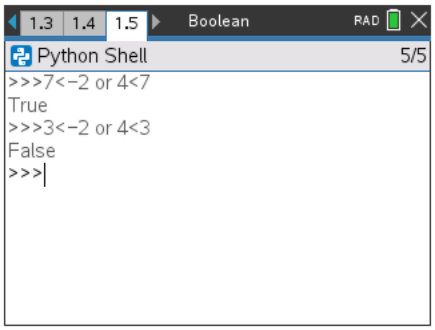
A	B	A and B	A or B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Enkele voorbeelden

$|x-1|<3 \Leftrightarrow -2<x$ en $x<4$



$|x-1|>3 \Leftrightarrow x<-2$ of $4<x$



1. Input

Met het input()-statement kan via de shell input geleverd worden aan een Python program, b.v. als volgt:

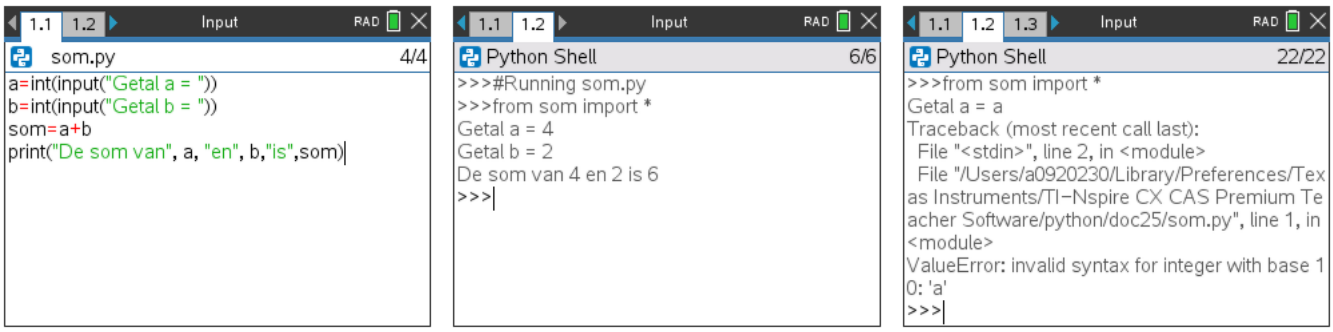
Hiernaast hoe stap voor stap

- Input van de twee getallen
- Berekenen van de som
- Tonen van de som

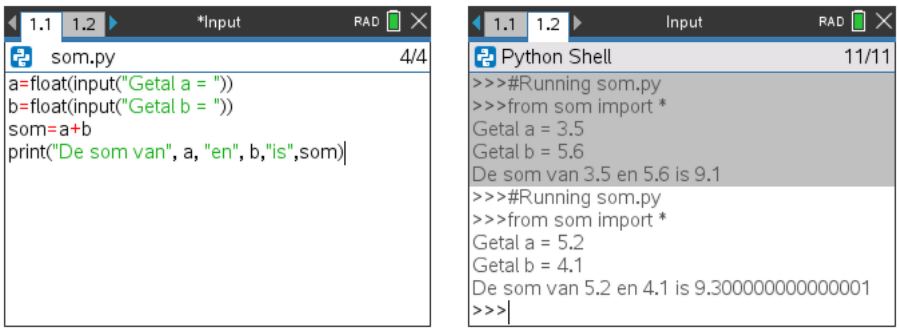
Alleen is de uitkomst van de som 4 + 2 niet echt wat we verwachten van een rekenkundige som.

Input in Python wordt altijd geïnterpreteerd als strings.

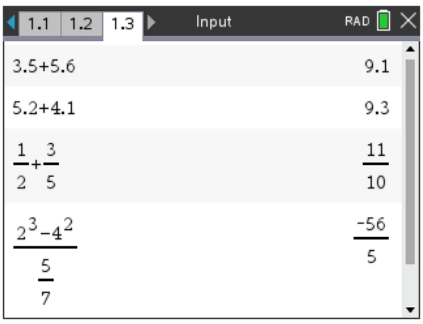
Om dit op te lossen, maken we gebruik van de ingebouwde Python-functie int(). Indien we bij de input niet een geheel getal ingeven, krijgen we de onderstaande foutmelding. Hoe we een foutmelding bij een verkeerde input kunnen voorkomen, bekijken we later in deze bootcamp.



Indien we willen werken met decimale getallen vervangen we int() door float().



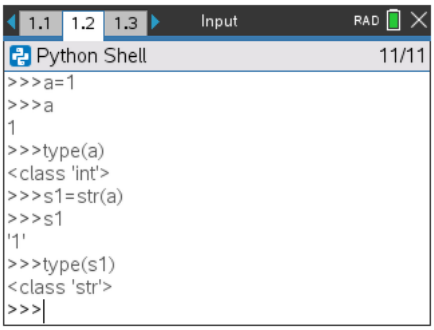
Wiskundig rekenen met decimale getallen is niet de sterkste kant van de programmeertaal Python. Het resultaat van een tweede run van het programma som.py geeft het resultaat 9.300000000000001.

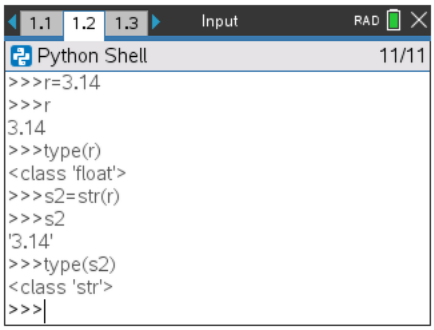


Dit heeft te maken met het feit dat Python gebruik maakt van binaire benaderingen afhankelijk van de hardware. Meer info hierover: docs.python.org/3/tutorial/floatingpoint.html.

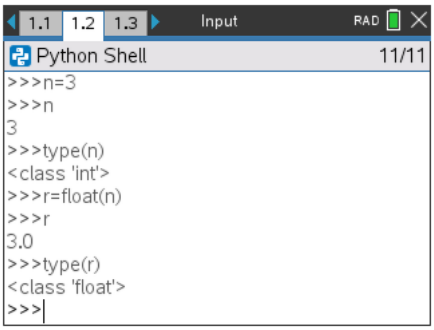
TI-technologie beschikt over een sterke wiskundige engine (numeriek of CAS) die best gebruikt wordt voor het uitvoeren van wiskundige berekeningen.

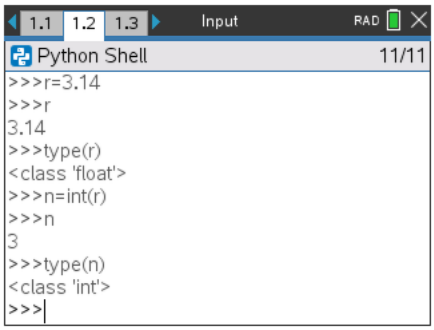
Omgekeerd kunnen getallen ook omgezet worden in strings met de str()-statement.





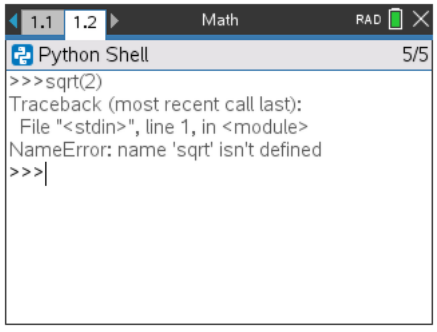
int() en float() kunnen ook gebruikt worden om gehele getallen om te zetten in een decimaal getallen en omgekeerd.





2. Extra wiskundige functionaliteit

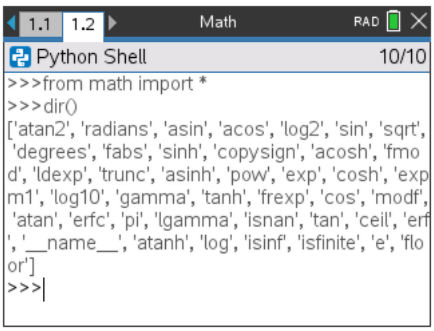
Indien we sqrt(2) willen berekenen, krijgen we de onderstaande foutmelding, m.a.w. de functionaliteit vierkantswortel in geen ingebouwde functie.

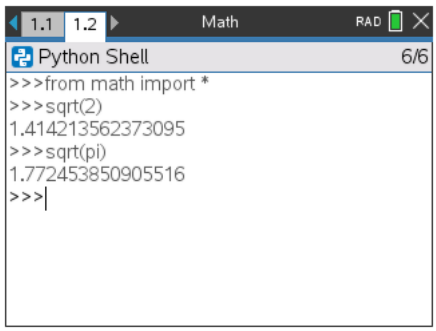


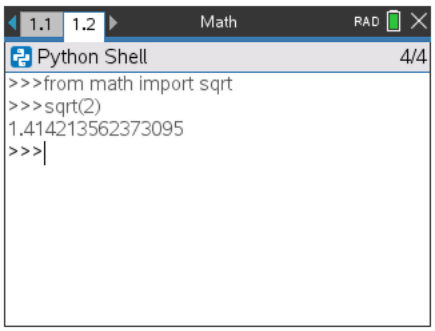
2.1. De module Math

Na het importeren van de module Math, met de code “from math import *”, kunnen we gebruik maken van heel wat extra wiskundige functies. dir() produceert een lijst met alle functionaliteit in de module Math.

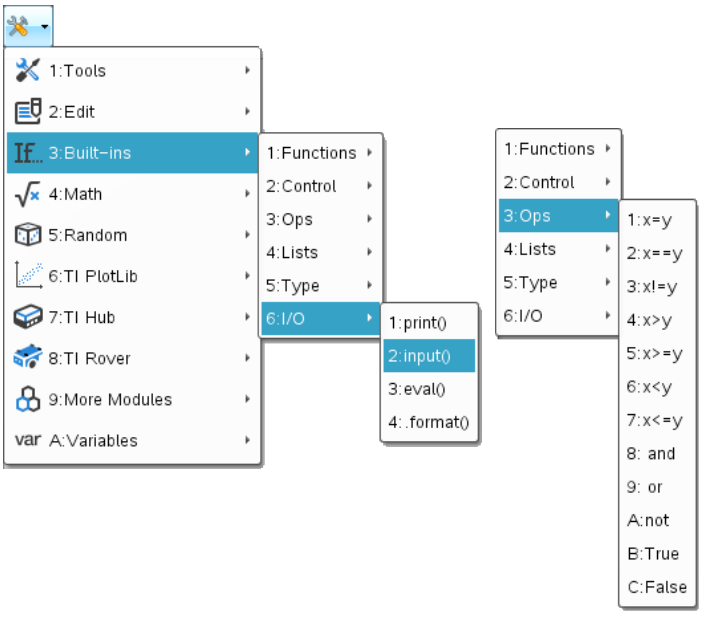
Indien je b.v. enkel sqrt() uit de module Math wil importeren, gebruik je de code “from math import sqrt”.

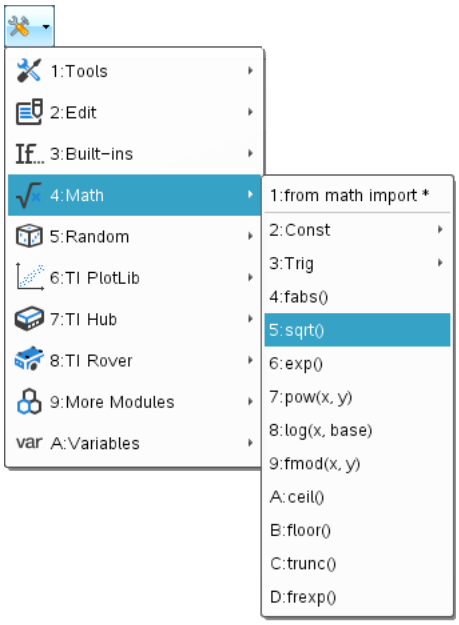






De meeste gebruikte functie, zowel voor de built-in functies als voor de geïntegreerde modules, zijn beschikbaar in een menustructuur.



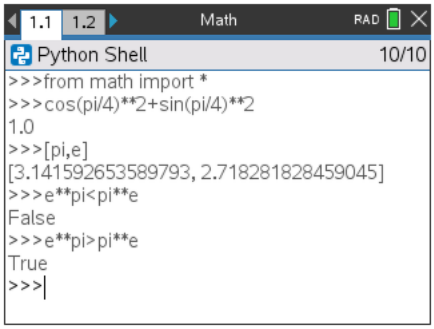


Door deze menustructuur is het eenvoudig om commando’s en functies terug te vinden, versnelt het ingeven van code (zeker voor het programmeren met de handheld) en gebruikt je steeds de juiste syntax.

Nog enkele andere voorbeelden van de module Math.

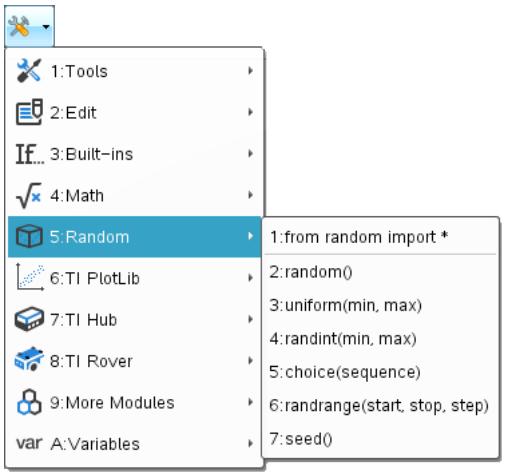
e^{π} or π^{e}

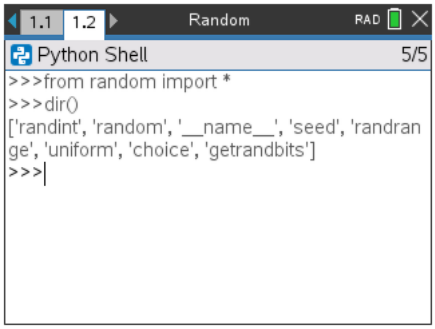
Which One is Greater?



2.2. De module Random

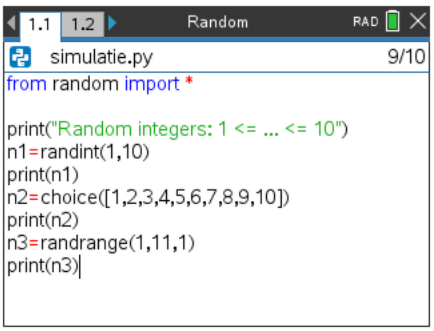
De module Random bevat functies die gebruikt kunnen worden om simulaties uit te voeren met een Python programma.

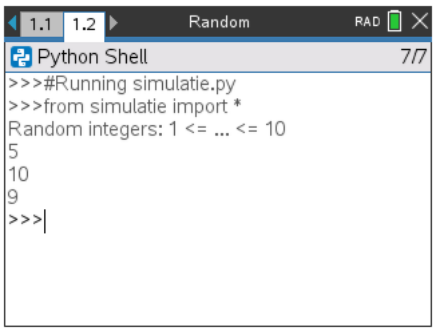


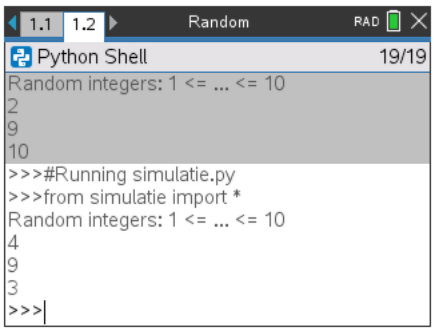


De onderstaande code genereert telkens een willekeurig geheel getal n tussen 1 en 10, 1 en 10 inclusief.

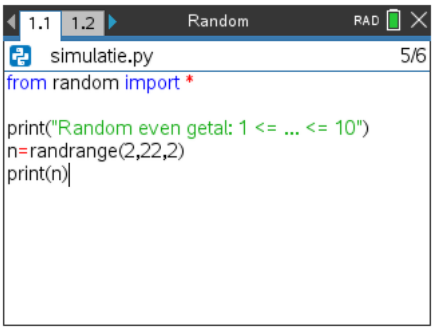
- randint(min,max) — random geheel getal tussen min en max: min <= … <= max
- choice(lijst) — random keuze uit een lijst
- randrange(start,stop,step) — random geheel getal tussen start en stop met stapgrootte step

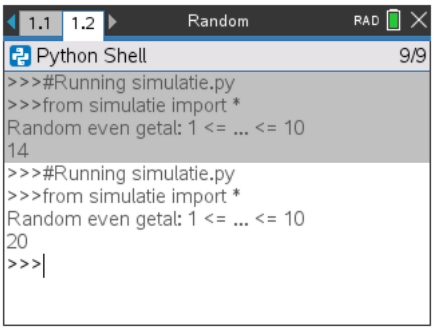


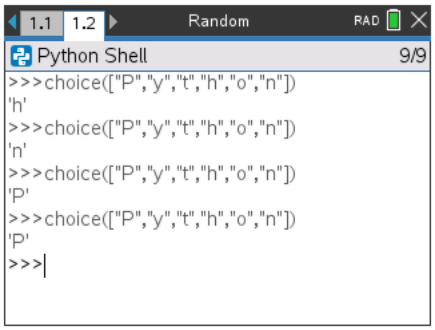




Met randrange kunnen we at random een even getal generen tussen tussen 1 en 20, 20 inbegrepen: randrange(2,22,2). Merk op dat de choice()-statement ook andere data kan bevatten dan gehele getallen.

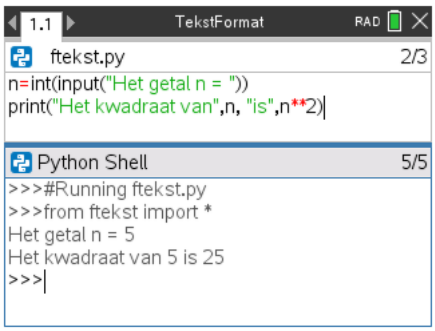






3. Print

We hebben het print()-statement reeds gebruikt om de waarde van een variabele weer te geven, al dan niet in combinatie met tekst.

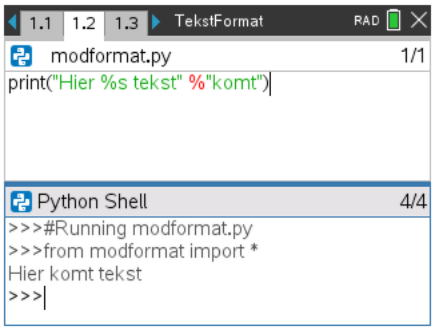


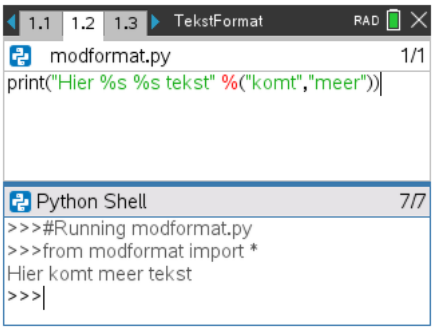
We bekijken twee string-methodes om de output van een Python programma te formateren:

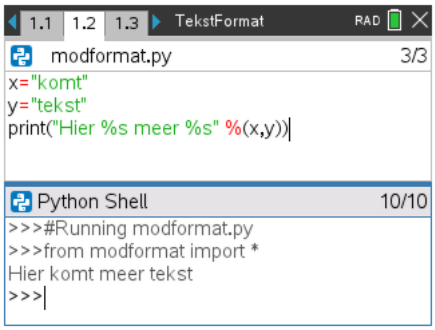
- % operator (de oudste)
- format()-methode (de verbeterde)

3.1. Formateren met de %-operator

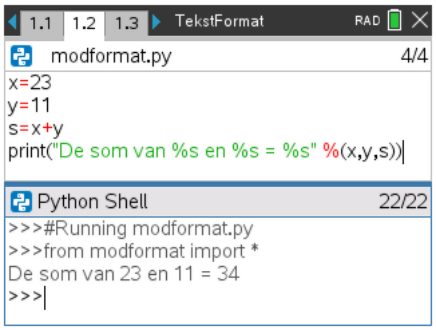
Enkele voorbeelden



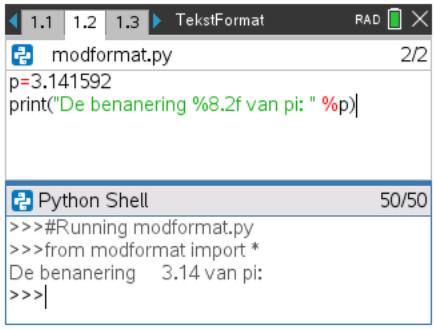


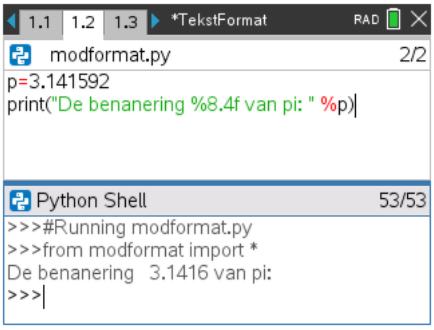


Tuurlijk geldt dit niet alleen voor tekst, maar ook voor andere data types zoals getallen.



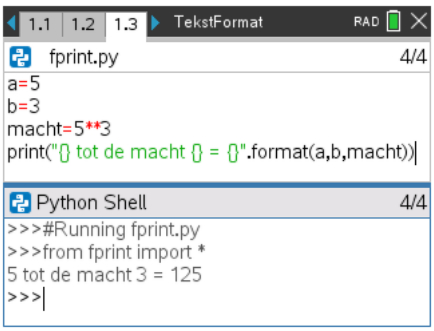
De format %8.2f betekent dat we minimum 8 karakters gebruiken om het getal weer te geven (eventueel opgevuld met spaties) en .2f bepaalt dat er slecht twee cijfers na de komma worden weergeven.

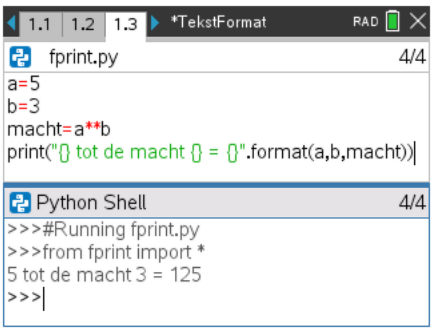




3.2. Formateren met format()

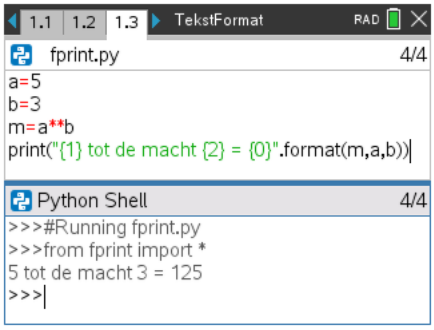
Een betere manier om strings te formateren in een print()-statement is gebruik te maken van de format()-methode:
print("{} tot de macht {} = {}".format(a,b,macht)).



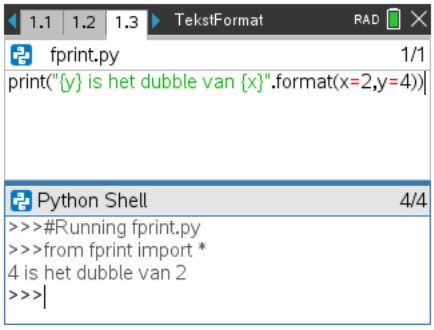


Voordelen van het gebruik van de format()-methode zijn o.a.:

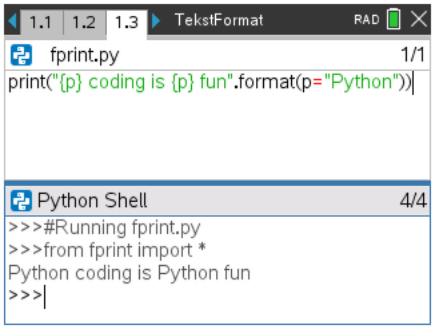
- In te voegen objecten kunnen geïndexeerd worden



- In te voegen objecten kunnen geassocieerd worden met labels

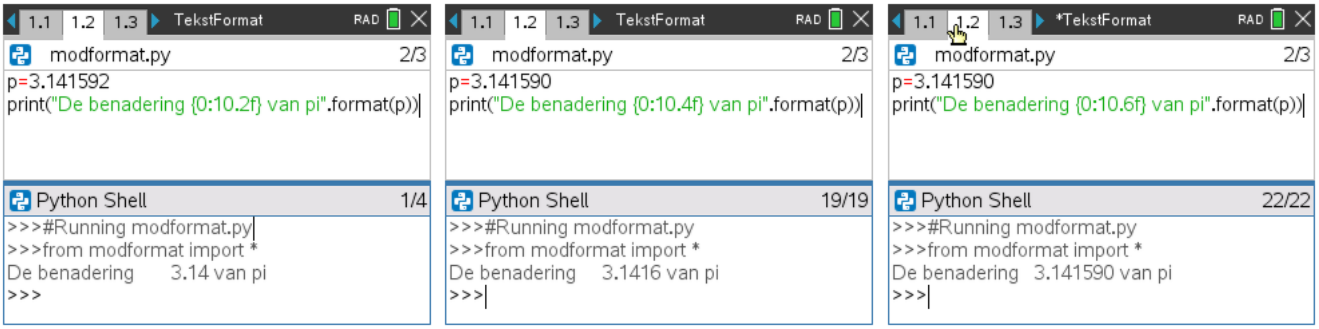


- In te voegen objecten kunnen meervoudig gebruikt worden



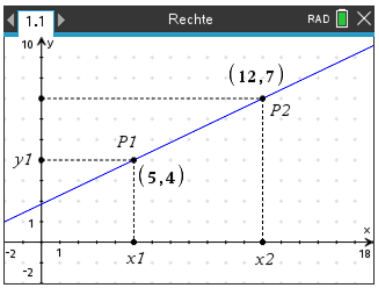
Het bepalen van het aantal decimalen na de komma met de format()-methode gaat als volgt.

Voor {0:10:2f} is 0 de index van het object,
10 het aantal minimale karakters (eventueel opgevuld met spatie) en
2f het aantal decimalen na de komma.



1. Inleidend voorbeeld

We tekenen een rechte in de TI-Nspire CX Graphs App door de punten $P1$ en $P2$.



We bewaren de coördinaten van de punten $P1$ en $P2$ in de volgende TI-Nspire CX variabelen $x1, y1$ en $x2, y2$ als volgt:

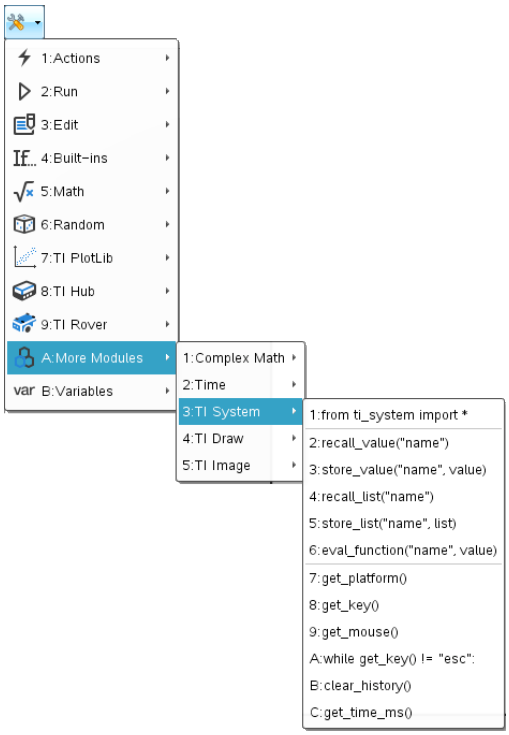
$P1\ (x1, y1)$ & $P2\ (x2, y2)$.

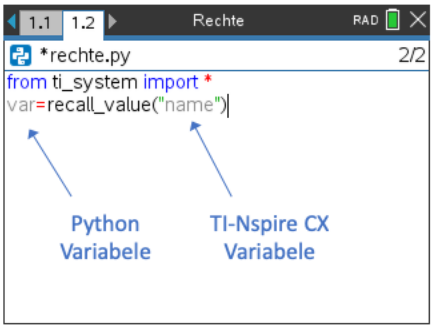
De vergelijking van de rechte is gelijk aan:

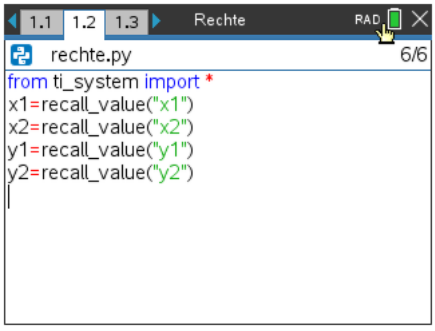
$$y - y1 = \frac{y2-y1}{x2-x1}(x - x1).$$

Met de module TI System kan variabelen vanuit TI-Nspire CX geïmporteerd worden in een Python-programma als Python variabelen. Na het importeren van de module ti_system selecteren we het commando recall_value().

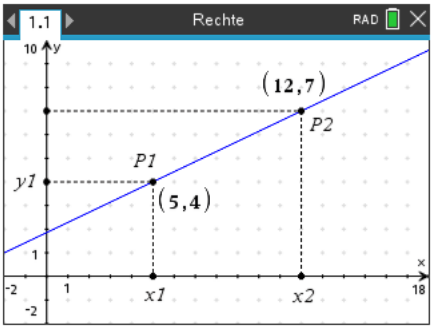
De onderstaande syntax-template verschijnt die aangeeft hoe de Python variabele te declareren op basis van de TI-Nspire CX variabele.

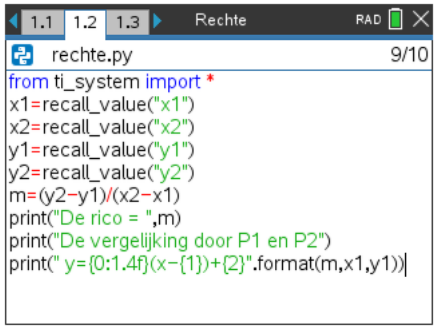




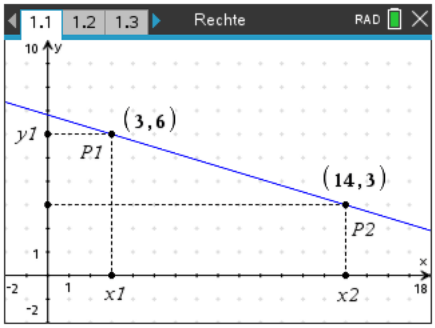


We berekenen de richtingscoëfficiënt m en bepalen de rechte door de twee punten $P1$ en $P2$.

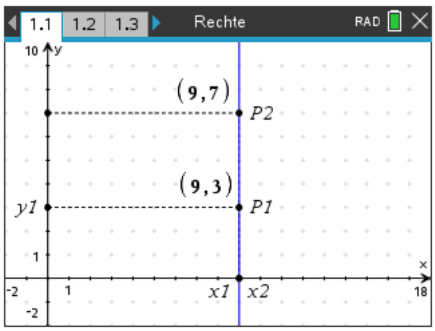


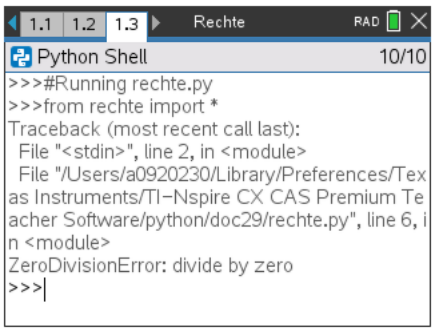


Als we de punten verplaatsen en het programma opnieuw runnen krijgen we de nieuwe vergelijking:



Maar wat met een verticale rechte?





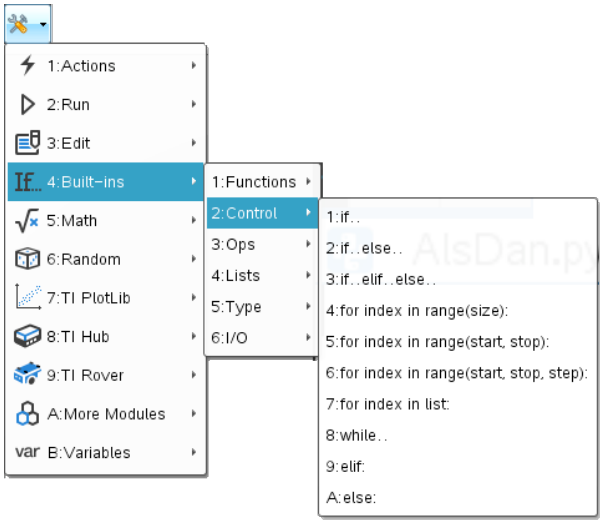
2. If-Statements

Om bovenstaande foutmelding te voorkomen testen we de conditie x1 == x2 of x1 != x2.

Hiervoor selecteren we in het Control-menu de optie 2: if..else.. en passen ons programma als volgt aan.

Om commentaar in te voeren start je de regel met #.

De template van het if-statement verschijnt automatisch in de Program Editor.

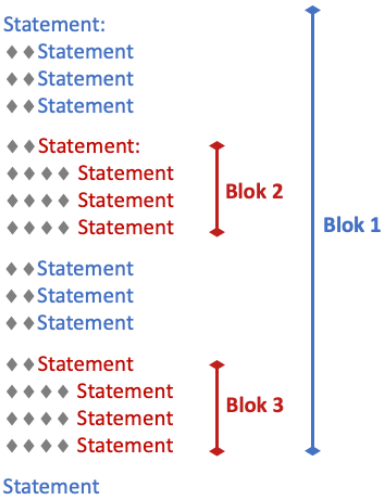


```
if BooleanExpr:
    block
else:
    block
```

Een if-statement bestaat uit het sleutelwoord if, gevolgd door een voorwaarde en een dubbelpunt. De statements na het dubbelpunt, moeten in een blok staan. Indien de voorwaarde True is, wordt alle statements in het blok uitgevoerd.

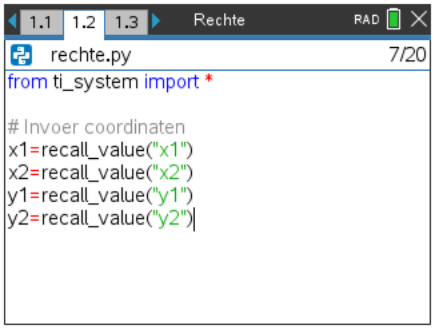
Een blok is een verzameling van gegroepeerde statements. De Whitespace voor de statements geef aan welke statements tot eenzelfde blok. Statements die op dezelfde positie starten, m.a.w. regels met dezelfde inspringingen, behoren tot hetzelfde blok.

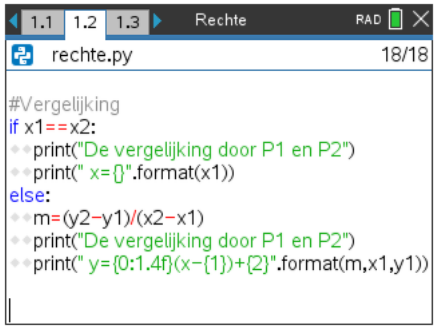
Merk op dat if-statements niet afgesloten worden met een End-statement maar begin en einde is gebaseerd op gelijke inspringingen. In TI-Technologie wordt de Whitespace aangeduid met ♦♦.



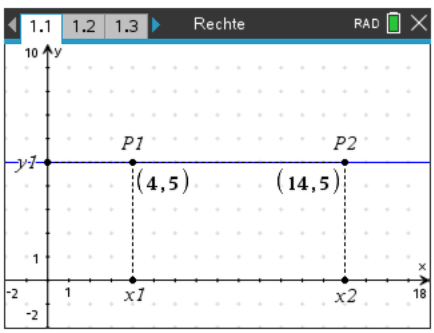
Een if-statement kan uitgebreid worden met een else-statement met eventueel een elif-statement tussenin.

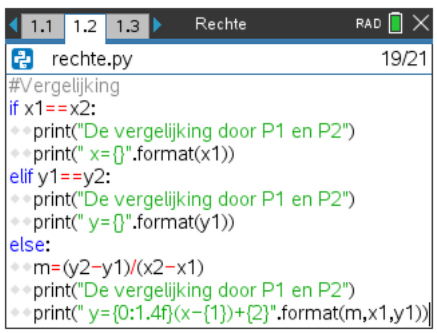
Voor ons programma geeft dit het volgende:





Met een extra elif-statement kunnen we ook de vergelijking van een horizontale rechte eleganter weergeven:

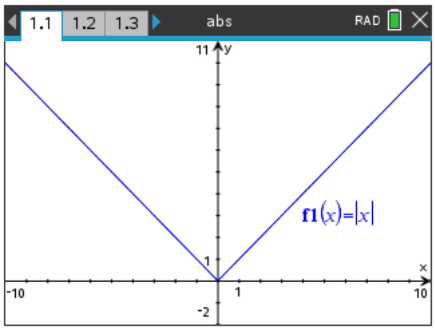


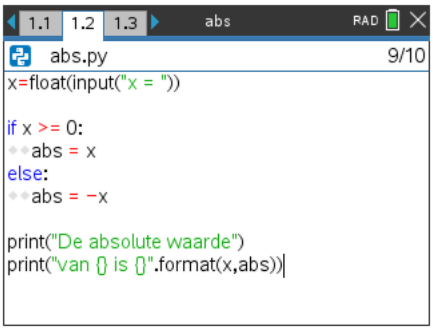


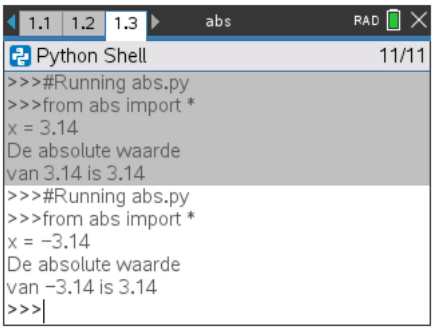
Om de structuur van Python-code goed op een rij te hebben, is het belangrijk steeds aandacht te hebben op hoe de structuur van een Python-programma gebaseerd is op inspringingen (indents of whitespace).

Nog twee voorbeelden om de structuur van de if-statements te illustreren.

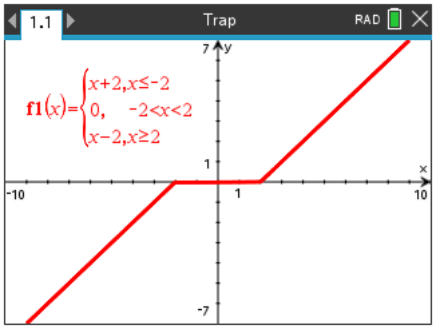
1. Bepaal de absolute waarde van een reëel getal baserend op de definitie $|x| = \begin{cases} x \ als\ x \leq 0 \\ -x\ als\ x < 0 \end{cases}$.

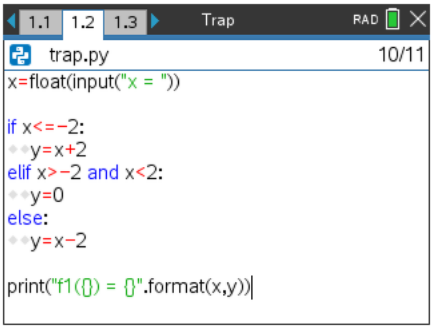


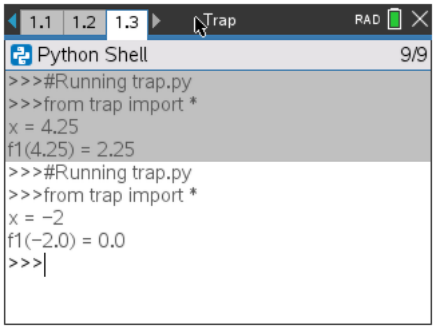




2. Bereken de functiewaarde voor de volgende stuksgewijs gedefinieerde functie $f1(x) = \begin{cases} x+2\ als\ x \leq -2 \\ 0\ als -2 < x < 2 \\ x-2\ als\ x \geq 2 \end{cases}$.



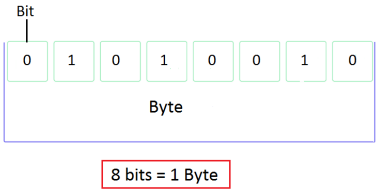




Opdracht 1
Schrijf een programma dat als output het gehele quotiënt en de rest geeft van twee gehele getallen a en b.

Opdracht 2
Schrijf een programma dat als output het maximum van twee reële getallen r en s geeft.

Opdracht 3: Binaire getallen



Bit
Ook gekend als Binary Digit. Het is de kleinste eenheid van informatie die bewaard
Kan worden op of gemanipuleerd door een computer. Een bit is ofwel 0 of 1.

Byte
Een byte is een groep van 8 bits. Oudere computers konden enkel 8 bits tegelijk bewerken.
Iedere Byte kan 1 karakter – 'A', 'k', '@', … – bewaren. $2^8 = 256$, van daar dat de uitgebreide ASCII-code tabel 256 (0-255) karakters bevat.

In Python worden binaire getallen voorgesteld met het voorvoegsel 0b gevolgd door een binair getal:

```
>>>bin=0b00001011
>>>bin
11
```

Python kent de bin()-functie om een decimaal getal om te zetten in een binair getal.

a. Ga na dat de uitvoer van deze functie van het type string is.

b. Schrijf een programma dat om een decimaal getal tussen 0 en 255 vraagt en als uitput een string geeft in de vorm zonder het voorvoegsel "0b": b.v. 27 wordt "11011".

Een 8-bits getal bestaat uit 8 nullen en/of enen.

c. Pas je programma aan zodat de uitvoer een string is die alle 8 bits geeft.
Het getal 27 wordt dan "00011011".

Opdracht 4: Schaar – Papier – Steen
Bij het spelletje *Schaar-papier-steen* kiezen twee spelers tegelijkertijd een van de drie objecten.

Kiezen beide hetzelfde, dan is de uitslag onbeslist.

In het andere geval wint schaar van papier, papier van steen en steen van schaar.

Schrijf een programma dat een speler tegen de computer laat spelen.

a. Begin met een input-opdracht in de vorm van:
```
k=int(input("0=schaar, 1=papier, 2=steen: "))
```

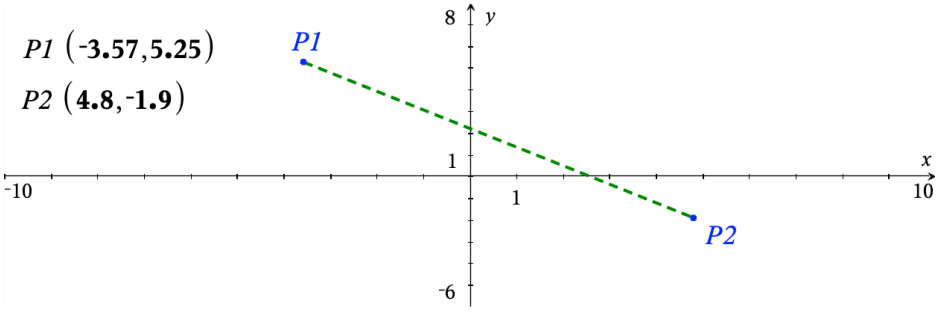
b. Laat vervolgens de computer een willekeurig getal kiezen tussen 0, 1 en 2.

c. Bepaal en druk de uitslag af.

Opdracht 5

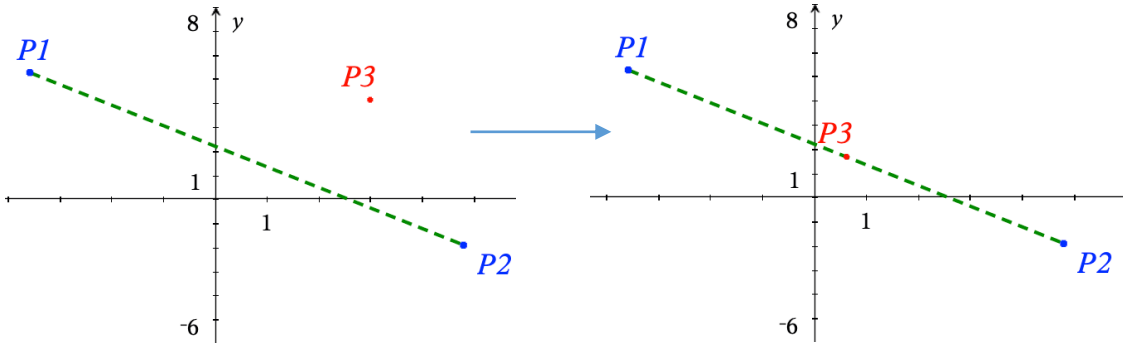
Teken twee punten $P1$ en $P2$ in de TI-Nspire CX graphs app (of gebruik het bijhorende tns-document).

Bewaar de coördinaten van de punten als volgt: $P1\ (x1, y1)$ en $P2\ (x2, y2)$.

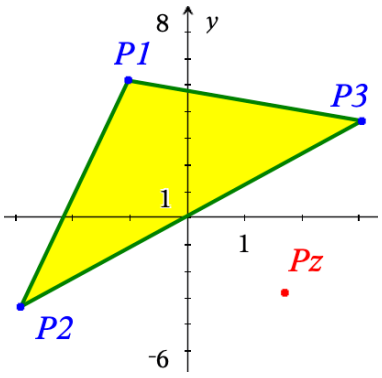


a. Schrijf een programma dat de afstand tussen de punten $P1$ en $P2$ (import coördinaten) berekent en als output de afstand geeft; tot op twee decimalen na de komma.

b. Schrijf een programma dat het midden $P3$ bepaalt tussen de punten $P1$ & $P2$ en $P3$ tekent in Graphs.
   - Teken een punt $P3$ in Graphs en bewaar de coördinaten als $P3\ (x3, y3)$.
   - Run dan het programma dat het midden berekent.



c. Schrijf een programma dat het zwaartepunt van een driehoek berekent en tekent.
   Gebruik o.a. store_value("NSvar",pyVar) van de TI Sytem module.



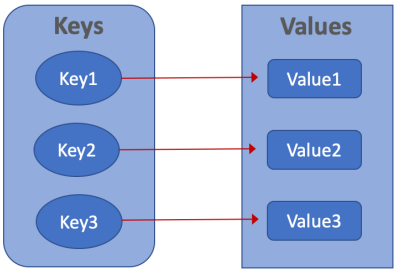
1. Dictionaries of woordenboeken

Een dictionary (data type: dict) is een verzameling van objecten die bewaard worden a.h.v. een sleutel (key). Iedere item van een dictionary bestaat uit een unieke key en een bijhorende waarde; waarbij deze waarde praktisch ieder Python object kan zijn.

Een belangrijk verschil met lijsten is dat voor dictionaries de volgorde van de items van geen belang is.

1.1. Syntax van een dictionary

Daar we elke key verbinden met een waarde creëren we een mapping-structuur bij het definiëren van een dictionary.

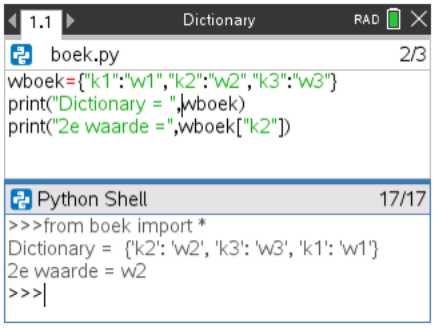


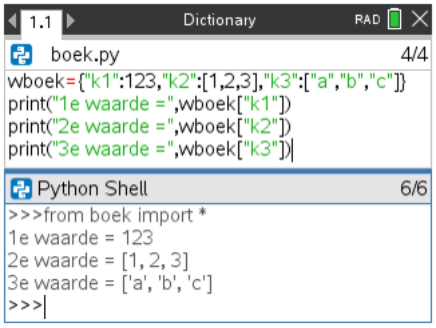
De syntax van een dictionary ziet er als volgt uit:

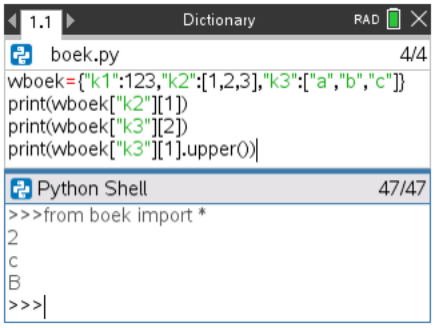
- De items van een dictionary staan tussen accolades { }
- Key en waarde worden gescheiden door een dubbele punt :
- De key staat altijd tussen aanhalingstekens “ “ (of ‘ ‘)

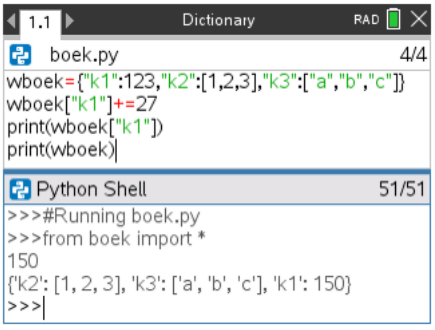
wboek={“key1”:”value1”, “key2”:”value2”, “key3”:”value3”}

We gebruiken de keys van een dictionary om een waarde aan te roepen, b.v. wboek[“key2”].



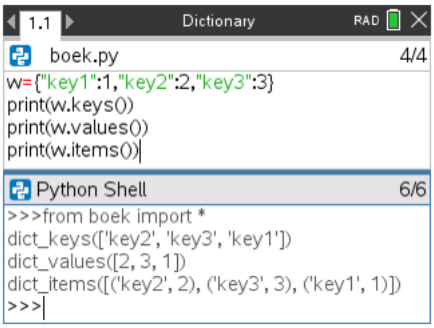


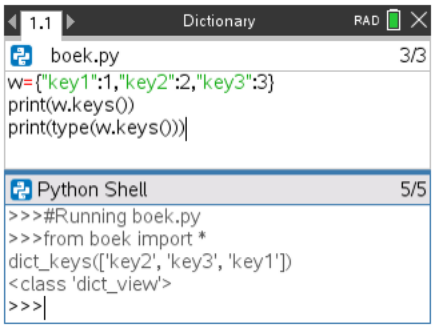




1.2. Enkele methodes voor dictionaries

Hieronder de methodes om de lijst met keys, de lijst met waarden en de lijst met items te bekijken.





1.3. Opdracht 1 – Schaar-Papier-Steen

Bij het spelletje *Schaar-papier-steen* is de uitslag onbeslist indien beide spelers dezelfde keuze maken. In het andere geval wint schaar van papier, papier van steen en steen van schaar.

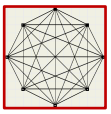
Hieronder een programma dat het spelen tegen de computer simuleert:

```
from random import *

keuzes=["schaar","papier","steen"]

k=int(input("1=schaar, 2=papier, 3=steen: "))
speler=k-1
comp=randint(0,2)

if speler==comp:
    print("Onbeslist, beide "+keuzes[speler])
else:
    if speler==0 and comp==1:
        print("jij wint met",keuzes[speler],"tegen",keuzes[comp])
    elif speler==1 and comp==2:
        print("jij wint met",keuzes[speler]+"tegen",keuzes[comp])
    elif speler==2 and comp==0:
        print("jij wint met",keuzes[speler],"tegen",keuzes[comp])
    else:
        print("jij verliest met",keuzes[speler],"tegen",keuzes[comp])
```

Pas bovenstaand programma aan om gebruikmakend van een dictionary met de spelregels voor winst, de voorwaardelijke statements wat eleganter te coderen, b.v.

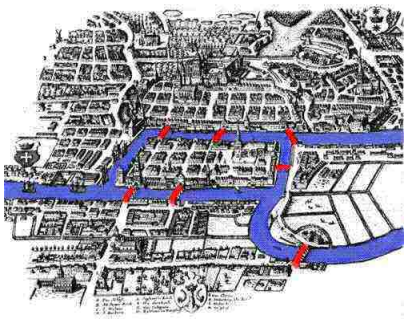
```
from random import *

keuzes=["schaar","papier","steen"]
regels={"schaar":"papier","papier":"steen",”steen":"schaar"}
.....
```

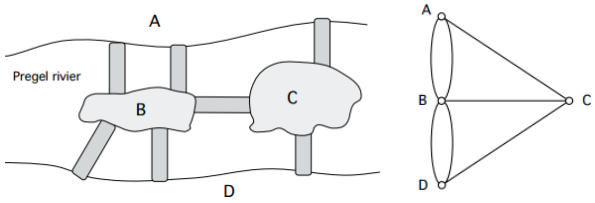
1.4. Opdracht 2 – Grafen



Een graaf is een verzameling van punten, knopen, waarvan sommige knopen verbonden zijn met lijnen, de zijden van de graaf. Grafen worden o.a. in de informatica gebruikt om het dataverkeer over netwerken weer te geven en te analyseren.

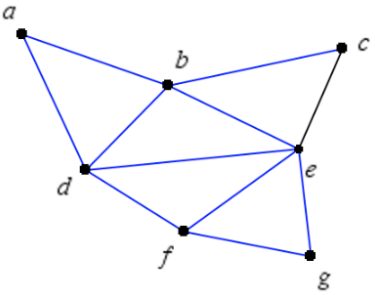
Eén van de eerste grafen-problemen is het probleem van de zeven bruggen van Koningsbergen: *Is het mogelijk een wandeling te organiseren zodat je precies één keer over iedere brug wandelt*. In 1736 loste Leonard Euler dit probleem op.





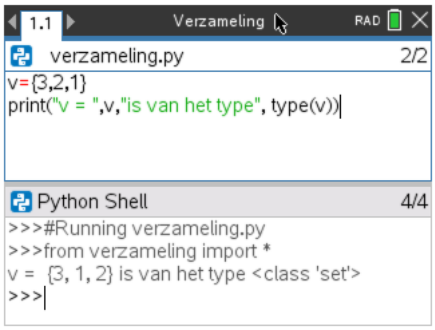
Euler bewees dat zo'n wandeling, een Euler pad, alleen mogelijk is indien de graaf geen of exact twee oneven knopen heeft. Een knoop is oneven als er een oneven aantal zijden samenkomen.

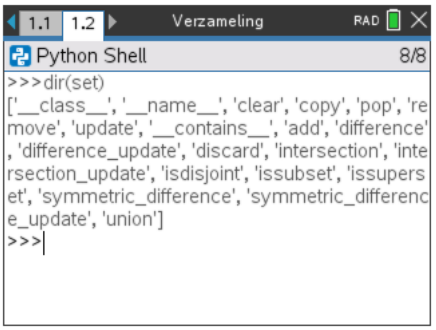
Schrijf een programma dat, gebruikmakend van een dictionary die als keys de knooppunten van de hiernaast afgebeelde graaf heeft, bij input van een knoop x als output een lijst van knopen geeft waarmee x verbonden is d.m.v. een zijde.

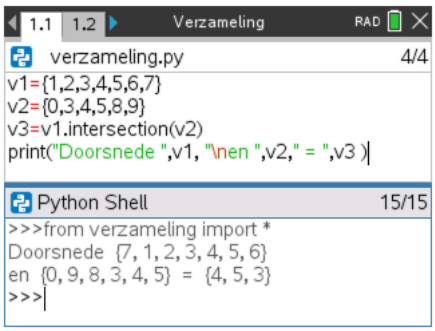


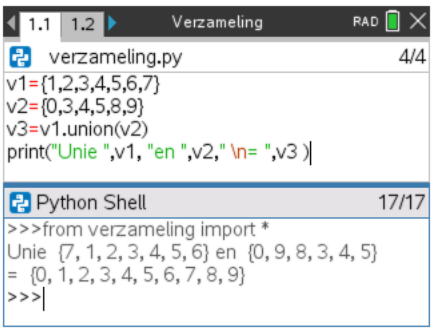
2. Sets of verzamelingen

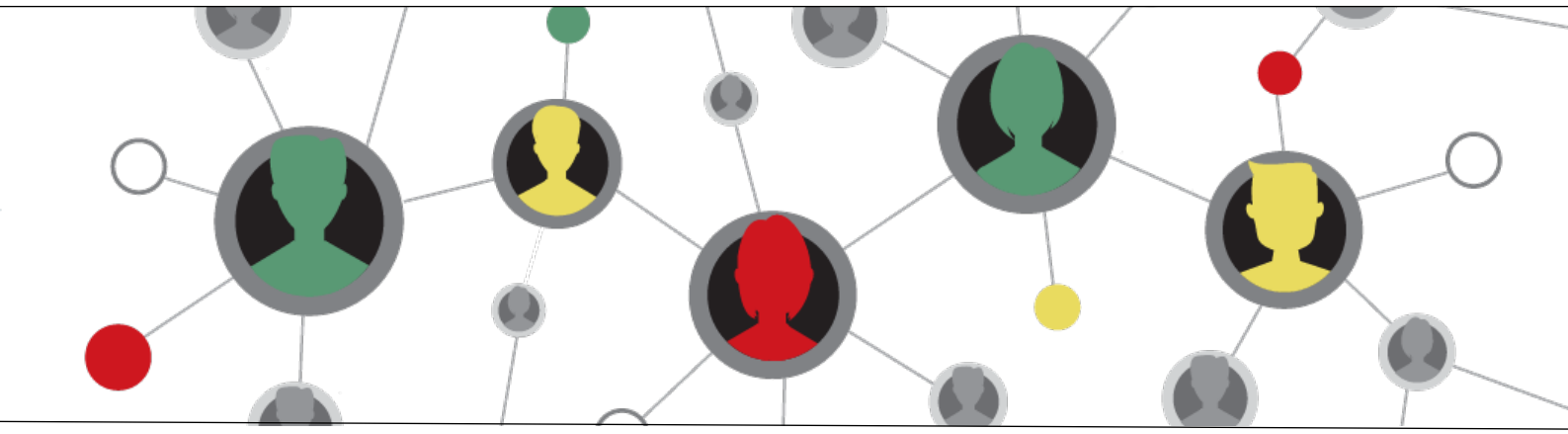
Het data type set is een ongeordende collectie van unieke elementen; m.a.w. wiskundig gezien een verzameling. We laten het aan de lezer over te experimenteren met sets en de methodes beschikbaar voor sets.











T³ NEDERLAND T³ VLAANDEREN

www.wil-depython.be
www.wil-depython.nl

Texas Instruments