



Unit 2 : Programmeren in Python

Oefenblad 2 : De for lus

In les 2 van deze unit leren we hoe je opdrachten kunt herhalen

Doelen :

- Hoe gebruik je de opdracht **for** .
- Wat is de syntax voor deze opdracht in Python

De kracht van een computerprogramma is de mogelijkheid om een serie opdrachten eventueel met een kleine wijziging te herhalen.

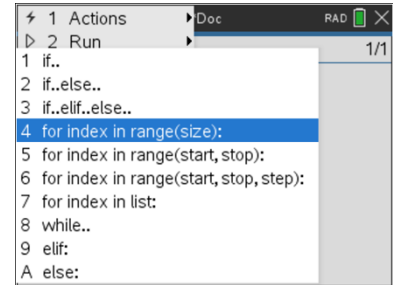
Een van de opdrachten hiervoor is de **for ... lus**

We maken een programma dat van de eerste 8 gehele getallen het kwadraat en de derde macht afdrukt.

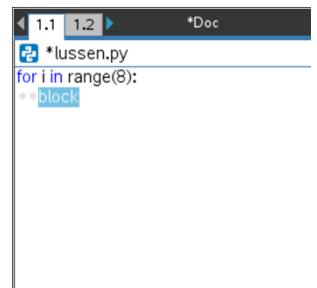
Open een nieuw Python programma en geef het een naam.

In het menu staan bij Built-ins onder Control de opties voor een **for..lus**.

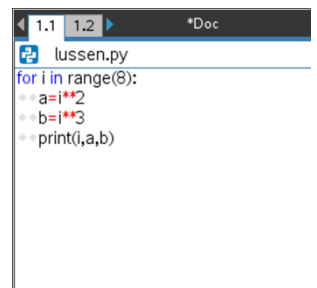
We kiezen de eerste.



De letter **i** is de zogenaamde teller. Die start als er geen beginwaarde is opgegeven altijd bij 0.



De **lus** wordt 8 keer doorlopen waarbij telkens de waarde van **i** een groter wordt.



De laatste waarde van **i** zal dus 7 zijn.

De opdrachten die in het blok staan worden nu 8 keer



herhaald.

Je ziet ook hier weer de dubbele punt aan het eind van de regel waar het blok begint.

We willen de kwadraten en derde machten afdrukken dus die gaan we binnen de lus berekenen.

In Python noteer je machtsverheffen met twee keertekens. Dus 2^5 wordt in Python geschreven als: `2**5`
In de for-lus moeten we telkens `i**2` en `i**3` berekenen en afdrukken. Voeg deze regels toe en bekijk het resultaat



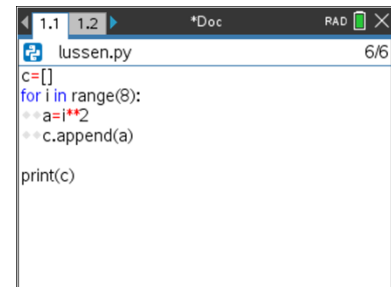
met Ctrl+R.

We kunnen de resultaten ook opslaan in een lijst.

Begin het programma dan met een lege lijst (Een lege lijst is een lijst zonder elementen)

Elke keer als de lus wordt doorlopen voegen we een nieuwe uitkomst toe aan de lijst met **.append**. (In Python is dit een manier om een lijst uit te breiden.)

Pas het programma aan zodat de lijst met kwadraten wordt gemaakt en druk deze uiteindelijk af.



```
1.1 1.2 *Doc RAD 6/6
lussen.py
c=[]
for i in range(8):
  a=i**2
  c.append(a)

print(c)
```

De printopdracht staat nu buiten de lus. Kijk ook eens wat er gebeurt als je deze printopdracht binnen de lus zet! (dus de laatste regel laat beginnen met twee spaties.)

In de wiskunde is 5! (5 faculteit) de uitkomst van $5*4*3*2*1 = 120$.

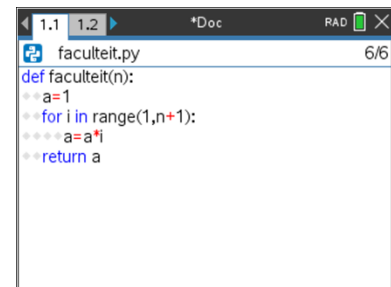
Maak een programma waarin de faculteit-functie wordt gedefinieerd.

Aanwijzing:

Start met het getal 1 en maak een for-lus waarbij elke keer de nieuwe uitkomst wordt vermenigvuldigd met de lus-variabele (i).

Ga na dat je hiermee de faculteit van grote getallen kunt berekenen.

(Als je het programma uitvoert kom je in de shell waar dan de functie is gedefinieerd. Met de **var** toets of met Ctrl+L kun je hem oproepen)



```
1.1 1.2 *Doc RAD 6/6
faculteit.py
def faculteit(n):
  a=1
  for i in range(1,n+1):
    a=a*i
  return a
```