



Unit 2 : Beginselen van programmeren in Python

Oefenblad 3 : De while lus

In deze les leren we een andere herhalingsopdracht.

Doelen :

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

Als je een serie opdrachten een aantal keren wilt herhalen, maar je weet van tevoren niet hoe vaak, dan kun je de **while...** opdracht gebruiken.

Dit kun je lezen als:

Zolang een bepaalde voorwaarde geldt blijf dan herhalen



We maken een programma voor het volgende probleem:

Gooi met een dobbelsteen en herhaal dit net zolang totdat er 6 is gegooid.

Je weet dus van tevoren niet hoe vaak je moet gooien.

Open een nieuw Python programma en geef het een naam.

Omdat we randomgetallen gaan gebruiken hebben we de module nodig waarin die opdrachten staan.

Die moeten we als eerste invoegen.

Kies F1(Fns...) Modul 2:random. (Dit kan ook met: » en dan 2:random)



```
EDITOR: ZES
PROGRAM LINE 0002
from random import *
```

We gaan in het programma twee variabelen gebruiken.

De variabele t die bijhoudt hoe vaak er is gegooid en de variabele d die bijhoudt wat er is gegooid. Beide variabelen geven we als beginwaarde 0.

Voeg het sjabloon voor **while ...** in met F1(Fns...) en dan **Ctl** optie 8.

Achter while komt de voorwaarde te staan.

Dat is in dit geval: d is niet gelijk aan 6

In Python noteer je dat als: d!=6. (je kunt dit vinden met y »)

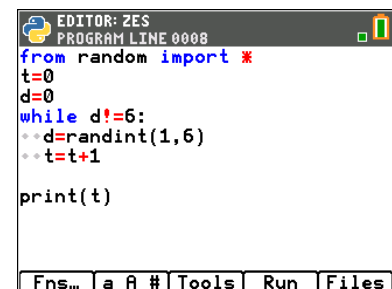


```
EDITOR: ZES
PROGRAM LINE 0005
from random import *
t=0
d=0
while d!=6:
**
```

In het herhalingsblok komen twee opdrachten te staan. De eerste is het kiezen van een randomgetal tussen 1 en 6 en de tweede is de teller (t) 1 verhogen.

Als de uitslag gelijk is aan 6 wordt de lus verlaten en volgt nog de opdracht om het aantal worpen af te drukken.

Als je het programma een aantal keren uitvoert (Run) dan zie je waarschijnlijk verschillende waarden voor t.



```
EDITOR: ZES
PROGRAM LINE 0008
from random import *
t=0
d=0
while d!=6:
**d=randint(1,6)
**t=t+1
print(t)
```

Tip voor de docent: Als je een programma hebt uitgevoerd en je wilt vanuit de shell hetzelfde programma nog een keer uitvoeren dan kan dat met F3(Tools) en dan optie 1: Rerun Last Program.



We gaan het programma wijzigen zodat we het experiment niet één keer maar honderd keer uitvoeren en dan het gemiddeld aantal worpen berekenen.

Verander eerst het vorige programma zodat het werpen met de dobbelsteen totdat zes is gegooid in een functie wordt gezet (noem die functie bijvoorbeeld `dobbelen()`).

Als je de functie aanroept geeft die als uitkomst het aantal worpen.

Als we deze functie 100 keer aanroepen en we tellen die uitkomsten bij elkaar op waarna we die som delen door 100, dan krijgen we het gemiddelde.

We voegen nu een **for**-lus toe die dit 100 keer doet.

Maar eerst maken we een nieuwe variabele voor de som en starten met de waarde 0.

In het programma hiernaast zie je dat de **for**-lus 100 keer doorlopen wordt en dat elke keer de som wordt opgehoogd met het nieuwe aantal worpen.

Uiteindelijk wordt de som gedeeld door 100 voor het berekenen van het gemiddelde.

```
EDITOR: ZES1
PROGRAM LINE 0009
from random import *

def dobbelen():
    t=0
    d=0
    while d!=6:
        d=randint(1,6)
        t=t+1
    return t
```

```
EDITOR: ZES1
PROGRAM LINE 0016
while d!=6:
    d=randint(1,6)
    t=t+1
return t

som=0
for i in range(100):
    a=dobbelen()
    som=som+a

print(som/100)
```

Voer het programma uit met Run.

Om het gemiddelde nauwkeuriger te bepalen kun je bijvoorbeeld het getal 100 in het programma wijzigen in 1000 (op twee plekken).

Je berekent dan het gemiddelde van 1000 experimenten.

```
PYTHON SHELL

>>> # Shell Reinitialized
>>> # Running ZES1
>>> from ZES1 import *
6.23
>>> |
```