PROJECT 6

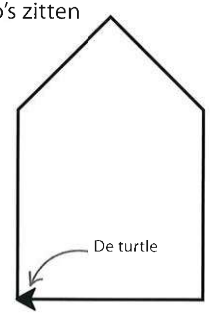
Tekenmachine

ZIE OOK
‹ 122–123 Loops in Python
Bibliotheken 152–153 ›

Het is tijd voor een wat ingewikkelder project. Dit programma, de tekenmachine, verandert een string met eenvoudige instructies in turtle-commando's om verschillende vormen te tekenen. De vaardigheden die je in dit programma gebruikt, zijn de basis van elke codeerder.

Kies een testvorm

Voor het schrijven van een programma dat elke vorm kan tekenen, is het handig eerst een vorm te kiezen om mee te beginnen. Met deze huis-vorm kun je het programma op elk niveau testen. Aan het eind kun je het huis tekenen met veel minder codering – door een enkele string te gebruiken waarin meerdere korte tekencommando's zitten (bijvoorbeeld 'F100').



▷ **Turtle tekent een huis**
De pijl geeft de uiteindelijke en huidige positie van de turtle. Hij is linksonder begonnen en beweegt met de klok mee rond het huis.

```
from turtle import *
reset()
left(90)
forward(100)
right(45)
forward(70)
right(90)
forward(70)
right(45)
forward(100)
right(90)
forward(100)
```

Laadt alle commando's die de turtle besturen

Reset de positie van de turtle en zet de pen klaar voor gebruik

Beweegt de turtle voorwaarts met 70

Laat de turtle 90 graden naar rechts draaien

△ **Programma om een huis te tekenen**
Deze code vertelt de turtle om een huis te tekenen. Er zijn veel coderingsregels nodig voor wat in feite een vrij eenvoudig programma is.

Drie delen van het programma

De tekenmachine zal een groot programma zijn. Om het te ordenen, kun je het in drie delen verdelen met elk een verschillende taak.

Functie 1

△ **Turtle-besturing**
Deze functie volgt een eenvoudig commando van de gebruiker en verandert dit in een turtle-commando. Het gebruikerscommando bestaat uit een enkele letter en een getal.

Functie 2

△ **String artist**
In dit programma voert de gebruiker een string van instructies in. Deze functie splitst de string op in kleinere eenheden, die worden doorgegeven aan de turtle-besturing.

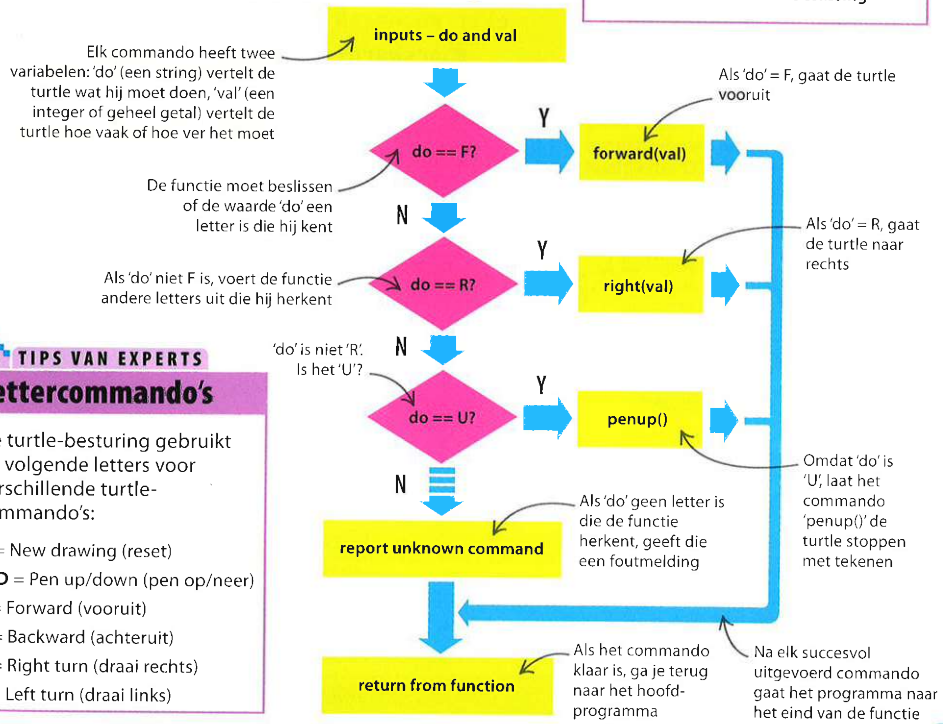
Hoofdprogramma

△ **Gebruikersinterface**
De string artist moet zijn input ergens vandaan krijgen. Met de gebruikers-interface kan de gebruiker iets intypen in een commandostring waar de string artist mee kan werken.

Een stroomschema tekenen

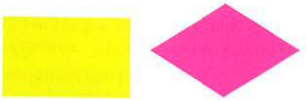
Codeerders plannen programma's vaak op papier om zo een betere codering te krijgen met minder fouten. Dit kan door een stroomschema te tekenen, een diagram met stappen en beslissingen dat het programma moet volgen.

1 Dit stroomschema toont het plan voor de turtle-besturingsfunctie. Het neemt een letter (input 'do') en een getal (input 'val') en verandert die in een turtle-commando. 'F' bijvoorbeeld en '100' worden veranderd in het commando 'forward(100)'. Als de functie de letter niet herkent, geeft het een foutmelding.



TIPS VAN EXPERTS
Rechthoeken en ruiten

Stroomschema's bestaan uit rechthoeken en ruiten. In de rechthoeken staan acties die het programma uitvoert, de ruiten zijn punten waarin de beslissing wordt genomen.



TIPS VAN EXPERTS
Lettercommando's

De turtle-besturing gebruikt de volgende letters voor verschillende turtle-commando's:

N = New drawing (reset)
U/D = Pen up/down (pen op/neer)
F = Forward (vooruit)
B = Backward (achteruit)
R = Right turn (draai rechts)
L = Left turn (draai links)

TEKENMACHINE

De turtle-besturing

Het eerste deel van het programma is een functie die de turtle laat bewegen, één commando per keer. Het wordt in het stroomschema op de vorige pagina gepland. Met deze codering kan de schildpad de waarden 'do' en 'val' converteren in bewegingscommando's.

2 De codering maakt de turtle-besturingsfunctie aan. Hij zet de 'do'-inputs om in richtingen voor de schildpad, en 'val'-inputs in hoeken en afstanden.


```
from turtle import *
def turtle_controller(do, val):
    do = do.upper()
    if do == 'F':
        forward(val)
    elif do == 'B':
        backward(val)
    elif do == 'R':
        right(val)
    elif do == 'L':
        left(val)
    elif do == 'U':
        penup()
    elif do == 'D':
        pendown()
    elif do == 'N':
        reset()
    else:
        print('Unrecognized command')
```

Laadt alle commando's die de schildpad besturen

Definieert 'do' en 'val' als inputs vooor de functie

Dit commando converteert alle letters in 'do' naar hoofdletters

Dit vertelt de functie een 'do'-waarde van F te veranderen in het turtle-commando 'forward'

Net als in het stroomschema vergelijkt de functie de 'do'-letter met alle letters die hij begrijpt

Dit commando instrueert de schildpad te stoppen met tekenen op de pagina

Dit commando reset de positie van de turtle naar het midden van het scherm

Dit bericht verschijnt als de waarde 'do' een letter is die de functie niet herkent

3 Hier volgen een paar voorbeelden van hoe je de turtle-besturing kunt gebruiken. Elke keer dat je dat doet, neemt het een 'do, val'-commando en verandert het in een code die de schildpad kan begrijpen.

Dit roept de functie op via zijn naam

De inputs 'do' en 'val' vertellen de schildpad 100 stappen vooruit te doen

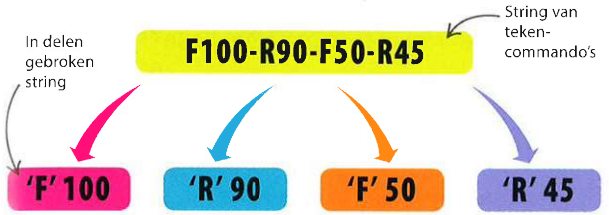
```
>>> turtle_controller('F', 100)
>>> turtle_controller('R', 90)
>>> turtle_controller('F', 50)
```

Hierdoor draait de schildpad 90 graden naar rechts

Schrijf een pseudocode

Een andere manier om een programma te ontwerpen is door het in pseudocode te schrijven. 'Pseudo' betekent vals, dus pseudocode is geen echte code die je kunt inzetten. Het is ruwe codering waarmee je je ideeën kunt schrijven alsof het echt is.

4 Het is tijd om een string artist te ontwerpen. Deze functie neemt een string van meerdere 'do'- en 'val'-inputs en breekt hem in paren van een letter en een cijfer. Vervolgens stuurt het de paren stuk voor stuk door naar de turtle-besturing.



5 Dit is de string artist geschreven in pseudocode. Hiermee kun je de ideeën en structuur van de codering organiseren zonder al over de details te hoeven nadenken.

```
function string_artist(input – the program as a string):
    split program string into list of commands
    for each command in list:
        check it's not blank
            – if it is go on to next item in list
        command type is the first letter
        if followed by more characters
            – turn them into a number
        call turtle_controller(command type, number)
```

De functie zal input door de gebruiker inbrengen in een string commando's (bijvoorbeeld 'F100-R90')

Splitst string in een lijst afzonderlijke commando's

Een leeg commando zal niet werken, dus de functie slaat het over

Herkent de eerste letter als een 'do'-commando

Herkent de volgende karakters als een 'val'-getal

Stuurt het eenvoudige commando door naar de turtle-besturing

TIPS VAN EXPERTS

Duidelijk coderen

Niet alleen computers moeten je codering kunnen lezen, mensen ook. Het is daarom belangrijk dat je codes gemakkelijk te begrijpen zijn.

Gebruik functies om je codering in kleinere stukken te breken. Elke functie mag slechts één taak vervullen in het programma.

Geef namen aan je functies en variabelen die zeggen wat ze doen: 'leeftijd_in_jaren' is duidelijker dan 'lij'.

Gebruik veel commentaren (met het symbool '#) om toe te lichten wat er gebeurt. Dat maakt het makkelijker om de code nog eens terug te lezen.

Gebruik geen symbolen die verward kunnen worden met andere: een hoofdletter 'O' lijkt op een nul, een kleine letter 'L' op een hoofdletter 'i' of een '1'.

TEKENMACHINE

De string artist maken

De pseudocode op de vorige pagina ontwerpt een functie, de zogenaamde string artist, die een string van waarden verandert in afzonderlijke commando's die naar de turtle-besturing worden gestuurd. De volgende stap is de pseudocode in een echte Pythoncode te veranderen met behulp van de functie 'split()'.

6 De functie 'split()' splitst een string op in een lijst met kleinere strings. Elk breekpunt wordt gemarkeerd met een speciaal karakter (in dit programma '-').

Deze string maakt een lijst met de commando's om de proef-huisvorm te maken

```
>>> program = 'N-L90-F100-R45-F70-R90-F70-R45-F100-R90-F100'
>>> cmd_list = program.split('-')
>>> cmd_list
['N', 'L90', 'F100', 'R45', 'F70', 'R90', 'F70', 'R45', 'F100', 'R90', 'F100']
```

De functie 'split()' breekt de string op in een lijst met afzonderlijke commando's

7 Schrijf nu met behulp van de echte Pythoncode de pseudocode uit voor de string artist. Gebruik de functie 'split()' om de input-string in kleine turtle-commando's op te delen.

```
def string_artist(program):
    cmd_list = program.split('-')
    for command in cmd_list:
        cmd_len = len(command)
        if cmd_len == 0:
            continue
        cmd_type = command[0]
        num = 0
        if cmd_len > 1:
            num_string = command[1:]
            num = int(num_string)
        print(command, ':', cmd_type, num)
        turtle_controller(cmd_type, num)
```

Vertelt het programma de string op te splitsen zodra het het karakter 'a' tegenkomt

Hierdoor doorloopt het programma de lijst met strings – elk item is een commando voor de turtle

Als de lengte van het commando 0 is (het is dan dus leeg), slaat de functie het over en gaat door naar het volgende

Krijgt de lengte van de commando-string

Neemt het eerste karakter van het commando (onthoud dat strings beginnen bij 0) en stelt het in als het commandotype ('F', 'U' enz.)

Checkt of het commando wordt gevolgd door meer karakters (het getal)

Dit neemt alle resterende karakters van het commando door de eerste eraf te halen

Converteert de karakters van strings in getallen

Print het commando op het scherm zodat je kunt zien wat de code doet

Geeft het commando door aan de schildpad

8 Als de string die de instructies voor de vorm van het huis bevat doorgegeven wordt aan de string artist, verschijnt deze output in het Shell-venster.

```
>>> string_artist('N-L90-F100-R45-F70-R90-F70-R45-F100-R90-F100')
N : N 0
L90 : L 90
F100 : F 100
R45 : R 45
F70 : F 70
R90 : R 90
F70 : F 70
R45 : R 45
F100 : F 100
R90 : R 90
F100 : F 100
```

Reset het scherm en plaatst de schildpad weer in het midden

De turtle-commando's worden gescheiden door een '-'

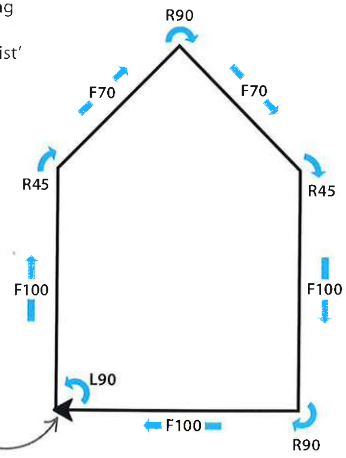
Voor het commando 'F100' is het commandotype 'F' en 'num' is '100'

Hierdoor draait de schildpad 45 graden voordat hij het dak tekent

Dit commando laat de schildpad de rechterzijde van het dak tekenen

De schildpad draait 90 graden naar rechts en is klaar om de onderkant van het huis te tekenen

9 Elk commando in de string dat wordt doorgegeven aan de functie 'string_artist' is losgemaakt, geïdentificeerd en uitgevoerd. In het tekenvenster van de schildpad wordt een afbeelding van het huis geplaatst.



Het programma laat de schildpad een huis tekenen

ONTHOUD

Commando's

Hier volgt een overzicht van de turtle-commando's in dit programma. Sommige zijn slechts een letter lang, andere bevatten een getal dat de schildpad vertelt waar hij moet gaan of draaien. Elke keer als je 'string_artist' activeert, voegt dat iets toe aan de tekening, totdat 'N' het scherm schoonveegt.

N = New drawing (nieuwe tekening)
U/D = Pen Up/Down (pen op/neer)
F100 = Forward 100 (vooruit 100)
B50 = Backwards 50 (terug 50)
R90 = Right turn 90 deg (rechts 90 gr)
L45 = Left turn 45 deg (links 45 gr)

TEKENMACHINE

De codering afronden met een gebruikersinterface

De tekenmachine heeft een interface nodig om hem makkelijker te kunnen gebruiken. Hiermee kan de gebruiker een string invoeren vanaf het toetsenbord om de machine te vertellen wat hij moet tekenen.

10 Deze code maakt een pop-upvenster aan waarin de gebruiker instructies kan zetten. Door een 'while True'-loop blijft hij nieuwe strings invoeren.

De drie aanhalingstekens (''') vertellen Python dat alles tot de volgende drie tekens deel uitmaakt van dezelfde string, inclusief de regelafbrekingen

```
instructions = '''Enter a program for the turtle:
eg F100-R45-U-F100-L45-D-F100-R90-B50
N = New drawing
U/D = Pen Up/Down
F100 = Forward 100
B50 = Backwards 50
R90 = Right turn 90 deg
L45 = Left turn 45 deg'''
screen = getscreen()
while True:
    t_program = screen.textinput('Drawing Machine', instructions)
    print(t_program)
    if t_program == None or t_program.upper() == 'END':
        break
    string_artist(t_program)
```

Vertelt je welke letters je moet gebruiken voor verschillende turtle-commando's

Einde van de string

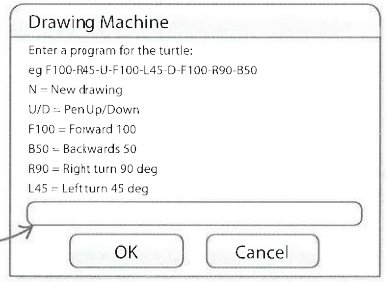
Krijgt de data die nodig zijn om het pop-upvenster te maken

Deze regel vertelt het programma wat er in het pop-upvenster moet verschijnen

Stopt programma als je 'END' typt of op de knop 'Cancel' klikt

Geeft de string door aan de functie string artist

11 Dit venster verschijnt in het turtle-venster, klaar om een programmastring in te typen voor de tekenmachine.



Typ hier de programmastring en klik op 'OK' om het programma te starten

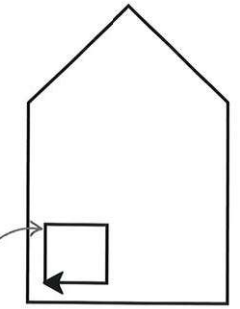
△ **Turtle-besturing**
Met dit programma kun je de schildpad makkelijker besturen en hoef je het programma niet opnieuw op te starten om een andere tekening te maken.

12 Met de tekenmachine kun je meer maken dan alleen omtrekken. Als je de pen omhoog tilt als je naar een nieuwe positie gaat, kun je de details in de vorm opvullen. Start het programma en probeer onderstaande string in te voeren.

```
N-L90-F100-R45-F70-R90-F70-R45-F100-R90-F100-
B10-U-R90-F10-D-F30-R90-F30-R90-F30-R90-F30
```

Neemt de pen van de turtle op, zodat hij beweegt zonder een lijn te trekken

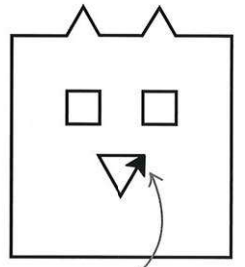
Zet de pen neer om een venster te tekenen



Tijd voor iets anders

Nu je weet hoe je details kunt toevoegen, heb je pas echt wat aan je tekenmachine. Teken deze uilenkop met de instructiestring hieronder.

```
N-F100-L90-F200-L90-F50-R60-F30-L120-F30-R60-F40-
R60-F30-L120-F30-R60-F50-L90-F200-L90-F100-L90-U-
F150-L90-F20-D-F30-L90-F30-L90-F30-L90-F30-R90-U-
F40-D-F30-R90-F30-R90-F30-R90-F30-L180-U-F60-R90-
D-F40-L120-F40-L120-F40
```



De string tilt de pen drie keer omhoog om de ogen en de snavel apart te tekenen

De pijl geeft aan waar de turtle is gestopt. Hiermee zie je dat de snavel van de uil het laatst is getekend

ONTHOUD

Wat heb je bereikt?

Je hebt een tekenmachineprogramma gemaakt via enkele kleinere stappen:

Een stroomschema gebruikt om een functie te ontwerpen door de beslissingspunten en de bijbehorende acties uit te werken.

Een pseudocode geschreven om een functie te ontwerpen alvorens de echte code te schrijven.

De functie 'turtle_controller' gemaakt, die bepaalt welke functie moet worden uitgevoerd via de letter en het getal die je eraan hebt toegekend.

De functie 'string_artist' gemaakt, die een schildpad heeft voortgebracht die tekent vanuit een string met instructies.

Een interface gemaakt waarmee je het programma met het toetsenbord kunt vertellen wat het moet tekenen.