**Programmeren in Python – Basis**

Welkom bij je eerste programmeerlesje! We gaan vandaag aan de slag met de programmeertaal Python. Het is een hele makkelijke programmeertaal, maar wordt ook in de wetenschap nog steeds gebruikt. Vandaag programmeren we een schildpad.

**1 Je eerste programma**

Het startprogramma staat al voor je klaar. Je moet het alleen nog even opslaan onder een naam die je zelf snel herkent. Je mag ook meteen de naam van de schildpad veranderen als je dat wilt. Overal waar ikke staat mag je de naam van je schildpad neerzetten.

import turtle

ikke = turtle.Turtle()

ikke.shape("turtle")

wn = turtle.Screen()

wn.bgcolor("white")

ikke.forward(50)

ikke.right(90)

ikke.forward(50)

ikke.right(90)

1 Druk op F5 en kijk wat je schildpad doet. Begrijp je alle regels code die er staan? Probeer elke regel uit te leggen.

Je schildpad is begonnen met het lopen/maken van een vierkant.

2 Maak het vierkant voor hem af.

3 Laat je schildpad nu een mooie driehoek lopen.

Bij het maken van een vierkant voer je steeds hetzelfde stukje uit. Je kunt dit ook sneller programmeren. Je doet dat met een for-loop. Je geeft in een for-loop aan hoe vaak die dat stukje moet herhalen. Python voert het stukje code onder de for-loop met vier spaties er tussen uit.

for i in range(4):

ikke.forward(50)

ikke.right(90)

4 Probeer het vierkant met de for-loop te maken.

5 Maak met behulp van een for-loop een cirkel.

Je kunt je schildpad ook een kleur geven en tijdens het programma van kleur laten veranderen. ikke.color("blue") maakt je schildpad bijvoorbeeld blauw.

6 Zorg er voor dat het looppad van de schildpad mooie kleurtjes krijgt.

Ook de dikte van het gelopen pad kun je veranderen. ikke.pensize(3) is een dikte van 3 pixels.

7 Maak je pad steeds dikker.

**2 Een schildpadden race**

import turtle

from random import randint

ikke = turtle.Turtle()

ikke.shape("turtle")

ikke.color("blue")

ikke.pensize(3)

ikke.setpos(0,10)

turtle.speed(0)

wn = turtle.Screen()

wn.bgcolor("white")

while True:

ikke.forward(randint(0,3))

We gaan twee schildpadden tegen elkaar laten racen.

1 Maak daarvoor eerst een tweede schildpad aan.

In het script maken we gebruik van twee modules: turtle en randint. Een module voegt extra mogelijkheden aan je programma toe, zoals in dit geval de functie randint. Ook staat eronder ‘while True:’. Dit is een oneindige loop. Zo lang waar ‘waar’ is, voert hij het stukje uit.

2 Laat de twee schildpadden tegen elkaar racen door ook voor je tweede schildpad een forward functie te maken.

De randomgenerator is niet echt random. Elke keer als je het programma start, zal dezelfde volgorde gegeven worden. Om dit te veranderen kun je gebruik maken van seed() Deze kijkt naar de tijd en gebruikt dat getal om een nieuwe serie getallen (random dus) te produceren. Hiervoor heb je naast randint ook de library seed nodig.

3 Breid je script uit met een echte random generator.

**3 Nog veel meer schildpad**

Je kunt nog veel meer tekenen en programmeren. Probeer het programma hiernaast eens. Je kunt alle getallen een voor een veranderen en kijken wat het doet.

import turtle

wn = turtle.Screen()

wn.bgcolor("lightgreen")

tess = turtle.Turtle()

tess.shape("turtle")

tess.color("blue")

tess.penup()

size = 5

for i in range(35):

tess.stamp()

size = size + 3

tess.forward(size)

tess.right(24)

wn.mainloop()

**4 Een stukje wiskunde**

Computerprogramma’s waarin je kunt programmeren worden vaak gebruikt om berekeningen uit te voeren. Je kunt het dan zo moeilijk maken als je wilt. Het standaardprogramma wat je als eerste gebruikt is echter hello world.

1 Open een nieuwe file en type: print ("Hello, World!");

2 Sla het bestand op en run (f5).

Voer nu je eerste berekening uit, bijvoorbeeld:

3 print(3 + 5);

a = 3

b = 5

print(a+b)

print(type(a))

Omdat je niet aangeeft wat de shell precies moet printen, zijn er geen aanhalingstekens. Het mag ook anders, namelijk met variabelen.

4 Neem het stukje code hiernaast over en kijk wat er gebeurt.

Je hebt verschillende soorten type variabelen. Als je een variabele aanmaakt, wordt er een klein stukje geheugen gereserveerd voor deze variabele. Het ene type heeft meer ruimte nodig dan het andere. In sommige programmeertalen moet je van te voren aangeven wat voor een type variabele je hebt.

int een geheel getal

long een groot geheel getal

float een decimaal getal

string een woord

char een karakter (teken)

5 Probeer verschillende sommetjes uit. Kijk ook eens naar het type variabele dat geprint wordt (type(a+b)).

We gaan nu naar echte wiskunde toe. We gaan een Fibonaccireeks maken. Bij een Fibonaccireeks tel je twee getallen bij elkaar op. Dit levert een nieuw getal op. Daarbij tel je weer de vorige waarde op. Je krijgt dan bijvoorbeeld een serie: 1 1 2 3 5 8 13…. Let op! Het printen gaat heel snel en de getallen worden snel groter. Afsluiten van het programma gaat met de sneltoets ctrl + c.

6 Neem het stukje code over en run. Probeer verschillende startwaarden voor a en b.

Processen die te snel gaan kun je vertragen. Daarvoor heb je de library time nodig. In deze library zitten weer stukjes code die je kunt gebruiken.

a = 1

b = 1

while True:

c = a + b

print(c)

a = b

b = c

7 Importeer de library time door bovenaan je code te zetten: import time. De code voor een seconde wachten met uitvoeren van een volgende coderegel is: time.sleep(1). Laat je code opnieuw runnen, maar dan met enige vertraging.

Een van de belangrijkste stukjes code in de programmeertaal is het if-statement. Als er aan een gestelde voorwaarde wordt voldaan, dan moet er een specifiek stukje code worden uitgevoerd. Bijvoorbeeld: als ik 1 miljoen win (geef aan met == in de code) , dan geef ik €100.000,- weg aan goede doelen. In programmeertaal moet er dus gekeken worden of er aan die voorwaarde wordt voldaan en als dat zo is, dan moet er een specifieke actie volgen. Elke programmeertaal kent deze voorwaarde, de manier waarop die voorwaarde precies wordt opgeschreven (syntax) verschilt per programmeertaal.

if c == 8:

a = 1

b = 1

8 Zet in je while loop de code hier rechts. Run het programma. Leg aan de hand van de code uit wat er gebeurt.

**5 Input**

Je hebt net geleerd hoe je dingen kunt printen en hoe je een wiskundige bewerking kunt uitvoeren met getallen die van te voeren bekend zijn. Je kunt ook vragen om een input van de gebruiker. Dan lijkt het al veel meer op een echte rekenmachine.

getal1 = input("geef een getal: ")

getal2 = input("geef nog een getal: ")

print("de som is: ", int(getal1) + int(getal2))

print("het product is: ", int(getal1)\*int(getal2))

1 Neem het script over en test het.

Hetgeen je intypt wordt gezien als een woord (string). Deze moet geconverteerd worden naar een getal (integer). Pas daarna kun je wiskundige bewerkingen uitvoeren. Met het if-statement kun je ook de gebruiker vragen om welke bewerking (optellen, aftrekken, vermenigvuldigen etc) er uitgevoerd moet worden met de twee getallen. Hiernaast staan enkele wiskundige bewerkingen.

+ optellen

- aftrekken

\* vermenigvuldigen

/ delen

% modulo

\*\* exponent

// delen (geheel getal)

2 Probeer 9//4 te printen. Welke waarde krijg je dan? En 17//4? En 17%4?

3 Breid het script uit zodat je een eenvoudig rekenmachine maakt.

**6 Counter**

In een stukje code is het vaak heel handig om te zien hoe vaak een event heeft plaatsgevonden. Denk aan het aantal keren dat een website is bezocht of het aantal keren dat er op een knopje is gedrukt. Dit wordt een counter genoemd.

import time

i = 0

while True:

print (i, "sec")

time.sleep(1)

i= i + 1

1 Open een nieuwe file en neem het stukje code over.

2 Leg uit dat je nu een soort klok hebt gemaakt.

3 Hoe zou je de tijd laten beginnen op 2400 s?

4 Verander je code zodat alleen elke 5e seconde de tijd geprint wordt. (tip: Je kunt werken met modulo. i%5 deelt eerst door 5 en berekend vervolgens hoeveel er dan nog over is: 7%5 = 2 en 28%5 = 3)

Je kunt meerdere voorwaarden tegelijk hebben en bekijken. In python gebeurt dit met or. Bijvoorbeeld: if i%5 == 0 or i%7 == 0: print(i)

Vergelijken (if)

== links gelijk aan rechts

!= links ongelijk aan rechts

> links groter dan rechts

< links kleiner dan rechts

>= links groter of gelijk aan rechts

<= links kleiner of gelijk aan rechts

5 Probeer bovenstaande en leg uit wat er gebeurt.