



KENNISMAKING MET PYTHON

Wil je graag leren programmeren? Hieronder vind je wat uitleg en leuke oefeningen. Snap je iets niet of zit je ergens vast, stel jouw vraag aan de coaches!

Om de volgende oefeningen te maken, maak gebruik van de overgangbladen van Scratch naar Python. Wanneer een hoofdstuk is afgerond neem je eens een kijkje naar de verbeteringen.

PRATEN MET DE COMPUTER VIA PYTHON

1. Als we een bericht willen sturen via de computer gebruiken we `print()`. Tussen de haakjes kunnen we een bepaalde tekst schrijven.

Voorbeeld:

```
print("Hallo!")
```

Probeer zelf `print()` uit met een bepaalde tekst tussenin!

Soms maken we gebruik van “variables”, dit zijn woorden waar we een bepaalde waarde in opslaan. Denk hier aan allemaal verschillende lades van een kast, waar je een woord of getal in kunt bewaren.

Ik geef een voorbeeld:

```
mijnNaam = "Tom"  
getal = 6  
mijnFavorieteZin = "Ik eet graag 5 pizzas"
```



Tekst zetten we tussen “ ” en wiskundige getallen niet.

2. Maak voor de tweede oefening een variable die jouw naam opslaat en probeer deze dan weer te geven via `print()`



Tip : Weet je niet meteen hoe eraan te beginnen? Neem dan een kijkje op de overgangbladen.

3. We willen natuurlijk ook bepaalde vragen kunnen stellen, waar de persoon die het programmatje gebruikt op kan antwoorden. We maken hier gebruik van de functie `input()`.

Voorbeeld:

```
print("Hoe oud ben jij?")  
leeftijd = input()  
print("Tom is " + leeftijd + "jaar oud")
```

Probeer dit zelf eens uit en als je dit snapt, kun je eens proberen om zowel de leeftijd als de naam op te vragen!

HOE SLIM IS DE COMPUTER?

Tijd voor wat wiskunde! Doe je dit niet zo graag? Geen probleem, de computer gaat al ons rekenwerk doen!

4. We gaan proberen twee getallen in te geven met `input()`, daarna gaat de computer kijken of `getal1` groter of kleiner is dan `getal2` en geeft hij het juiste antwoord weer.

Deze opdracht bestaat uit drie stappen, waarvan we twee stappen hierboven al geleerd hebben!

Stap 1. Met `input()` gaan we twee getallen opvragen. We maken hierbij gebruik van twee variables: `getal1` en `getal2`.

Stap 2. We gaan nu controlleren of `getal1` groter is dan `getal2`, dit doen we met `if/else-statement`.



`if` is Engels voor 'als' en `else` is Engels voor "als dit niet zo is".

Voorbeeld:

```
if getal1 > getal2:
    Doe dit
else:
    Doe dat
```



Bij de "doe dit" en "doe dat" moet je opletten. Telkens de computer wat moet uitvoeren bij de if of de else moet er een kleine inspringsing zijn met behulp van de tab-toets op je computer.

```
if 5 > 8:
→ print("5 is groter dan 8")
  print("8 is kleiner dan 5")
  print("Deze tekst wordt enkel weergegeven wanneer 5>8")
print("dit wordt altijd weergegeven")
```

Deze tab is zeer belangrijk.



Je kan niet enkel 'is kleiner dan' (<) of 'is groter dan' (>) gebruiken, maar ook 'is gelijk aan'. Dit doe je door '==' te schrijven in plaats van '='.

```
print("gok het getal")
gok = int(input())
if gok == 7:
    print("gewonnen!")
```

Stap 3. Tenslotte vervangen we "doe dit" en "doe dat" naar

```
print(). Een voorbeeld wat je zou kunnen gebruiken als tekst is:
print("het eerste getal is groter dan het tweede")
en print("het eerste getal is kleiner dan het
tweede").
```



COMPUTER LATEN TELLEN TOT 100

Een computer is niet alleen heel slim, maar ook heel snel. Zo kan een computer in een seconde tellen van 1 tot 100.

5. Om de computer te laten tellen gebruiken we een `for-lus`. Deze lus blijft zichzelf herhalen tot hij zijn doel heeft bereikt. (In dit geval, tot hij aan 100 zit)

Voorbeeld:

```
for i in range(100):  
    print(i)
```



i is een variable die zichzelf telkens + 1 gaat doen tot i (onze variabele) dit 100 keer gedaan heeft. Je kan i ook vervangen door getal, nummer, x, abc,... Maar programmeurs hebben de rare gewoonte om i te gebruiken.

Als je dit uitvoert, dan merk je op dat de computer niet van 1 – 100 telt maar van 0 – 99. Dit is omdat een computer begint te tellen vanaf 0 en daarom dus altijd eentje minder telt.

Dit kun je oplossen door:

```
for i in range(100):  
    print(i + 1)
```

Probeer eens in plaats van tot 100 te tellen, met een `int(input())` een bepaald getal op te vragen en de computer tot dit getal te laten tellen. Omdat we met wiskunde bezig zijn, moeten we de computer duidelijk maken dat dit getallen zijn. Hierdoor gaan we `int(input())` doen. `int()` vertelt de computer dat het hier om een getal gaat.



EEN LIJST MET DEELNEMERS

6. We gaan nu een lijstje maken van deelnemers die zogezegd meedoen met Coderdojo.

Aan de hand van simpele vragen gaan we drie acties kunnen ondernemen. We kunnen deelnemers toevoegen, bekijken en verwijderen.

Hoe gaan we dit aanpakken?

We geven de gebruiker de keuze om voor 1, 2 of 3 te kiezen. Afhankelijk van de keuze gaat de computer een bepaalde actie uitvoeren.

Bij deze oefening gaan we het weer stap voor stap aanpakken.

Eerst maken we een lijstje met opties waarbij je kan kiezen tussen 'deelnemers toevoegen', 'deelnemers verwijderen' of 'deelnemers bekijken'.

*Een lijst **maken** we door deze aan te geven met '['-haken. Een lijst is zoals een variable, alleen is dit een heel grote lade en kan men niet alleen één waarde erin steken, maar meerdere elementen. :*

```
deelnemers = ["Tom", "Lotte", "Sofie", "Sam"]
```

Aangezien we meerdere waardes (bijvoorbeeld namen, getallen, zinnen, ... gaan bijhouden, schrijven we deze variabelen in het meervoud 'deelnemers' in plaats van 'deelnemer'.

*Wil je een nieuwe deelnemer **toevoegen** (in het Engels: append):*

```
deelnemers.append("Stijn")
```

*Wil je de lijst van alle deelnemers **bekijken**:*

```
Print(deelnemers)
```

*Wil je een deelnemer **verwijderen** (in het Engels: remove)::*

```
deelnemers.remove("Lotte")
```



Nu we meer weten over lijstjes, gaan we kijken naar de acties die moeten uitgevoerd worden. We gaan een menu ontwikkelen waar we opvragen welke actie we willen uitvoeren.

We maken gebruik van `print()`, `input()` en een `if`-statement.

Voorbeeld:

```
print("Druk op 1 als je een deelnemer wilt  
toevoegen")  
print("Druk op 2 als je alle deelnemers wilt  
bekijken")  
print("Druk op 3 als je een deelnemer wilt  
verwijderen")
```

Eerst en vooral moeten we nu de actie opvragen die onze gebruiker wil uitvoeren. Dit doen we met een `input()`.

Voorbeeld:

```
actie = input()
```

Daarna gaan we met een `if`-statement controleren welke actie de computer moet uitvoeren. Bij nummer 1 en 3 moet je ook eerst opvragen welke naam ze willen toevoegen of welke naam je wilt verwijderen.

Voorbeeld:

```
if actie == "1":  
    Doe dit  
elif actie == "2":  
    Doe dat  
elif actie == "3":  
    Doe nog wat anders  
else:  
    print "Gelieven te kiezen tussen 1, 2 of 3"
```

Probeer nu aan de hand van de informatie die we hierboven hebben bekeken, de oefening te maken. Je hebt een lege lijst nodig want er



zijn nog geen deelnemers aanwezig. Dat doe je dus zo:

Voorbeeld:

```
deelnemers = []
```

Lukt het niet? Vraag hulp!

Wanneer je dit programmatje uittest, merk je misschien dat de computer maar eenmalig vraagt welke actie je wilt uitvoeren en dat je daarom dus ook maar één actie kunt uitvoeren. Dit is zeer vervelend, want hierdoor kun je geen goede deelnemerslijst opstellen. Om dit te verbeteren hebben we `while()` nodig. While betekent 'zolang' en de computer gaat iets blijven uitvoeren zolang een bepaalde waarde waar is.

Voorbeeld:

```
stoppen = false
while stoppen == false:
    print(stoppen)
```



Hier moet je even oppassen. Het voorbeeld gegeven hierboven is een oneindige lus. Dit is omdat deze voor altijd "false" of ook wel onwaar zal zijn. Wanneer jouw computer in zo'n oneindige lus geraakt moet je ctrl+c doen om het programmaatje even te stoppen.

Waarom is dit belangrijk voor ons coderdojo lijstje? We willen dat we oneindig lang kunnen werken aan onze deelnemerslijst, maar dat we met een bepaalde actie het programma kunnen stilleggen.

Voorbeeld:

```
Deelnemers []
actie = "een beginwaarde"
while actie != 0:
```

Hiertussen komt ons programma die we gemaakt hebben



(behalve onze lijst van deelnemers, want die zetten we erboven.
Waarom? Omdat deze lijst maar eenmalig moet aangemaakt worden).
Maar we toevoegen nog een extra print:

```
print("Druk op 0 als je het programma wilt  
afsluiten")
```

MEER DAN GEWOON TEKST

7. Wat we voorlopig gezien hebben is dat we door middel van code een bepaalde uitkomst in ons python venster krijgen. Door middel van *tkinter* kunnen we werken met een canvas (vergelijkbaar met een blad papier) waar we meer opties hebben dan alleen maar wat tekst.

Dit klinkt allemaal nog wat onduidelijk, dus laten we aan de slag gaan!

Tkinter moeten we importeren zodat python dit kan begrijpen, dit doen we door in het begin van ons programmaatje te schrijven. Je kunt dit vergelijken met het nemen van een blad papier:

```
from tkinter import *
```



Het sterretje '' zegt eigenlijk dat we alles uit tkinter gaan invoegen.*

Wat we ook moeten toevoegen is:

```
root = Tk()  
C = Canvas(root, bg="white", height=250,  
            width=300)
```



Dit kun je vergelijken met de eigenschappen van jouw papier: welke kleur dit heeft, hoe groot je papier is,...

```
C.pack() → het aanmaken  
root.mainloop() → dit moet helemaal op het einde van het  
programmaatje, hierdoor laat je de tekening zien.
```




Nu gaan we er een knop aan toevoegen.

Voorbeeld:

Press

```
knop = Button(root, text = 'Press', command =  
Pressed) → hier omschrijven we de knop.  
knop.pack() → hier word de knop gemaakt.
```

Laten we deze structuur even bekijken.

'knop' is een variable, je kan deze naam dus aanpassen. Daarna roepen we de functie `Button()` op. Let hier op de hoofdletter!

'root' is ons canvas (zo hebben we die benoemd), tekst = 'Press' is de tekst die wordt weergegeven op je knop, je kunt hier ook eens wat anders schrijven!

'command = tekst' dit is de actie die moet uitvoeren wanneer er op de knop wordt gedrukt. Aan het woordje tekst moeten we nu een actie koppelen zoals je hieronder ziet.

Voorbeeld:

```
def Pressed():  
    print("hallo, ik heb op de knop gedrukt")
```

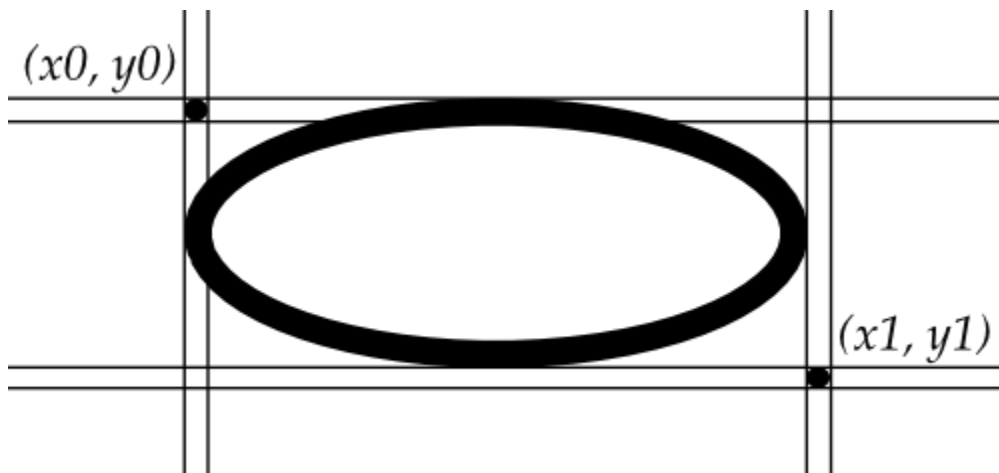
Probeer dit nu eens zelf uit. Als je alle elementen zou toevoegen en je zou het uitproberen zou je bij het drukken van de knop de opgegeven tekst krijgen: hallo, ik heb op de knop gedrukt

Is dit gelukt? Dan gaan we een stap verder. Zoniet, roep je er even een coach bij die je dit zal uitleggen. We gaan nu bekijken hoe we een figuur gaan tekenen op het canvas.



Vervang de print() door:

```
oval = C.create_oval(50, 50, 100, 100, fill="red")
```



De getallen zijn de coördinaten(x_0 , y_0 , x_1 , y_2) en als je op de knop drukt wordt er een cirkel getekent. Natuurlijk zou het interessant zijn als er elke keer dat je op de knop drukt een cirkel op een willekeurige plaats zou tevoorschijn komen. We doen dit aan de hand van de functie `random` toe te voegen aan ons programma. Dit doen we helemaal bovenaan op de volgende manier:

Voorbeeld:

```
import random → random is Engels voor willekeurig
```

Daarna schrijven we het volgende:

```
def Pressed():  
    getal1 = random.randint(1, 250)  
    getal2 = random.randint(1, 300)  
    oval = C.create_oval(getal1, getal2, getal2,  
        getal1, fill="red")
```



Wat doet `'getal1 = random.randint(1, 250)'`?

Aangezien we ons blad papier 250 hoog hebben gemaakt en 300 breed, gaan we bij `getal1` een willekeurig getal laten kiezen tussen 1 en 250 en bij `getal2` van 1 tot 300.

`Getal1` gebruiken we dan als verschil in hoogte en `getal2` in breedte

Lukt dit? Probeer dan eens 'oval' te vervangen door rectangle (rechthoek) en de kleur 'red' door een andere kleur.

Je kunt kiezen tussen deze kleuren:

'red', 'green', 'blue', 'yellow', 'orange', 'white', 'cyan', 'purple', 'black'

Je hebt kennis gemaakt met python en je hebt ontdekt wat de verschillende mogelijkheden zijn van python. Hieronder vind je nog bepaalde oefeningen die je kunt proberen te maken! Heb je nog vragen over Python? Is er iets dat je nog niet goed snapt of je bent nieuwsgierig naar nog meer mogelijkheden van python, stel gerust je vragen aan de begeleiders!



VERBETERINGEN

Bij elk hoofdstuk geef ik het uiteindelijke eindresultaat. Heb je iets anders? Of werkt het bij jou niet? Vraag gerust hulp!

PRATEN MET DE COMPUTER VIA PYTHON

Vul je eigen naam in een variable:

```
naam = "Aagje"  
print(naam)
```

Vraag zowel de naam als de leeftijd op:

```
print("Wat is je naam?")  
naam = input()  
print("Wat is je leeftijd?")  
leeftijd = input()  
print("Hallo ik ben " + naam + " en ik ben " + leeftijd + " jaar oud")
```

HOE SLIM IS DE COMPUTER?

Controleer of getal1 groter of kleiner is dan getal2:

```
print("geef getal1 op")  
getal1 = input()  
print("geef getal2 op")  
getal2 = input()  
if getal1 > getal2:  
    print("getal1 is groter dan getal2")  
else:  
    print("getal2 is groter dan getal1")
```



COMPUTER LATEN TELLEN TOT 100

Kies zelf een getal:

```
getal = int(input())
for i in range(getal):
    print(i+1)
```

EEN LIJST MET DEELNEMERS

Een deelnemers van coderdojo toevoegen, bekijken en verwijderen:

```
deelnemers= []
actie = "een beginwaarde"
while actie != "0":
    print("Druk op nul als je wilt afsluiten")
    print ("Druk op 1 als je een deelnemers wilt toevoegen")
    print ("Druk op 2 als je alle deelnemers wilt bekijken")
    print ("Druk op 3 als je een deelnemer wilt verwijderen")
    actie = input()
    if actie == "1":
        print("Welke naam wil je toevoegen")
        naam = input()
        deelnemers.append(naam)
    elif actie == "2":
        print(deelnemers)
    elif actie == "3":
        print("Welke naam wil je verwijderen uit de lijst?")
        print(deelnemers)
        naam = input()
        deelnemers.remove(naam)
    else:
        print("gelieve te kiezen tussen 1, 2 of 3")
```



MEER DAN GEWOON TEKST

Willekeurige bollen op het scherm tekenen:

```
from tkinter import *
import random
def Pressed():
    getal1 = random.randint(1, 250)
    getal2 = random.randint(1, 300)
    oval = C.create_rectangle(getal1, getal2, getal2, getal1, fill="yellow")

root = Tk()
C = Canvas(root, bg="white", height=250, width=300)
button = Button(root, text = 'Press', command = Pressed)
button.pack()
C.pack()
root.mainloop()
```



EXTRA OEFENINGEN

Ben je helemaal klaar met de cursus? Of heb je alvast een goede kennis van Python? Hieronder vind je verschillende oefeningen die je kunt proberen te maken.

1. In onze cursus hebben we een programmaatje gemaakt die cirkels willekeurig in ons canvas plaatst als we drukken op een knop.
Probeer nu het zo te maken dat deze cirkel elke keer een andere kleur krijgt.
Tip: Zet alle kleuren in een lijstje (`kleur = ['red', 'green', 'blue', 'yellow', 'orange', 'white', 'cyan', 'purple']`) en om hier willekeurig een kleur uit te kiezen: `gekozenKleur = random.choice(kleur)`
Probeer de variable *gekozenKleur* op te roepen bij de 'fill' van de cirkel.
2. Maak een gokspelletje met de volgende regels:
Er wordt een willekeurig nummer gegenereerd tussen 0 en 20 (in de cursus staat `random()` uitgelegd). De speler mag telkens een nummer raden. Je mag blijven raden totdat je het nummer geraden hebt (gebruik `while()`). Elke keer als je een gok waagt krijg je een bericht of je onder het getal zit of erboven. ('>' of '<')
3. Maak een rekenmachientje. Vraag op of ze willen optellen, aftrekken, vermenigvuldigen of delen (In Python noteer je deze wiskundige berekeningen met '+', '-', '*' en '/'). En vraag daarna op welke twee getallen.
Tip: wanneer je twee getallen opvraagt, let op dat je met wiskunde bezig bent. Deze moeten dus omgezet worden naar een "getal" door middel van `int()`.