BEGINNEN MET PYTHON

CoderDojo

PYTHON CONFIGURATIE

Kaart 1 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Je gaat leren programmeren en een spel bouwen met de programmeertaal Python.

Wist je dat...?

Websites zoals YouTube en Instagram zijn gebouwd met Python.

Voordat je leert programmeren, moet je jouw pc zo instellen dat je in Python kunt schrijven. Je moet Python en een teksteditor installeren en weten hoe een Python programma uitgevoerd wordt.

2 Om Python te installeren, ga je naar [dojo.soy/py-setup] (http://dojo.soy/py-setup) en klik je op de **Download Pyhton3** knop. Er komen meer nummers na de 3, maar die veranderen zo vaak dat we die niet gaan gebruiken. Sla ze maar over.

Als het installatieprogramma gedownload is, start je het op en klik je steeds op 'volgende' waarbij je de standaardinstellingen accepteert.

3 Nu moet je een teksteditor downloaden waarin je Python schrijft. Wij raden Atom aan dat je kunt downloaden van http://atom.io, maar je mag ook een andere teksteditor gebruiken als je daar meer vertrouwd mee bent.

CoderDojo

PYTHON CONFIGURATIE

Kaart 1 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

4

Als je beide programma's geïnstalleerd hebt, ben je klaar om te beginnen. Je moet alleen zeker weten dat alles werkt en hoe je een Python programma moet uitvoeren. Voer deze stappen uit:

- Maak een nieuwe map aan voor je Python Sushi Kaarten project.
- Open je teksteditor en maak een nieuw bestand aan. Bewaar het in de map die je net gemaakt hebt en geef het de naam **beginner_sushi.py**.
- Open de command line (opdrachtprompt) op je computer (dit is **Opdrachtprompt** bij Windows en **Terminal** bij Mac) en ga naar map met behulp van het **cd** commando.
- Als je de map geopend hebt op de command line, dan kun je dit lege bestand opstarten met het volgende commando:

```
python3 beginner_sushi.py
```

Als dit goed werkt, zie je geen berichten als je het commando uitvoert.

Een programma uitvoeren

Later zal je gevraagd worden een programma **uit te voeren**. Dat betekent: wat je net gedaan hebt, naar een map gaan waarin het programma staat en met het **python3** commando en de bestandsnaam het programma uit te voeren!

BEGINNEN MET PYTHON

CoderDojo

JE EERSTE PYTHON PROGRAMMA

Kaart 2 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Tijd voor je eerste stukje Python code. Je gaat de computer hallo laten zeggen tegen iedereen. Typ het volgende in je bestand:

```
print ("Hallo allemaal")
```

Voer deze code uit en zie wat er gebeurt! Verander wat er binnen de aanhalingstekens staat, bijvoorbeeld met je naam, en voer het opnieuw uit.

2 Voeg nu een regel toe, probeer je code er zo uit te laten zien:

```
print ("Hallo allemaal")
print ("De Code... Roept je. Luister dan toch.")
```

Voer de code opnieuw uit. Zie je hoe de tekst (dat heet een **string**) van de tweede **print** op een nieuwe regel staat? Dit komt omdat de instructie die de computer krijgt als je de **print** opdracht geeft, is:

- Lees de tekst binnen de haakjes en bedenk wat het resultaat is.
- Als je bedacht hebt wat er gezegd wordt, **print** dat dan op het scherm.
- Plaats een onzichtbare "begin een nieuwe regel" aan het eind.

CoderDojo

JE EERSTE PYTHON PROGRAMMA

Kaart 2 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

3 Waarom moet de computer bedenken wat de code tussen de haakjes wil? Omdat de computer de **string** samen kan voegen met delen die je in het programma meegeeft. Probeer het uit! Gebruik deze code, maar typ jouw naam op de plek "mijn naam" (gebruik wel de aanhalingstekens "!):

```
name = "mijn naam"
print ("Hallo "+name)
print ("De Code... Roept je. Luister dan toch.")
```

De spatie na "Hello"

Je moet een spatie toevoegen vóór de **variabele** 'name' anders verschijnt er "Hallomijn naam" op je scherm!

4 Je hebt een **variabele** gebruikt, namelijk **name** (naam). Dit lijkt op een vakje in de computer met een etiket erop. Je kunt van alles in dat vakje stoppen. Vervolgens kun je het etiket gebruiken om Python de inhoud van het vakje in je code te laten gebruiken. Je hebt de variabele **name** gemaakt en er **"[mijn naam]"** in opgeslagen. In de volgende regel heb je de **variabele** gebruikt om die naam in de begroeting te zetten, door het **+** teken te gebruiken om het aan het eind van de **string** toe te voegen.

BEGINNEN MET PYTHON

CoderDojo

PRATEN MET DE GEBRUIKER

Kaart 3 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Leuk hoor, om de computer jouw naam achter "Hello" te laten zetten, maar waarom kun je niet gewoon "Hello [my name]" schrijven? Omdat je met een **variabele** niet hoeft te weten wat er in komt te staan als je een programma schrijft. Je kunt zelfs aan de gebruiker van het programma vragen om jou te vertellen wat er in gezet moet worden. Pas je Python programma aan:

```
name = input("Wat is je naam?")
print ("Hallo "+name)
print ("De Code... Roept je. Luister dan toch.")
```

Voer het uit. Je moet op de "Enter" toets drukken als je de naam hebt ingetypt.

2 Probeer nu eens een getal van je gebruiker te krijgen. Let op: je kunt het + teken aan beide kanten van een variabele gebruiken.

Voer dit programma uit, beantwoord de vragen, en kijk wat er gebeurt.

```
name = input("Wat is je naam?")
my_number = input("Hallo "+name+" kies een getal uit")
print ("Je getal is "+my_number)
```

BEGINNEN MET PYTHON

CoderDojo

PRATEN MET DE GEBRUIKER

Kaart 3 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

3 Wat als je een getal aan je variabele wilt toevoegen? Voeg een regel aan je programma toe die het cijfer 1 optelt bij de **my_number** variabele.

```
name = input("Wat is je naam?")
my_number = input("Hallo "+name+" kies een getal uit")
my_number = int(my_number) + 1
print ( "Je getal is "+str(my_number))
```

Je hebt een waarde van een variabele gepakt, die aangepast en opgeslagen in diezelfde variabele en dat allemaal op één regel!

Maar waarom staat er **int()** en **str()** rond **my_number**?

Dat komt omdat als Python aan het getal '1' denkt, wanneer het gebruikt wordt voor rekenen, dat niet hetzelfde is als '1' wanneer het in een zin gebruikt wordt. Door **int()** om een variabele heen te zetten wordt deze gezien als een **integer** (een 'geheel getal'), en door er **str()** omheen te zetten beschouwt Python de variabele als tekst.

Integers en **strings** zijn **typen** variabelen. Bepaalde stukken code (zoals **+** en **print**) werken alleen als je ze het juiste type variabele geeft.

Rekenen

Je hebt nu geleerd hoe je kunt optellen, maar je kunt ook:

- aftrekken door **-** te gebruiken
- vermenigvuldigen door * te gebruiken
- delen door / te gebruiken

BEGINNEN MET PYTHON

CoderDojo

VERGELIJKEN & 'IF' UITDRUKKINGEN

Kaart 4 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Je kunt Python twee getallen met elkaar laten vergelijken. Dit kan heel handig zijn (heeft de speler genoeg geld om te betalen?). Je kunt deze speciale tekens gebruiken:

- **a > b** vraagt of **a** groter is dan **b**
- **a < b** vraagt of **a** kleiner is dan **b**
- **a == b** vraagt of **a** hetzelfde is als **b**
- **a != b** vraagt of **a** niet even groot is als **b**
- **a >= b** vraagt of **a** groter is dan of even groot is als **b**
- **a <= b** vraagt of **a** kleiner is dan of even klein is als **b**

==

Het dubbele = teken wordt gebruikt om variabelen te **vergelijken**, een enkel = teken wordt gebruikt om waarden **toe te kennen**.

2 Je gebruikt een vergelijking binnen een **if** (als) uitdrukking: code die alleen uitgevoerd wordt als de uitdrukking (binnen de haakjes) waar is.

```
if (my_number > 100):
    print ("Dat is een groot getal!")
```

Inspringen

De **print** **springt in**. Er staan vier spaties aan het begin van de regel. Die heeft Python nodig om je programma te begrijpen.

BEGINNEN MET PYTHON

CoderDojo

VERGELIJKEN & 'IF' UITDRUKKINGEN

Kaart 4 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

3 Nu voeg je dat stukje code toe aan je programma van de vorige kaart. Pas het programma aan zodat het er zo uit komt te zien:

```
name = input("Wat is je naam?")
my_number = input("Hallo "+name+" kies een getal uit")
my_number = int(my_number)
print ("Je getal is "+str(my_number))

if (my_number > 100):
    print ("Dat is een groot getal!")
```

Voer de code uit door verschillende getallen in te voeren, zowel boven als onder 100 om te zien wat er gebeurt. Wat zou er gebeuren als je 100 intypt?

4 Je kunt ook voorwaarden combineren, door gebruik te maken van **and** (en) en **or** (of), zodat je de volgende code kunt schrijven:

```
if (my_number >=20 and my_number < 30):
    print ("Dat getal ligt tussen de 20 en 30!")
```

Of bijvoorbeeld:

```
if (food == "Cake" or food == "Chocolade" or food ==
"Taart"):
    print ("Klinkt lekker!")
```

BEGINNEN MET PYTHON

CoderDojo

BIJZONDERE 'IF' UITDRUKKINGEN

Kaart 5 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Wat als je wilt kijken of het getal van de gebruiker groot genoeg is en je ze wilt kunnen melden als dat niet zo is? Stel dat het getal groter is dan 100. Of je complimenteert de gebruiker dat hij een getal opgeeft dat groot genoeg is, of je vertelt de gebruiker waarom het getal niet goed is. Probeer dit uit:

```
name = input("Wat is je naam?")
my_number = input("Hallo "+name+", kies een getal
boven de 100")
my_number = int(my_number)
print ( "Je getal is "+str(my_number))

if (my_number > 100):
    print ("Dat is een groot getal!")
else:
    print ("Dat getal is te klein!")
```

Hier werkt de **else** (anders) als een soort tegengestelde **if** (als) uitdrukking. De code onder de 'else' wordt alleen uitgevoerd als dat wat in de 'if' uitdrukking staat **niet** waar is.

CoderDojo

BIJZONDERE 'IF' UITDRUKKINGEN

Kaart 5 van 7
Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

2

Wat als je de gebruiker wil vertellen dat ze dicht bij het juiste antwoord zitten? Stel dat ze een getal boven de 90 hebben uitgekozen. Dan gebruik je een **elif**. Dat is een samengevoegde **else** en **if** uitdrukking, want dit gebeurt alleen als de invoer in de **if** uitdrukking **niet** waar is en als de invoer in de **elif** uitdrukking **wel**. Dit is wat je toe moet voegen om de gebruiker te vertellen dat ze dicht bij het goede antwoord zitten:

```
elif (my_number > 90):
    print ("Bijna goed!")
```

En zo ziet het eruit samen met de rest van het programma. Merk op dat **elif** tussen **if** en **else** moet staan.

```
name = input("Wat is je naam?")
my_number = input("Hallo "+name+" kies een getal boven
de 100")
my_number = int(my_number)
print ("Je getal is "+str(my_number))

if (my_number > 100):
    print ("Dat is een groot getal!")
elif (my_number > 90):
    print ("Bijna goed!")
else:
    print ("Dat getal is te klein!")
```

CoderDojo

LUSSEN

Kaart 6 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Nu kun je de gebruiker vragen om een getal, nakijken of het goed is, en zo niet, ze dat ook vertellen. Wat als je net zo lang door wilt gaan tot het antwoord goed is? Je kunt **if** uitdrukkingen binnen **if** uitdrukkingen blijven schrijven, maar wat als de gebruiker steeds het verkeerde antwoord blijft geven?

Je moet een manier vinden om de vraag steeds opnieuw te stellen tot je het juiste antwoord krijgt. Binnen computerprogrammeren heet dit een **loop** (lus). Je gaat een lus gebruiken die de **while** (terwijl) loop heet..

2 Een **while** loop lijkt een beetje op een **if** uitdrukking: het heeft code die alleen uitgevoerd word als de voorwaarde binnen de haakjes waar is. Het verschil is dat een **while** loop blijft doorgaan, tot de voorwaarde niet langer waar is. Je moet wel zorgen dat de **while** loop gestopt kan worden, anders blijft deze eeuwig doorgaan! Het ziet er zo uit:

```
while (my_number < 100):
    my_number = input("Hallo "+name+" kies een getal
boven de 100")
    my_number = int(my_number)
```

CoderDojo

LUSSEN

Kaart 6 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

3 Voeg nu een **while** loop aan je programma toe.

```
name = input("Wat is je naam?")
    my_number = 0

# doe de while loop zolang "my_number" kleiner is dan
100
while (my_number < 100):
    # Vraag de gebruiker om een getal
    my_number = input("Hallo "+name+" kies een getal
boven de 100")
    # Converteer van een string naar een getal
    my_number = int(my_number)
    print ("Je getal is "+str(my_number))
    # Controleer of het getal groter is dan 100
    if (my_number > 100):
        print ("Dat is een groot getal!")
    elif (my_number > 90):
        print ("Bijna goed! Probeer het nog eens!")
    else:
        print ("Dat getal is te klein! Probeer het nog
eens!")
    # Als my_number kleiner is dan 100, herhaal de
loop.
```

Commentaarregels

Dit zijn tekstregels voor programmeurs (of voor jezelf, in de toekomst) waar de computer niets mee doet. In Python beginnen die regels met een # en dat werkt door tot het einde van de regel.

BEGINNEN MET PYTHON

CoderDojo

EEN SPEL MAKEN

Kaart 7 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

1 Nu heb je het volgende geleerd:

- **print** uitdrukkingen: praten met de gebruiker.
- variabelen: een manier voor je programma om waarden te onthouden en aan te passen.
- strings: regels tekst.
- **input**: hoe je informatie van je gebruiker krijgt.
- rekenen: berekeningen uitvoeren met getallen.
- integers: 'gehele getallen' voor berekeningen.
- **if** uitdrukkingen: iets doen op basis van een voorwaarde.
- **while** lussen: blijf iets doen tot een voorwaarde niet waar is.

2 Probeer dit te gebruiken om een spel te maken:

- Er is een getal (een integer), tussen 1 en 9, dat het programma stiekem kiest
- de speler kan 5 keer raden
- het spel leert de speler de spelregels
- de speler wordt na elk antwoord verteld of het te laag, te hoog of goed is, en hoe vaak ze nog mogen antwoorden
- als de speler het juiste antwoord raadt, krijgt hij een speciale boodschap dat hij gewonnen heeft
- als de speler voor de vijfde keer het foute antwoord geeft, is het spel voorbij en verliest hij.

3 Een voorbeeld van het spel vind je op [dojo.soy/py-dice] (http://dojo.soy/py-dice).

CoderDojo

EEN SPEL MAKEN

Kaart 7 van 7

Ik leer: Python

Geproduceerd door CoderDojo[kennemerwaard]

4 Er ontbreekt nog één stukje om dit spel te kunnen maken: een manier om een willekeurig getal tussen 1 en 9 te kiezen. De benodigde code gaat wat verder dan deze les, dus zie het volgende maar even als iets magisch. Het wordt in latere Sushi Kaarten uitgelegd.
Zet dit als **eerste regel** in je programma:

```
from random import randint as dice
```

Als je nu een willekeurig getal tussen 1 en 9 nodig hebt, gebruik je: dice(1,9). Bijvoorbeeld:

```
secret_number = dice(1,9)
```

5 Probeer nu het spel te maken! Gebruik daarbij ook de vorige kaarten. Als je vastzit, of klaar bent, kijk dan voor de oplossing op [dojo.soy/py-guess] (http://dojo.soy/py-guess). Het geeft niet als jouw code er anders uit ziet, als de code maar werkt! Succes!

Wat vond je ervan?

Je bent klaar met deze serie Sushi Kaarten. Ik wil graag weten wat je ervan vond. Als je wilt, laat het me weten via [dojo.soy/py-beginner] (http://dojo.soy/py-beginner).