



Boter kaas en Eieren

Python workshop

Gemaakt door Quinten Cabo

We gaan vandaag boter kaas en eieren maken! Ik ga er nu van uit dat je een klein beetje basis python kan.

Eerst heb je een bord nodig met print!

```
print("""
-----
| {} | {} | {} |
-----
| {} | {} | {} |
-----
| {} | {} | {} |
-----""")
```

Door een string te """" gebruiken in de print kan je meer dan 1 regel printen zonder gedoe met \n

Dan gaan we nu deze print in een functie zetten zodat we steeds makkelijk het bord kunnen printen in een keer.

Een functie is een commando dat je zelf maakt! Best wel cool. Je maakt een functie zo:

```
def naam_functie():
```

Dus eerst "def" dat staat voor het Engelse word "define",

Dan de naam van de functie

Dan 2 haakjes om input te geven aan de functie

En dan een :

Die : is erg belangrijk want die zegt dat de volgende code 1 tab vooruit moet.

Alle code die 1 tab vooruit staat na de : wordt uitgevoerd als je de naam van de functie gebruikt

Dus:

```
def print_bord():  
    print("""  
    -----  
    | {} | {} | {} |  
    -----  
    | {} | {} | {} |  
    -----  
    | {} | {} | {} |  
    -----""")
```

Super nu kunnen we het bord neerzetten door alleen maar `print_bord()` onder onze code te zetten!

Goed nu hebben we een bord maar er gebeurt nog niet zo veel. We moeten nu er voor zorgen dat je input kan geven

En dat vervolgens het bord aangepast word. Dit kunnen we goed doen met een lijst. Een lijst is in python net zoals een lijst in real-life. Je kan er alles wat je wilt opschrijven. Je maakt een lijst met deze `[]` haakjes.

```
nieuwe_lijt = []  
print(nieuwe_lijt)
```

Woah nu hebben we een lege lijst!

Laten we nu een lijst maken met wat erin!

```
nieuwe_lijt = ["appel", "peer", 1, 2, 3, 4.5, "Daniel"]  
print(nieuwe_lijt)
```

woah nu hebben we een lijst met van alles erin!

Je kan ook 1 specifiek ding uit een lijst printen dat doe je met een index

```
nieuwe_lijt = ["appel", "peer", "banaan", 2, 3, 4.5, "Daniel"]  
print(nieuwe_lijt[0])
```

Dit print het item van de lijst dat op plek 0 staat in de lijst. Plek 0 is de eerste plek dus er wordt nu alleen maar appel geprint van de lijst. Wat je hier dus ziet is dat je begint met tellen vanaf plek 0. Dus *nieuwe_lijt[1]* is peer en *nieuwe_lijt[2]* is banaan. Terwijl banaan dus op plek 3 staat in de lijst. Dat maakt niet uit want de index is 2.

Ok maar wat hebben we hier aan? Nou die indexen zijn super handig want je kan ermee ook een specifiek punt in de lijst aanpassen

```
nieuwe_lijt = ["appel", "peer", "banaan", 2, 3, 4.5, "Daniel"]
print(nieuwe_lijt)
nieuwe_lijt[1] = "kiwi"
print(nieuwe_lijt)
```

Hier hebben we dus banaan in kiwi veranderd! Dus als we dit kunnen we ook plekken voor het boter kaas en eieren bord aanpassen
Maar eerst moeten we nu de lijst verbinden met de print

```
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0], l[1], l[2], l[3], l[4], l[5], l[6], l[7], l[8])

bord_lijt = [1, 2, 3, 4, 5, 6, 7, 8, 9]
print_bord(bord_lijt)
```

Ok woah dit ziet er wel ineens er ingewikkeld uit! Maar opzich als je erover nadenkt valt het wel mee. Met functies kan je een input geven
Die input is bord_lijt. Die word mee geven aan de print bord functie met de naam l. Vervolgens wordt elke {} in de print vervangen met de juiste index van de bord lijst met de format methode

Nu kunnen we dus met *print_bord(bord_lijt)* altijd de status van het bord printen!

Dan nu nog 1 stap om ook echt een input te geven!

```
def print_bord(l):  
    print("""  
    -----  
    | {} | {} | {} |  
    -----  
    | {} | {} | {} |  
    -----  
    | {} | {} | {} |  
    -----  
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8]))  
  
bord_lijsjt = [1,2,3,4,5,6,7,8,9]  
print_bord(bord_lijsjt)  
vakje = input("Welk vakje wil je pakken:")  
vakje = int(vakje)-1  
bord_lijsjt[vakje] = "X"  
print_bord(bord_lijsjt)
```

Probeer deze code te runnen en kijk wat er gebeurt. Nu komt de uitleg.

We pakken een input. Die slaan we op in vakje. Daar halen maken we een nummer van en dan halen we 1 van ervan af om de cijfers te laten kloppen met de index.

Vervolgens gebruiken we vakje als index om de lijst aan te passen. En dan printen we het bord met de nieuwe lijst! Niet zo heel ingewikkeld!

Als we dit in een loop zetten hebben we al heel wat!

```
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8]))

bord_lijst = [1,2,3,4,5,6,7,8,9]
while True:
    print_bord(bord_lijst)
    vakje = input("Welk vakje wil je pakken:")
    vakje = int(vakje)-1
    bord_lijst[vakje] = "X"
```

Dit is de basis. Alleen nu blijft het wel doorgaan. Dus we moeten een paar checks toevoegen dat je niet kan vals spelen en altijd een nummer doet

```
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8]))

bord_lijst = [1,2,3,4,5,6,7,8,9]
while True:
    print_bord(bord_lijst)
    vakje = input("Welk vakje wil je pakken:")
    if vakje not in ["1","2","3","4","5","6","7","8","9"]:
        print("Geen goede input!")
        continue
    vakje = int(vakje)-1
    if bord_lijst[vakje] not in [0,1,2,3,4,5,6,7,8]:
        print("Dat vakje is al bezet!")
        continue
    bord_lijst[vakje] = "X"
```

Super! Nu kan je niet echt meer vals spelen. Alleen nu word het bord alleen nog maar gevuld met X je hebt ook O nodig voor dit spel.
Dus moeten we nu een beurt toevoegen. En wat comment om het allemaal wat duidelijker te maken

```
# Functie om het bord te printen
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8]))

bord_lijsjt = [1,2,3,4,5,6,7,8,9]
beurt = "O"
while True:
    print_bord(bord_lijsjt)
    vakje = input("Welk vakje wil je pakken:")
    # check of de input goed is
    if vakje not in ["1","2","3","4","5","6","7","8","9"]:
        print("Geen goede input!")
        continue
    vakje = int(vakje)-1
    if bord_lijsjt[vakje] not in [0,1,2,3,4,5,6,7,8]:
        print("Dat vakje is al bezet!")
        continue
    # verander de beurt
    if beurt == "X":
        beurt = "O"
    elif beurt == "O":
        beurt = "X"
    bord_lijsjt[vakje] = beurt
```

Nu veranderd de beurt elke keer! namelijk als het een X is wordt het een O en als het een O is wordt het een X
Prachtig! in principe hebben we nu het basis spel gemaakt.
Nu gaan we nog een paar leuke dingen toevoegen zoals auto win detectie. Dit doen we met 8 lange if statements


```

# Functie om het bord te printen
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8])

while True:
    bord_lijsjt = [1,2,3,4,5,6,7,8,9]
    beurt = "0"
    winnaar = ""
    while True:
        print_bord(bord_lijsjt)
        vakje = input("Welk vakje wil je pakken:")
        # check of de input goed is
        if vakje not in ["1","2","3","4","5","6","7","8","9"]:
            print("Geen goede input!")
            continue
        vakje = int(vakje)-1
        if bord_lijsjt[vakje] not in [0,1,2,3,4,5,6,7,8]:
            print("Dat vakje is al bezet!")
            continue
        # verander de beurt
        if beurt == "X":
            beurt = "0"
        elif beurt == "0":
            beurt = "X"
        bord_lijsjt[vakje] = beurt

        if bord_lijsjt[0] == bord_lijsjt[1] and bord_lijsjt[1] == bord_lijsjt[2]:
            winnaar = bord_lijsjt[0]
        elif bord_lijsjt[3] == bord_lijsjt[4] and bord_lijsjt[4] == bord_lijsjt[6]:
            winnaar = bord_lijsjt[3]
        elif bord_lijsjt[6] == bord_lijsjt[7] and bord_lijsjt[7] == bord_lijsjt[8]:
            winnaar = bord_lijsjt[6]

        elif bord_lijsjt[0] == bord_lijsjt[3] and bord_lijsjt[3] == bord_lijsjt[6]:
            winnaar = bord_lijsjt[0]
        elif bord_lijsjt[1] == bord_lijsjt[4] and bord_lijsjt[4] == bord_lijsjt[7]:
            winnaar = bord_lijsjt[1]
        elif bord_lijsjt[3] == bord_lijsjt[5] and bord_lijsjt[5] == bord_lijsjt[8]:
            winnaar = bord_lijsjt[3]

        elif bord_lijsjt[0] == bord_lijsjt[4] and bord_lijsjt[4] == bord_lijsjt[8]:
            winnaar = bord_lijsjt[0]
        elif bord_lijsjt[2] == bord_lijsjt[4] and bord_lijsjt[4] == bord_lijsjt[6]:
            winnaar = bord_lijsjt[2]

        if winnaar:
            nog_een_keer = input("{} heeft gewonnen! Wil je nog een keer spelen (y/n):".format(winnaar)).lower()
            if nog_een_keer in ("n","nee","no"):
                exit()
            else:
                break

```

Wat we hier doen is checken voor elke mogelijke win conditie. Als er een gehaald word dan word de while loop gebroken en de game gereset.
Lees de code even goed!

Nu willen we het tegen een AI laten spelen. AI staat voor artificiële intelligentie. Dat klinkt erg cool natuurlijk. Maar onze AI wordt niet zo heel slim hij gaat vooral gewoon een beetje gokken. En dat gokken dat gaan we doen met de random module.

```
import random
print(random.randint(0,8))
```

Met dit kan je een random nummer printen tussen 2 getalen hier is het zonder enige reden een getal tussen 0 en 8

Met dit kunnen we de AI laten gokken op een nummer. Nu gaan we als het de beurt is van de ai gokken totdat we een vakje krijgen dat nog niet genomen is En dat wordt dat de keuze van de ai

```
import random
if beurt == "O":
    beurt = "X"
    ai_keuze = random.randint(0, 8)
    while bord_lijst[ai_keuze] not in [0, 1, 2, 3, 4, 5, 6, 7, 8]:
        ai_keuze = random.randint(0, 8)
    bord_lijst[ai_keuze] = beurt
    continue
```

Nu plakken we dat in de code (De code gaat door op de volgende pagina)

```
import random
def print_bord(l):
    print("""
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    | {} | {} | {} |
    -----
    """).format(l[0],l[1],l[2],l[3],l[4],l[5],l[6],l[7],l[8]))
```

```

while True:
    bord_lijst = [1,2,3,4,5,6,7,8,9]
    beurt = "0"
    winnaar = ""
    while True:
        if beurt == "X":
            beurt = "0"
        elif beurt == "0":
            beurt = "X"
            ai_keuze = random.randint(0, 8)
            while bord_lijst[ai_keuze] not in [0, 1, 2, 3, 4, 5, 6, 7, 8]:
                ai_keuze = random.randint(0, 8)
            bord_lijst[ai_keuze] = beurt
            continue
    print_bord(bord_lijst)

    vakje = input("Welk vakje wil je pakken:")
    # check of de input goed is
    if vakje not in ["1", "2", "3", "4", "5", "6", "7", "8", "9"]:
        print("Geen goede input!")
        continue
    vakje = int(vakje)-1
    if bord_lijst[vakje] not in [0,1,2,3,4,5,6,7,8]:
        print("Dat vakje is al bezet!")
        continue
    # verander de beurt

    bord_lijst[vakje] = beurt

    if bord_lijst[0] == bord_lijst[1] and bord_lijst[1] == bord_lijst[2]:
        winnaar = bord_lijst[0]
    elif bord_lijst[3] == bord_lijst[4] and bord_lijst[4] == bord_lijst[6]:
        winnaar = bord_lijst[3]
    elif bord_lijst[6] == bord_lijst[7] and bord_lijst[7] == bord_lijst[8]:
        winnaar = bord_lijst[6]

    elif bord_lijst[0] == bord_lijst[3] and bord_lijst[3] == bord_lijst[6]:
        winnaar = bord_lijst[0]
    elif bord_lijst[1] == bord_lijst[4] and bord_lijst[4] == bord_lijst[7]:
        winnaar = bord_lijst[1]
    elif bord_lijst[3] == bord_lijst[5] and bord_lijst[5] == bord_lijst[8]:
        winnaar = bord_lijst[3]

    elif bord_lijst[0] == bord_lijst[4] and bord_lijst[4] == bord_lijst[8]:
        winnaar = bord_lijst[0]
    elif bord_lijst[2] == bord_lijst[4] and bord_lijst[4] == bord_lijst[6]:
        winnaar = bord_lijst[2]

    if winnaar:
        nog_een_keer = input("{} heeft gewonnen! Wil je nog een keer spelen (y/n):".format(winnaar)).lower()
        if nog_een_keer in ("n", "nee", "no"):
            exit()
        else:
            break

```

En klaar!