

---

# **Python @ CoderDojo Gent**

**Peter Van Hese**

**05 april 2019**



<b>1</b>	<b>Starten met Python</b>	<b>3</b>
1.1	Waarom Python? . . . . .	3
1.2	Installatie . . . . .	3
1.3	Kennismaking met Python . . . . .	8
1.4	Schildpad . . . . .	9
<b>2</b>	<b>Boeken</b>	<b>13</b>
<b>3</b>	<b>Projecten</b>	<b>15</b>
3.1	Snake . . . . .	15
3.2	Pong . . . . .	19
<b>4</b>	<b>Extra oefeningen</b>	<b>23</b>
4.1	Koekjesmonster . . . . .	23
4.2	Thors hamer . . . . .	27
4.3	Raak de muren niet aan! . . . . .	28
4.4	Geld uit de lucht . . . . .	30
4.5	Vallende bloempotten . . . . .	31
4.6	Geheimtaal . . . . .	31
4.7	Vlucht door de stad . . . . .	33
4.8	O x O . . . . .	34
<b>5</b>	<b>Zit je even vast?</b>	<b>37</b>
<b>6</b>	<b>Links</b>	<b>39</b>
6.1	CoderDojo's . . . . .	39
6.2	Python . . . . .	39
6.3	IDEs en editors . . . . .	39



Python is een leuke programmeertaal met vele mogelijkheden (zie [Waarom Python?](#)).

In [Starten met Python](#) maak je kennis met Python. Eens je de basis onder de knie hebt, ben je klaar om bv. spelletjes te maken met Python (zie [Projecten](#)).



### 1.1 Waarom Python?

- Elegante en compacte taal
- Uitgebreide (standaard)bibliotheken (*batteries included*)
- Interactief gebruik mogelijk (bv. `rekenmachine`, `IPython`)
- Uitstekende documentatie
- Open source
- Schaalt goed: *start small, grow big*
- Meest populaire taal volgens laatste IEEE en PYPL rangschikking, en op 4e plaats in TIOBE index
- Gebruikt in vele verschillende toepassingsdomeinen: web, wetenschappelijke wereld, desktop, onderwijs, software ontwikkeling, ...
- Steeds vaker gebruikt als eerste taal voor beginners
- *Fun*

### 1.2 Installatie

- *Windows*
- *Mac OS X*

#### 1.2.1 Windows

Installatie-instructies voor `Python` onder Windows.

### Python

Download en gebruik de [Windows x86-64 executable installer](#) voor Python 3.7.2 64-bit (zie [Python Releases for Windows](#)).

In de installer, vink “Add Python 3.7 to PATH” aan.

---

#### Test

Open een Windows commando-venster met Win-R, `cmd`, en controleer de Python-versie:

```
> python -V
Python 3.7.2
```

---

#### Test

Open IDLE vanuit het Start-menu: “IDLE (Python 3.7 64 bit)”. Dit toont de “Python 3.7.2 Shell”.

Meer uitleg is te vinden in [Installatie-instructies](#) [[Installatie Python.docx](#)].

### pygame

Maak een [virtuele omgeving](#) aan:

```
> mkdir CoderDojo
CoderDojo> cd CoderDojo
CoderDojo> python -m venv venv37
```

Gebruik de aangemaakte virtuele omgeving:

```
CoderDojo> venv37\Scripts\activate
(venv37) CoderDojo>
```

Update pip:

```
(venv37) CoderDojo> python -m pip install --upgrade pip
```

Installeer [pygame](#) m.b.v. `pip`:

```
(venv37) CoderDojo> pip install -U pygame
```

---

#### Test

```
(venv37) CoderDojo> python -m pygame.examples.aliens
```

Meer uitleg is te vinden in [Pygame Installation](#).

### IPython

Installeer [IPython](#) (onderdeel van [Jupyter](#)) m.b.v. `pip`:



```
(venv37) CoderDojo> pip install jupyter
```

## Test

```
(venv37) CoderDojo> ipython -V
7.3.0
```

```
(venv37) CoderDojo> ipython
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit_
↪ (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.3.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("Hi!")
Hi!

In [2]: 2 + 2
Out[2]: 4

In [3]:
```

Sluit IPython met Ctrl-D.

## Kivy

Kivy heb je enkel nodig voor de *Extra oefeningen*.

Installeer Kivy m.b.v. pip:

```
(venv37) CoderDojo> python -m pip install --upgrade pip wheel setuptools
(venv37) CoderDojo> python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2_
↪ kivy.deps.glew
(venv37) CoderDojo> python -m pip install kivy.deps.gstreamer
(venv37) CoderDojo> python -m pip install kivy
```

## Test

```
(venv37) CoderDojo> python
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit_
↪ (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import kivy
[INFO ] [Logger      ] Record log in C:\Users\P\.kivy\logs\kivy_19-03-18_1.txt
[INFO ] [Kivy         ] v1.10.1
[INFO ] [Python       ] v3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC_
↪ v.1916 64 bit (AMD64)]
```

Meer uitleg is te vinden in [Kivy Installation on Windows](#).

### 1.2.2 Mac OS X

Installatie-instructies voor Python onder Mac OS X.

#### Python

Download en gebruik de [macOS 64-bit installer](#) voor Python 3.7.2 64 bit (zie [Python Releases for Mac OS X](#) ).

---

#### Test

Open een terminal-venster met `cmd + spatie`, `Terminal`, en controleer de Python-versie:

```
~ user$ python3 -V
Python 3.7.2
```

---

#### pygame

Maak een [virtuele omgeving](#) aan:

```
~ user$ mkdir CoderDojo
~ user$ cd CoderDojo
CoderDojo user$ python3 -m venv venv37
```

Gebruik de aangemaakte virtuele omgeving:

```
CoderDojo user$ source venv37/bin/activate
(venv37) CoderDojo user$
```

Update pip:

```
(venv37) CoderDojo user$ pip install --upgrade pip
```

Installeer [pygame](#) m.b.v. `pip`:

```
(venv37) CoderDojo user$ pip install -U pygame
```

---

#### Test

```
(venv37) CoderDojo user$ python -m pygame.examples.aliens
```

Meer uitleg is te vinden in [Pygame Installation](#).

#### IPython

Installeer [IPython](#) (onderdeel van [Jupyter](#)) m.b.v. `pip`:

```
(venv37) CoderDojo user$ pip install jupyter
```

---

#### Test

```
(venv37) CoderDojo user$ ipython -V
7.3.0
```

```
(venv37) CoderDojo user$ ipython
```

```
Python 3.7.2 (...)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.3.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: print("Hi!")
Hi!
```

```
In [2]: 2 + 2
Out[2]: 4
```

```
In [3]:
```

Sluit IPython met Ctrl-D, y.

## Kivy

Kivy heb je enkel nodig voor de *Extra oefeningen*.

Installeer Homebrew:

```
~ user$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/
↪install/master/install)"
```

Gebruik Homebrew voor de installatie van de bibliotheken die Kivy nodig heeft:

```
~ user$ brew install pkg-config sdl2 sdl2_image sdl2_ttf sdl2_mixer gstreamer
```

In je virtuele omgeving, installeer Cython en Kivy m.b.v. pip:

```
~ user$ cd CoderDojo
CoderDojo user$ source venv37/bin/activate
(venv37) CoderDojo user$ pip install -U Cython
(venv37) CoderDojo user$ pip install kivy
```

## Test

```
(venv37) CoderDojo user$ python
```

```
Python 3.7.2 (...)
[Clang ...] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import kivy
[INFO ] [Logger      ] Record log in /Users/user/.kivy/logs/...txt
[INFO ] [Kivy        ] v...
[INFO ] [Python       ] v3.7.2 (...)
[Clang ...]
```

Sluit met Ctrl-D.

Meer uitleg is te vinden in [Kivy Installation on OS X](#).

## 1.3 Kennismaking met Python

Niveau 

Bron Aagje Reynders

### 1.3.1 Wat zal je leren?

- `print`
- Gebruik van variabelen
- `input` om tekst in te lezen
- `if/elif/else`
- `for`
- Gebruik van lijsten
- `def`
- `random`

### 1.3.2 Wat heb je nodig?

- Een werkende Python-installatie (zie *Installatie*)
- Interactieve Python-prompt, bij voorkeur IPython (zie *Installatie*)
- Een Python-editor, bv. IDLE. IDLE kan je openen vanuit het Start-menu: “IDLE (Python 3.7 64 bit)”. Dit toont de “Python 3.7.0 Shell”. De editor open je door een nieuw bestand aan te maken via *File* → *New File*.

### 1.3.3 Aan de slag



**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!

Zie [Kennismaking met Python \[Kennismaking met Python.pdf\]](#) (Aagje Reynders).

Inhoudstafel:

1. Toon tekst op het scherm m.b.v. `print`
2. Toon je naam op het scherm waarbij je een variabele gebruikt
3. Vraag leeftijd van gebruiker m.b.v. `input`
4. Vergelijk twee getallen en toon welke het grootste is
5. Tel van 1 t.e.m. 100
6. Beheer lijst van deelnemers
7. Teken ovals op het scherm

### 1.3.4 Extra

1. Gebruik willekeurige kleur voor ovals
2. Maak een gokspelletje waarbij je een willekeurig getal probeert te raden
3. Maak een rekenmachientje voor twee getallen

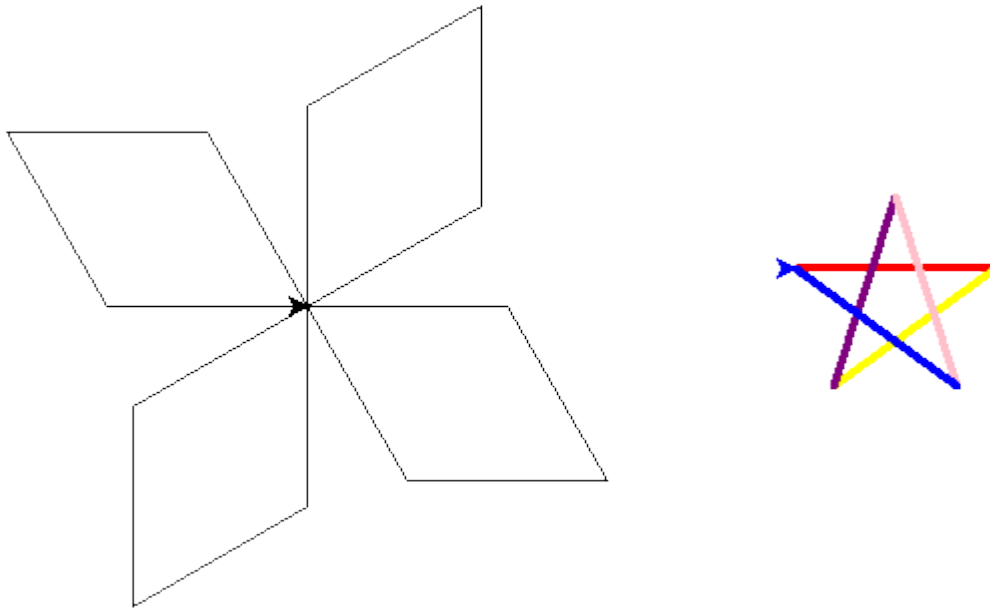
## 1.4 Schildpad

Niveau 

Bronnen [Codingkids](#), Johan Vereecken

### 1.4.1 Doel

Maken van diverse figuren in Python door het bewegen van de Python schildpad (*Turtle*) over het scherm.



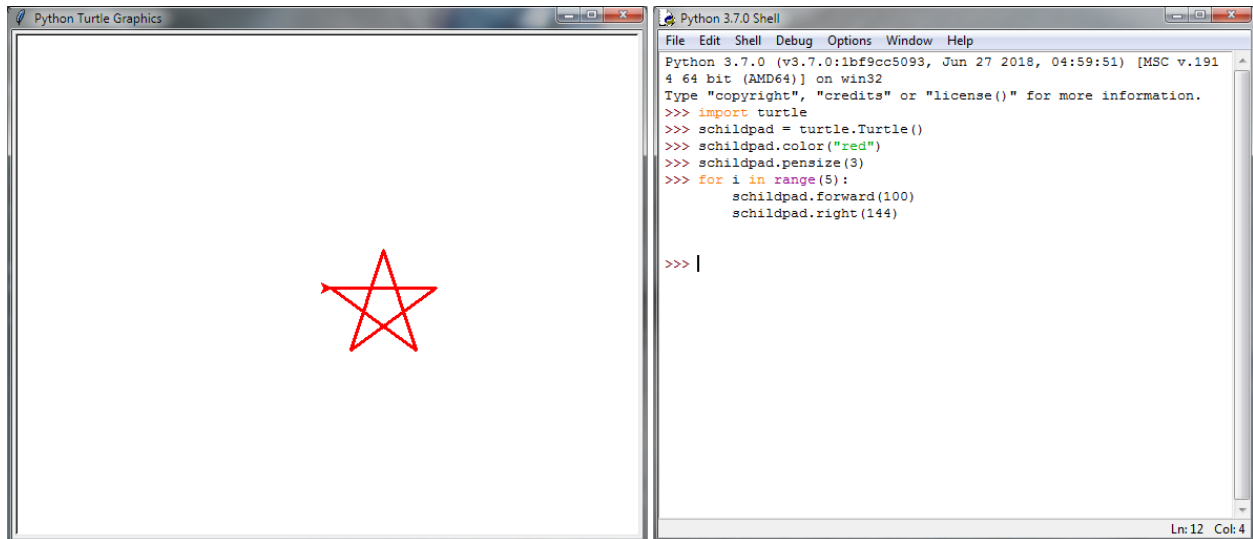
### 1.4.2 Wat heb je nodig?

- Een werkende Python-installatie (zie *Installatie*)
- Een Python-editor. In deze opdrachten gebruiken we **IDLE**. IDLE kan je openen vanuit het Start-menu: “IDLE (Python 3.7 64 bit)”. Dit toont de “Python 3.7.0 Shell”. De editor open je door een nieuw bestand aan te maken via *File* → *New File* (zie verder).

### 1.4.3 Wat zal je leren?

- Gebruik van de Python-omgeving IDLE
- Gebruik van variabelen
- `for`, `while`
- Mogelijke opdrachten (*commands*) voor de schildpad
- Verband tussen Scratch en Python

### 1.4.4 Aan de slag



Zie Programmeren met Python op Codingkids.



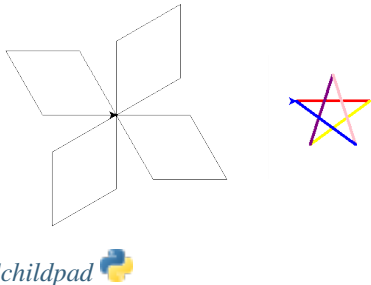

### 1.4.5 Extra

Zie Python opdrachtjes [Python\_beginner\_opdrachtjes\_Turtle.pdf] (Johan Vereecken).

### 1.4.6 Nuttige links

- Turtle graphics
- Python Turtle Graphics, How to Think Like a Computer Scientist

Is dit je eerste keer? Start dan hier. Installeer Python en leer Python stap voor stap.

<p><i>Installatie</i></p>	 <p><i>Kennismaking met Python</i> </p>	 <p><i>Schildpad</i> </p>
---------------------------	---	--





## HOOFDSTUK 2

---

### Boeken

---

Wil je meer Python leren door een goed boek te lezen? Zie dan bv. de volgende boeken:

- “Python voor kids, beginnen met programmeren”, Johan Vereecken, 2015 (gratis voor niet-commerciële doeleinden). Nederlandstalig boek over de beginselen van het programmeren in Python, specifiek voor kinderen, geschreven door Johan Vereecken (lead-coach CoderDojo Mol). Bruikbaar voor zelfstudie.
- “Programmeren voor kinderen - Python”, Carol Vorderman, 2018. Op de coderdojo is er een exemplaar dat je vrij kan gebruiken. Je kan het ook vinden in bibliotheek De Krook of Brugse Poort.



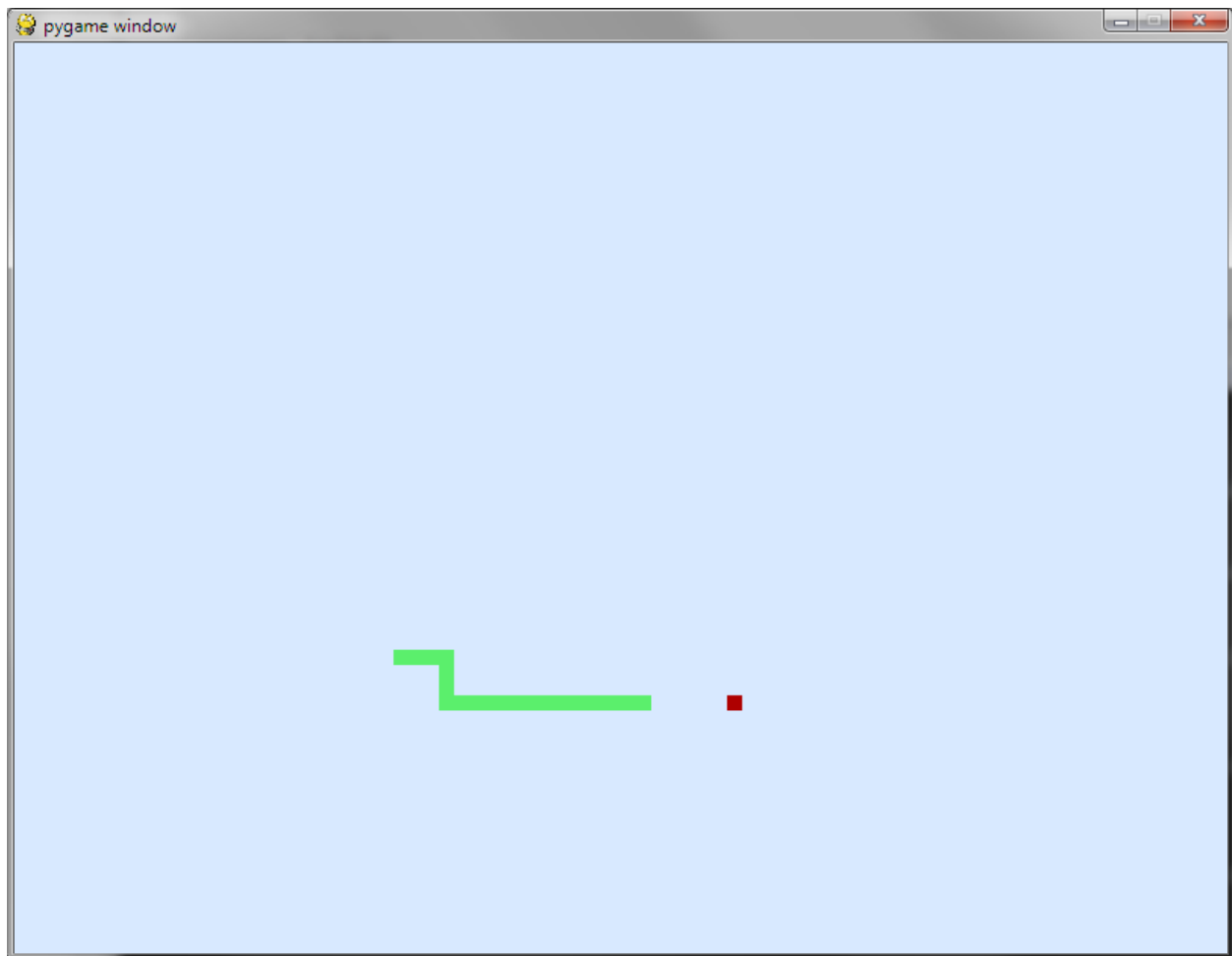
### 3.1 Snake

**Niveau** 

**Bron** Aagje Reynders

#### 3.1.1 Doel

Snake in Python.



Snake is een spelletje waar een slang moet proberen appels te eten. Deze appels komen willekeurig op het scherm. Met de pijltjestoetsen probeer je met de slang naar de appel te kruipen zonder tegen de randen of jezelf te botsen. Elke keer dat je een appel eet wordt je slang langer.

### 3.1.2 Wat zal je leren?

- pygame concepten: *events*, werken met kleuren, tekenen op het scherm, ...
- spellogica

### 3.1.3 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)
- Basiskennis Python (zie *Starten met Python*)

### 3.1.4 Aan de slag



Volg de stappen zoals beschreven in [Games ontwikkelen \[pygame cursus.pdf\]](#) (Aagje Reynders).

Overzicht stappen:

1. Werken met events
2. Teken op het scherm: slang = rechthoek
3. Bewegen van slang (rechthoek)
4. Randen herkennen
5. Plaats appel(s)
6. Laat slang langer worden

### 3.1.5 Mogelijke oplossing

```
import pygame
import random

pygame.init()

gameDisplay = pygame.display.set_mode((800, 600))

FRAME_RATE = 15 # FPS

clock = pygame.time.Clock()
font = pygame.font.SysFont(None, 25)
groen = (91, 239, 108)
rood = (174, 0, 0)
achtergrondkleur = (216, 233, 255)

def toon_bericht(bericht, kleur=(51, 40, 255)):
    tekst = font.render(bericht, True, kleur)
    tekst_rect = tekst.get_rect()
    tekst_rect.center = (400, 300)
    gameDisplay.blit(tekst, tekst_rect)
    pygame.display.update()
```

(Vervolgt op volgende pagina)

```
def teken_slang(slang_lichaam):
    for x, y in slang_lichaam:
        pygame.draw.rect(gameDisplay, groen, [x, y, 10, 10])

x = 400
y = 300
slang_lengte = 1
slang_lichaam = []
richting = 'RECHTS'
game_exit = False
game_over = False
game_pauzed = False

rand = 5
appel_x = random.randint(0 + rand, 79 - rand) * 10
appel_y = random.randint(0 + rand, 59 - rand) * 10

while not game_exit:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT and richting != 'RECHTS':
                richting = 'LINKS'
            elif event.key == pygame.K_RIGHT and richting != 'LINKS':
                richting = 'RECHTS'
            elif event.key == pygame.K_UP and richting != 'ONDER':
                richting = 'BOVEN'
            elif event.key == pygame.K_DOWN and richting != 'BOVEN':
                richting = 'ONDER'
            elif event.key == pygame.K_q:
                game_exit = True
            elif event.key == pygame.K_SPACE:
                game_pauzed = not game_pauzed
                if game_pauzed:
                    toon_bericht("PAUZE")

    if game_pauzed:
        continue

    if richting == 'LINKS':
        x -= 10
    elif richting == 'RECHTS':
        x += 10
    elif richting == 'BOVEN':
        y -= 10
    elif richting == 'ONDER':
        y += 10

    slang_hoofd = (x, y)
    slang_lichaam.append(slang_hoofd)
    if len(slang_lichaam) > slang_lengte:
        del slang_lichaam[0]
```

(Vervolgt op volgende pagina)

(Vervolgd van vorige pagina)

```

if (not 0 < x < 800) or (not 0 < y < 600):
    game_over = True

if slang_hoofd in slang_lichaam[:-1]:
    game_over = True

if x == appel_x and y == appel_y:
    appel_x = random.randint(0 + rand, 79 - rand) * 10
    appel_y = random.randint(0 + rand, 59 - rand) * 10
    slang_lengte += 1

gameDisplay.fill(achtergrondkleur)
pygame.draw.rect(gameDisplay, rood, [appel_x, appel_y, 10, 10])
teken_slang(slang_lichaam)
pygame.display.update()

clock.tick(FRAME_RATE)

if game_over:
    toon_bericht("GAME OVER! Druk 'r' op opnieuw te proberen, "
                "'c' om verder te gaan, of 'q' om af te sluiten")
while game_over:
    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_q: # 'a' op AZERTY toetsenbord...
                game_exit = True
                game_over = False
            elif event.key == pygame.K_c: # continue
                game_over = False
                x = 400
                y = 300
            elif event.key == pygame.K_r: # restart
                game_over = False
                x = 400
                y = 300
                slang_lengte = 1
                slang_lichaam = []

pygame.quit()

```

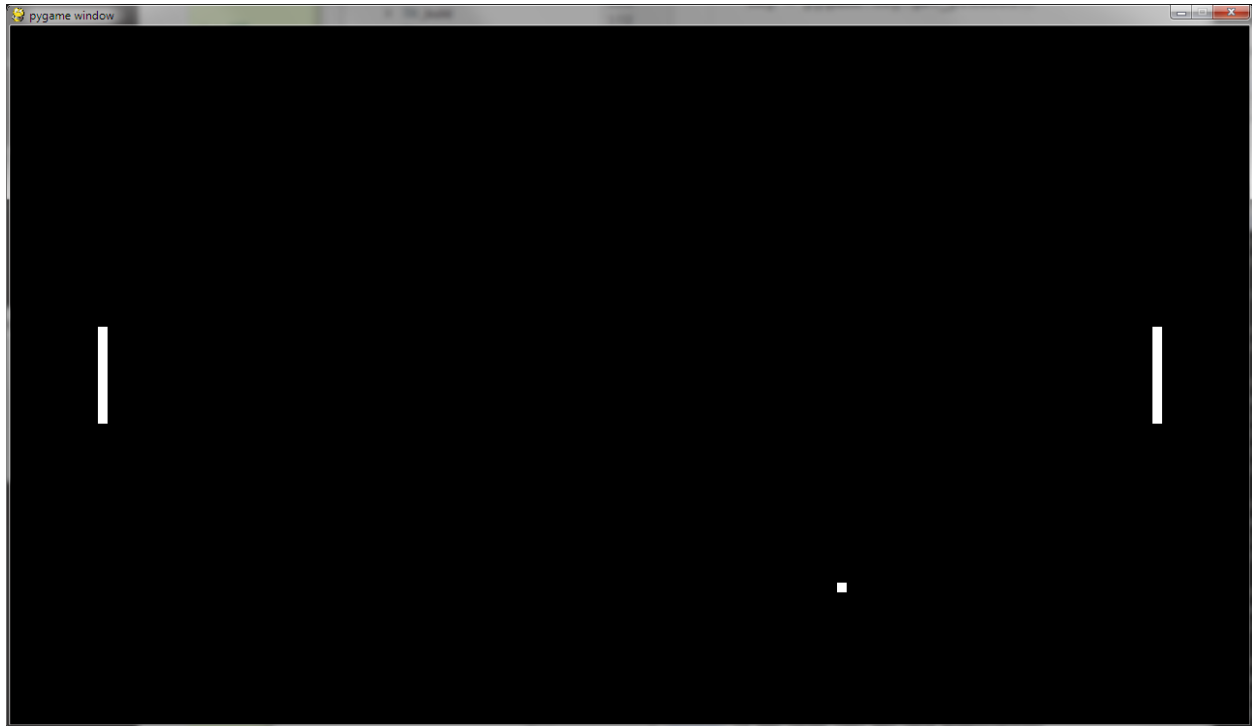
## 3.2 Pong

Niveau 

Bron Raf Schoenmaekers

### 3.2.1 Doel

Pong in Python.



Pong is ...

### 3.2.2 Wat zal je leren?

...

### 3.2.3 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie [Installatie](#))
- Basiskennis Python (zie [Starten met Python](#))
- Basiskennis pygame (zie [Snake](#))

### 3.2.4 Aan de slag

Stappen: ...

### 3.2.5 Mogelijke oplossing

```
import sys
import pygame

size = width, height = 1280, 720

black = (0, 0, 0)
white = (255, 255, 255)
```

(Vervolgt op volgende pagina)



(Vervolg van vorige pagina)

```

ballSpeed = [2, 2]
ballRect = pygame.Rect(width / 2, height / 2, 10, 10)

racketOffset = 100
racketHeight = 100
racketWidth = 10

player1Rect = pygame.Rect(racketOffset - racketWidth,
                           (height - racketHeight) / 2,
                           racketWidth,
                           racketHeight)
player2Rect = pygame.Rect(width - racketOffset,
                           (height - racketHeight) / 2,
                           racketWidth,
                           racketHeight)

player1Speed = player2Speed = 0.0
playerAcceleration = 1.0
player1Score = player2Score = 0

# start game
pygame.init()
screen = pygame.display.set_mode(size)

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()

    # move ball
    ballRect = ballRect.move(*ballSpeed)

    if ballRect.left < 0 or ballRect.right > width:
        if ballRect.left < 0:
            player2Score += 1
        else:
            player1Score += 1
        print(f' SCORE {player1Score} - {player2Score} ')
        # reset ball to center of screen
        ballRect.x = width / 2
        ballRect.y = height / 2

    if ballRect.top < 0 or ballRect.bottom > height:
        ballSpeed[1] = -ballSpeed[1]

    # move players
    player1Speed *= 0.9
    player2Speed *= 0.9

    key = pygame.key.get_pressed()

    if key[pygame.K_x]:
        player1Speed += playerAcceleration
    if key[pygame.K_s]:
        player1Speed -= playerAcceleration
    if key[pygame.K_UP]:
        player2Speed -= playerAcceleration

```

(Vervolgt op volgende pagina)

(Vervolgd van vorige pagina)

```
if key[pygame.K_DOWN]:
    player2Speed += playerAcceleration

player1Rect.y += player1Speed
player2Rect.y += player2Speed

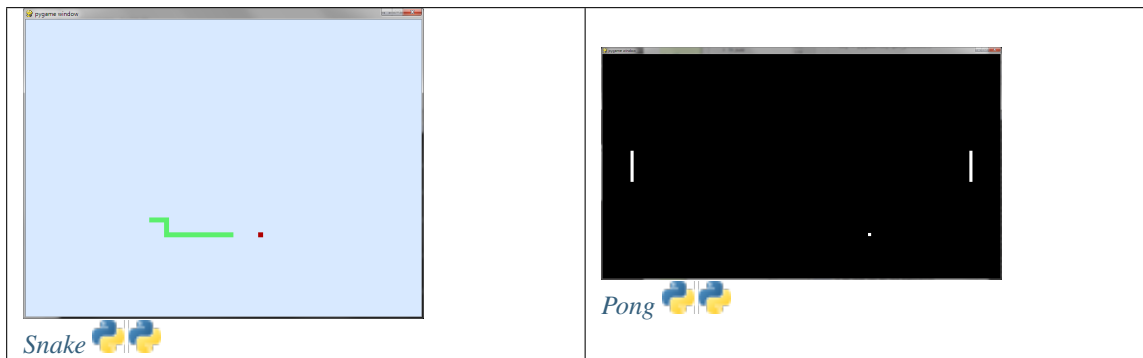
if ballRect.colliderect(player1Rect):
    ballSpeed[0] = abs(ballSpeed[0])
if ballRect.colliderect(player2Rect):
    ballSpeed[0] = -abs(ballSpeed[0])

# start drawing
screen.fill(black)
pygame.draw.rect(screen, white, ballRect)
pygame.draw.rect(screen, white, player1Rect)
pygame.draw.rect(screen, white, player2Rect)

# wait
pygame.display.update()
pygame.time.delay(10)
```

Heb je de basis van Python onder de knie? Hier vind je verschillende leuke projecten waarin je stap voor stap leert hoe je spelletjes kan maken met Python en pygame.

Je mag natuurlijk ook een eigen project bedenken. Je kan daarbij steeds hulp vragen aan een coach.



## HOOFDSTUK 4

---

Extra oefeningen

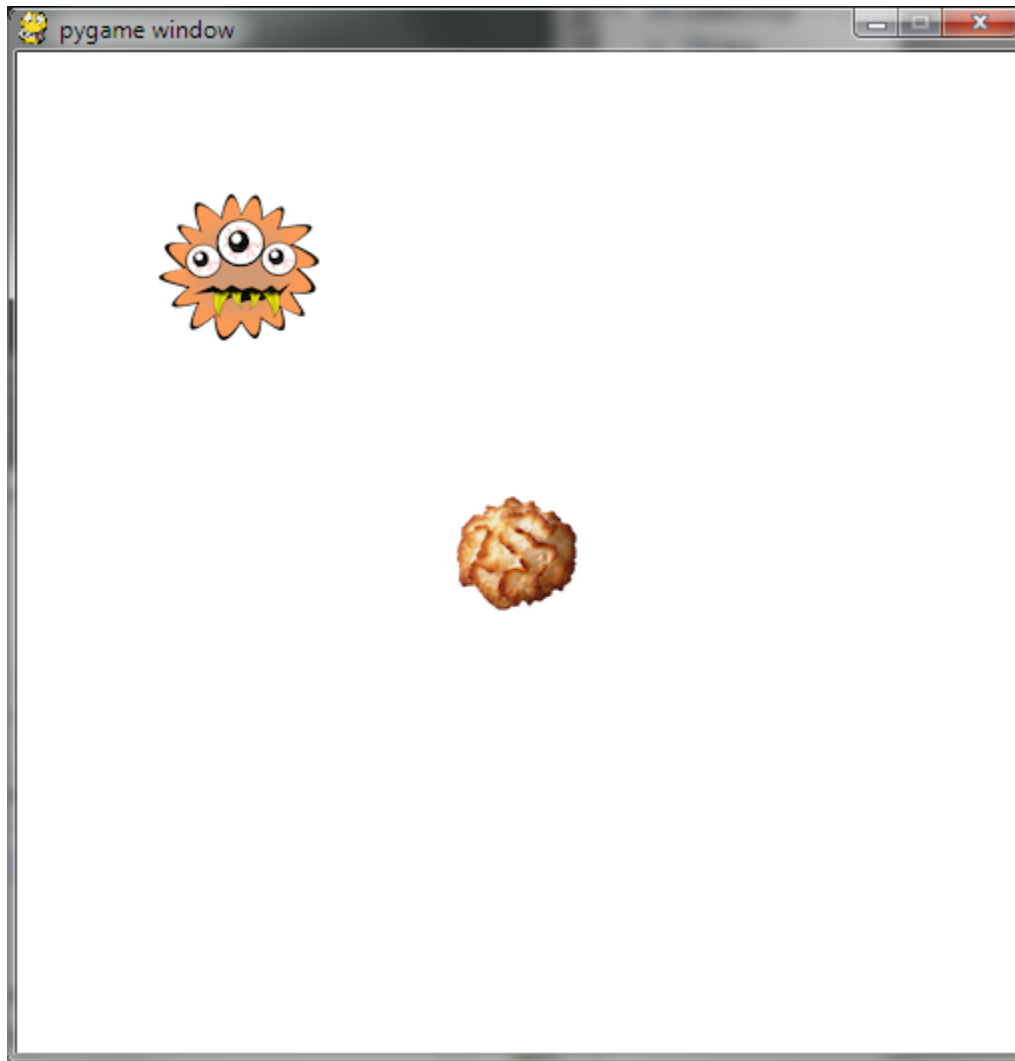
---

### 4.1 Koekjesmonster

Niveau 

Bron [coderdojo-gent-pygame](#) van Yannick Clybouw

### 4.1.1 Doel



Programmeer het koekjesmonster zó dat ze haar weg vindt naar de koekje(s) en deze kan opeten!

### 4.1.2 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie [Installatie](#))
- De Python code voor deze oefening: [download deze bestanden](#) en plaats deze op je computer in dezelfde map waar je ook je oplossing zal aanmaken en bewaren

### 4.1.3 Oefening 1A

Er verschijnt een monster en één koekje. Vul de programmalus in zó dat het monster op weg gaat naar het koekje om het op te eten.

```
from oefening_01 import OpdrachtA
```

(Vervolgt op volgende pagina)

(Vervolgd van vorige pagina)

```

opdracht = OpdrachtA(500, 500)
monster = opdracht.monster
koekje = opdracht.koekje

def programmalus():
    """Beweeg het monster naar het koekje."""
    # -----#
    #                                           #
    #   Vul hier je programmalus in!           #
    #                                           #
    # -----#

opdracht.start(programmalus)

```

De positie op het scherm van het koekje is:

*(koekje.x, koekje.y)*

De positie op het scherm van het monster is:

*(monster.x, monster.y)*

Hierbij komt  $(0, 0)$  overeen met links-boven op het scherm.

Je kan het monster bewegen met de volgende acties:

- ga één stap naar links: *monster.stap\_links()*
- ga één stap naar rechts: *monster.stap\_rechts()*
- ga één stap naar omhoog: *monster.stap\_omhoog()*
- ga één stap naar omlaag: *monster.stap\_omlaag()*

## Scratch vs. Python

Scratch	Python
als ... dan	if ... :
anders	else:
anders, als ... dan	if ... : elif ...: else ...:

### 4.1.4 Oefening 1B

Er verschijnt een monster en een koekje. Elke keer als je het spelletje start, zal het koekje op een *willekeurige* plaats verschijnen.

Vul de programmalus in zó dat het monster op weg gaat naar het koekje om het op te eten.

```

from oefening_01 import OpdrachtA

opdracht = OpdrachtB(500, 500)
monster = opdracht.monster
koekje = opdracht.koekje

def programmalus():

```

(Vervolgt op volgende pagina)

(Vervolgd van vorige pagina)

```

"""Beweeg het monster naar het koekje."""
# ----- #
#                                     #
#   Vul hier je programmalus in!     #
#                                     #
# ----- #

opdracht.start(programmalus)

```

### 4.1.5 Oefening 1C

Er verschijnt een monster, samen met *verschillende* koekjes op *willekeurige* plaatsen.

Vul de programmalus in zó dat het monster één voor één op weg gaat naar de koekjes om ze op te eten.

```

from oefening_01 import OpdrachtC

opdracht = OpdrachtC(500, 500)
monster = opdracht.monster
koekjesgroep = opdracht.koekjesgroep
# De koekjesgroep heeft verschillende sprites.
# Telkens je een koekje opeet, verdwijnt die uit de groep.

def programmalus():
    """Beweeg het monster naar de koekjes, één voor één."""
    # ----- #
    #                                     #
    #   Vul hier je programmalus in!     #
    #                                     #
    # ----- #
    yield

opdracht.start(programmalus)

```

Alle koekjes zitten in *koekjesgroep*. Je kan alle koekjes overlopen met een lus.

Telkens het monster een koekje opeet, verdwijnt het koekje uit de groep. Je kan testen of een koekje nog in de groep zit met:

```
koekje in koekjesgroep
```

Merk op: je dient `yield` te gebruiken binnen in je lus.

### Scratch vs. Python

Scratch	Python
herhaal (voor elke X uit de GROEP)	for X in GROEP
omgekeerde van “herhaal tot”	while VOORWAARDE

### Lussen in Python

break = verlaat de lus

continue = sla alles over tot de volgende herhaling van de lus

### 4.1.6 Oefening 1C extra

Als oefening 1C gelukt is, kun je als extra uitdaging ervoor zorgen dat het monster telkens het dichtste koekje opeet voordat ze naar het volgende koekje gaat.

De afstand tussen het monster en een koekje kan je berekenen met het volgende stukje code:

```
import math

def afstand(monster, koekje):
    return math.sqrt((monster.x - koekje.x) ** 2 +
                     (monster.y - koekje.y) ** 2)
```

Oefening:

```
import math

from oefening_01 import OpdrachtC

opdracht = OpdrachtC(500, 500)
monster = opdracht.monster
koekjesgroep = opdracht.koekjesgroep

def afstand(monster, koekje):
    return math.sqrt((monster.x - koekje.x) ** 2 +
                     (monster.y - koekje.y) ** 2)

def programmalus():
    """Beweeg het monster naar de koekjes,
    te beginnen met het dichtsbijzijnde koekje."""
    # ----- #
    #                                           #
    #   Vul hier je programmalus in!           #
    #                                           #
    # ----- #
    yield

opdracht.start(programmalus)
```

## 4.2 Thors hamer

Niveau 

Bron [pythoncourse](#) van Yannick Clybouw

### 4.2.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)
- Kivy (zie *Installatie*)
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.2.2 Doel

## Thor's hammer



Oh nee! Loki heeft Thor's hamer (Mjolnir) gestolen! Gelukkig heeft Thor een GPS tracker aan zijn hamer hangen zodat hij altijd weet waar die zich bevindt.



Thor is echter niet zo goed in navigeren en heeft jouw hulp nodig. Kun jij Thor vertellen welke richting hij uit moet?

### 4.2.3 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start oefening met:

```
(venv37) CoderDojo> python main.py 1
```

### 4.2.4 Oplossing

[Voorbeeldoplossingen](#) [pythoncourse](#)

## 4.3 Raak de muren niet aan!

Niveau 

Bron [pythoncourse](#) van Yannick Clybouw

### 4.3.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)

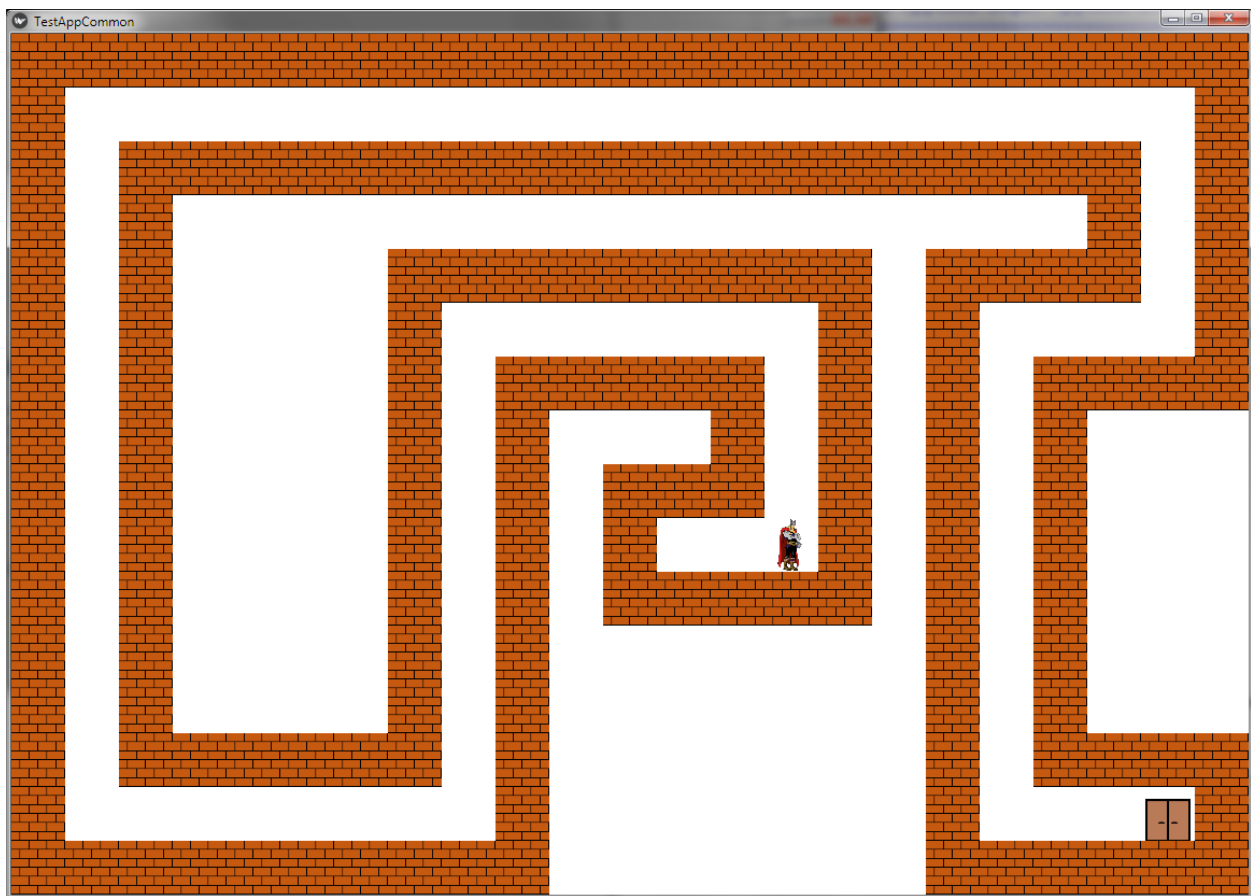


- Kivy (zie *Installatie*)
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.3.2 Doel

## Raak de muren niet aan!

Help, Thor zit opgesloten in een behekst kasteel. Hij wil de smalle gang volgen naar de uitgang, maar alle muren zijn giftig en mag hij niet aanraken.



### 4.3.3 Wat zal je leren?

- voorwaarden
- vergelijken
- booleaanse operatoren

### 4.3.4 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 2
```

### 4.3.5 Oplossing

Voorbeeldoplossingen [pythoncourse](#)

## 4.4 Geld uit de lucht

Niveau 

Bron [pythoncourse](#) van Yannick Clybouw

### 4.4.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie [Installatie](#))
- Kivy (zie [Installatie](#))
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.4.2 Doel

## Geld uit de lucht

Je wandelt op straat en er valt plots geld uit de lucht.  
Probeer alle vallende munten te vangen.

### 4.4.3 Wat zal je leren?

- lijsten
- elementen uit lijsten nemen

### 4.4.4 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 3
```

## 4.4.5 Oplossing

Voorbeeldoplossingen pythoncourse

## 4.5 Vallende bloempotten

Niveau 

Bron pythoncourse van Yannick Clybouw

### 4.5.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie [Installatie](#))
- Kivy (zie [Installatie](#))
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.5.2 Doel

## Vallende bloempotten

Je wandelt op straat en er is plots een aardbeving. Alle bloempotten van het flatgebouw achter je vallen naar beneden. Probeer ze te ontwijken.

### 4.5.3 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 4
```

### 4.5.4 Oplossing

Voorbeeldoplossingen pythoncourse

## 4.6 Geheimtaal

Niveau 

Bron pythoncourse van Yannick Clybouw

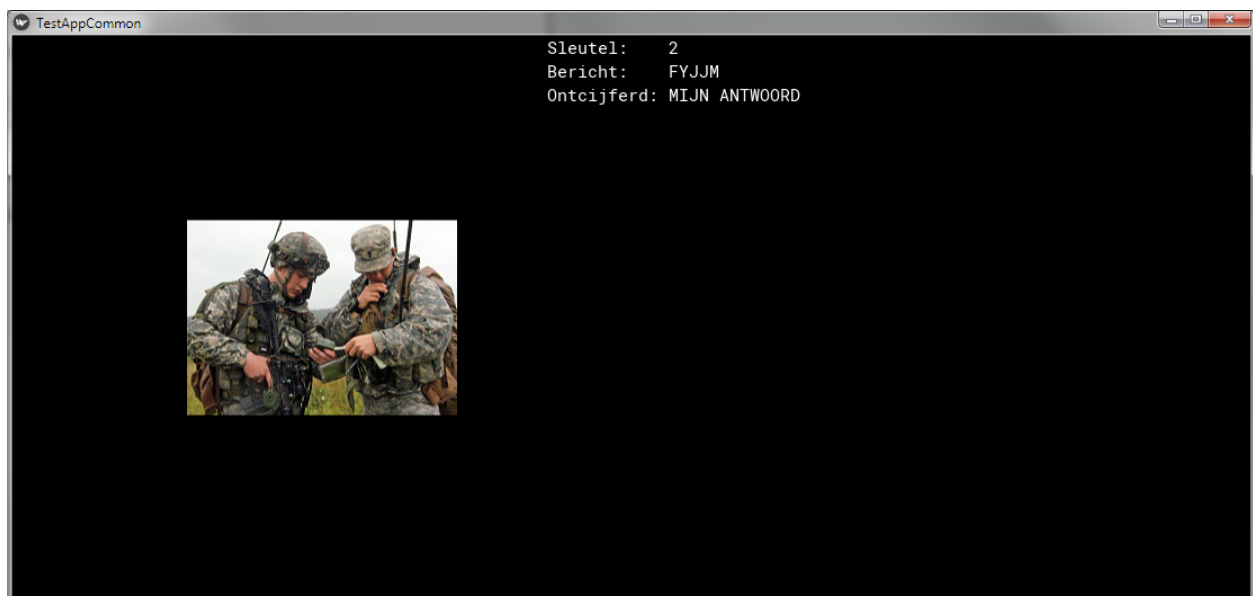
### 4.6.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)
- Kivy (zie *Installatie*)
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.6.2 Doel

## Geheimtaal

De legerofficier zendt je boodschappen in een geheimtaal. Ontcijfer ze!



### 4.6.3 Wat zal je leren?

- datatypes string, integer
- [ASCII](#)
- for lus over string
- concatenatie van strings

### 4.6.4 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 5
```

## 4.6.5 Oplossing

Voorbeeldoplossingen pythoncourse

## 4.7 Vlucht door de stad

Niveau 

Bron pythoncourse van Yannick Clybouw

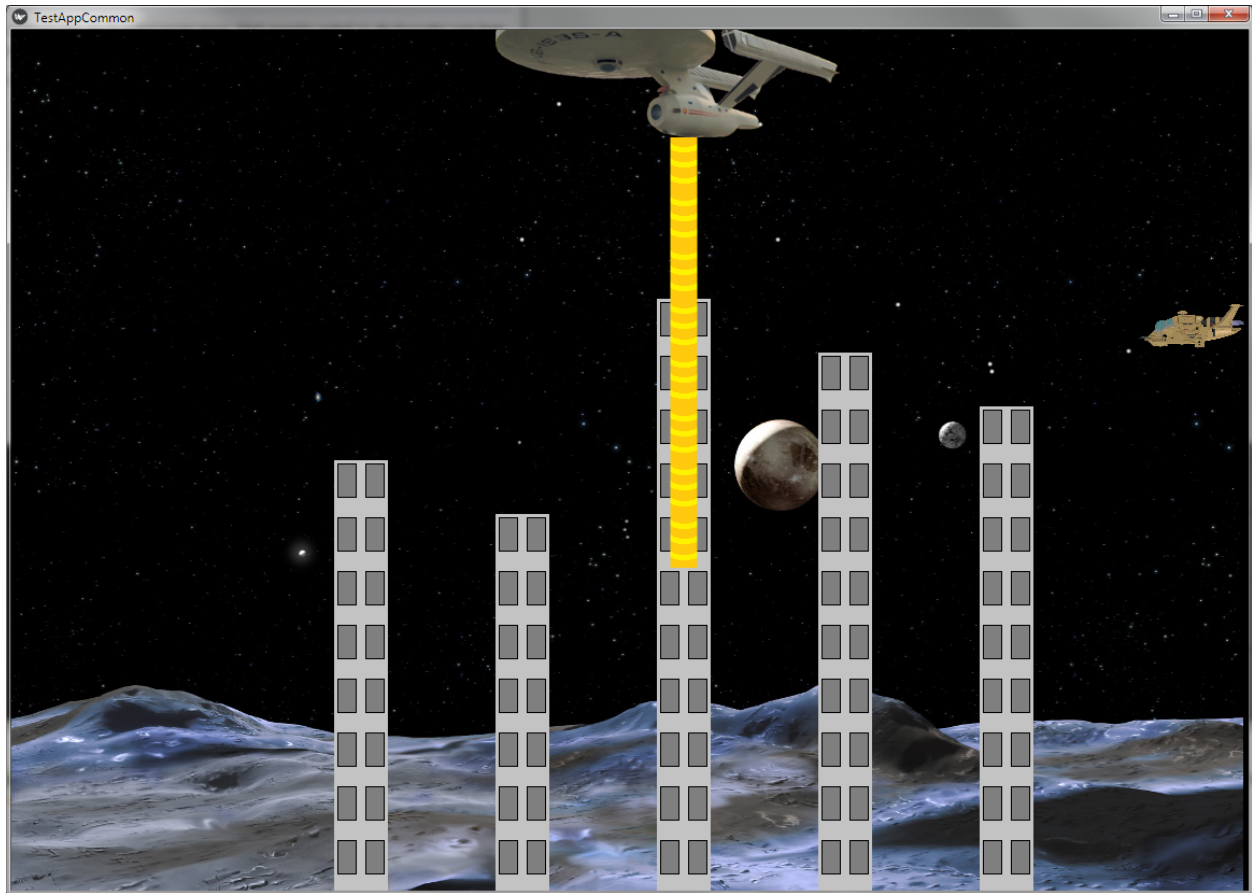
### 4.7.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)
- Kivy (zie *Installatie*)
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.7.2 Doel

## Vlucht door de stad

Je ben met je ruimteschip aangekomen op een onbekende planeet op zoek naar overlevenden van een kernoorlog. Een van je shuttles is op verkenning doorheen een stad en vliegt telkens lager en lager om overlevenden te vinden. Bij elke passage dien je vanaf het mothership het hoogste gebouw te bombarderen zodat je shuttle een vrije doorgang heeft.



### 4.7.3 Wat zal je leren?

- lijsten
- vergelijken

### 4.7.4 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 6
```

### 4.7.5 Oplossing

Voorbeeldoplossingen [pythoncourse](#)

## 4.8 O x O

Niveau 

Bron [pythoncourse](#) van Yannick Clybouw

### 4.8.1 Wat heb je nodig?

- Een werkende Python-installatie (inclusief pygame) (zie *Installatie*)
- Kivy (zie *Installatie*)
- De Python code voor deze oefening: [download](#) deze bestanden en plaats deze op je computer in een map naar keuze

### 4.8.2 Doel

# O X O

Maak een programma om te winnen met O X O. Jij speelt met O, de tegenstander met X. Jij mag altijd beginnen. De winnaar is de eerste die drie op een rij heeft, zowel verticaal, horizontaal als diagonaal.

### 4.8.3 Wat zal je leren?

- geneste lijsten
- spellogica

### 4.8.4 Oefening

Ga naar de map waar je de oefenbestanden bewaarde, en start de oefening met:

```
(venv37) CoderDojo> python main.py 7
```

### 4.8.5 Oplossing

Voorbeeldoplossingen [pythoncourse](#)

Wil je nog meer oefenen met Python? Zie dan de volgende oefeningen:

- *Koekjesmonster* 
- *Thors hamer* 
- *Raak de muren niet aan!* 
- *Geld uit de lucht* 
- *Vallende bloempotten* 
- *Geheimtaal*  
- *Vlucht door de stad*  
- *O x O*  





## HOOFDSTUK 5

---

### Zit je even vast?

---

Het is normaal dat je af en toe even vast zit. Programmeren is ook leren oplossingen vinden. Enkele tips voor wanneer je vast zit:

- Zoek uit wat het probleem precies is:
  - Wat verwacht je te zien? Waarom zie je dat niet?
  - Welke waarden heeft een bepaalde variabele (print deze eens)? Hoort dat zo te zijn?
  - Kan je een kleinere stukje code schrijven dat wel werkt?
  - Wat gebeurt als je je code een beetje aanpast? Werkt het dan nog?
- Als je een bepaalde foutmelding krijgt, zoek dan eens op internet (bv. met [Duck Duck Go](#)) om te achterhalen wat het probleem precies is en of je er oplossingen voor vindt.
- Vraag hulp aan je vrienden.
- Vraag hulp aan een coach.
- Neem een pauze.



### 6.1 CoderDojo's

- [CoderDojo België](#)
- [CoderDojo Gent](#)

### 6.2 Python

- [The Zen of Python](#)
- [The Python Tutorial](#)
- [Officiële website Python](#)

### 6.3 IDEs en editors

- [IDLE](#) (Python's Integrated Development and Learning Environment) is een goede editor om kennis te maken met Python.
- [Mu editor](#)
- [PyCharm](#) is een professionele ontwikkelomgeving die je kan gebruiken eens je Python al wat beter kent.

Laatste wijziging op 05 april 2019.