

Coderingen per project

Hier vind je de complete Python-codering voor elk project in dit boek, behalve voor de hacks en tweaks. Werken je projecten niet goed, controleer je scripts dan zorgvuldig aan de hand van de coderingen hieronder.

Dierenquiz (blz. 36)

```
def check_raad(raad, antwoord):
    global score
    nog_radend = True
    poging = 0
    while nog_radend and poging < 3:
        if raad.lower() == antwoord.lower():
            print('Correct antwoord')
            score = score + 1
            nog_radend = False
        else:
            if poging < 2:
                raad = input('Sorry, verkeerd antwoord. Probeer opnieuw ')
                poging = poging + 1

    if poging == 3:
        print('Het correcte antwoord is ' + antwoord)

score = 0
print('Raad het dier')
raad1 = input('Welke beer leeft op de Noordpool? ')
check_raad(raad1, 'ijsbeer')
raad2 = input('Wat is het snelste landdier? ')
check_raad(raad2, 'luipaard')
raad3 = input('Welk dier is het grootst? ')
check_raad(raad3, 'blauwe vinvis')

print('Jouw score is ' + str(score))
```

Wachtwoordkiezer (blz. 52)

```
import random
import string

adjectieven = ['slaperige', 'langzame', 'stinkende',
               'natte', 'vette', 'rode',
               'oranje', 'gele', 'groene',
               'blauwe', 'paarse', 'donzige',
```

```
'witte', 'trotse', 'dappere']
substantieven = ['appel', 'dinosaurus', 'bal',
                 'toaster', 'geit', 'draak',
                 'hamer', 'eend', 'panda']

print('Welkom bij Wachtwoordkiezer!')

while True:
    adjectief = random.choice(adjectieven)
    substantief = random.choice(substantieven)
    getal = random.randrange(0, 100)
    speciaal_teken = random.choice(string.punctuation)

    wachtwoord = adjectief + substantief + str(getal) + speciaal_teken
    print('Je nieuwe wachtwoord is: %s' % wachtwoord)

    antwoord = input('Wil je een ander wachtwoord? Typ j of n: ')
    if antwoord == 'n':
        break
```

Negen levens (blz. 60)

```
import random

levens = 9
woorden = ['pizza', 'elfje', 'toets', 'shirt', 'otter', 'plein']
geheime_woord = random.choice(woorden)
hint = list('?????')
hart_symbool = u'\u2764'
geraden_woord_juist = False

def update_hint(geraden_letter, geheime_woord, hint):
    index = 0
    while index < len(geheime_woord):
        if geraden_letter == geheime_woord[index]:
            hint[index] = geraden_letter
        index = index + 1

while levens > 0:
    print(hint)
    print('Levens over: ' + hart_symbool * levens)
    raad = input('Raad een letter of het hele woord: ')

    if raad == geheime_woord:
        geraden_woord_juist = True
        break

    if raad in geheime_woord:
        update_hint(raad, geheime_woord, hint)
    else:
```

```

print('Fout. Je verliest een leven')
levens = levens - 1

if geraden_woord_juist:
    print('Gewonnen! Het geheime woord was ' \
          + geheime_woord)
else:
    print('Verloren! Het geheime woord was ' \
          + geheime_woord)

```

Robotbouwer (blz. 72)

```

import turtle as t

def rechthoek(horizontaal, verticaal, kleur):
    t.pendown()
    t.pensize(1)
    t.color(kleur)
    t.begin_fill()
    for teller in range(1, 3):
        t.forward(horizontaal)
        t.right(90)
        t.forward(verticaal)
        t.right(90)
    t.end_fill()
    t.penup()

t.penup()
t.speed('slow')
t.bgcolor('Dodger blue')

# voeten
t.goto(-100, -150)
rechthoek(50, 20, 'blue')
t.goto(-30, -150)
rechthoek(50, 20, 'blue')

# benen
t.goto(-25, -50)
rechthoek(15, 100, 'grey')
t.goto(-55, -50)
rechthoek(-15, 100, 'grey')

# lichaam
t.goto(-90, 100)
rechthoek(100, 150, 'red')

# armen
t.goto(-150, 70)

```

```

rechthoek(60, 15, 'grey')
t.goto(-150, 110)
rechthoek(15, 40, 'grey')
t.goto(10, 70)
rechthoek(60, 15, 'grey')
t.goto(55, 110)
rechthoek(15, 40, 'grey')

```

```

# nek
t.goto(-50, 120)
rechthoek(15, 20, 'grey')

```

```

# hoofd
t.goto(-85, 170)
rechthoek(80, 50, 'red')

```

```

# ogen
t.goto(-60, 160)
rechthoek(30, 10, 'white')
t.goto(-55, 155)
rechthoek(5, 5, 'black')
t.goto(-40, 155)
rechthoek(5, 5, 'black')

```

```

# mond
t.goto(-65, 135)
rechthoek(40, 5, 'black')
t.hideturtle()

```

Kaleidospiraal (blz. 82)

```

import turtle
from itertools import cycle

kleuren = cycle(['red', 'orange', 'yellow', \
                'green', 'blue', 'purple'])

def teken_cirkel(grootte, hoek, draaiing):
    turtle.pencolor(next(kleuren))
    turtle.circle(grootte)
    turtle.right(hoek)
    turtle.forward(draaiing)
    teken_cirkel(grootte + 5, hoek + 1,
                 draaiing + 1)

turtle.bgcolor('black')
turtle.speed('fast')
turtle.pensize(4)
teken_cirkel(30, 0, 1)

```

Sterrennacht (blz. 90)

```
import turtle as t
from random import randint, random

def teken_ster(punten, grootte, kleur, x, y):
    t.penup()
    t.goto(x, y)
    t.pendown()
    hoek = 180 - (180 / punten)
    t.color(kleur)
    t.begin_fill()
    for i in range(punten):
        t.forward(grootte)
        t.right(hoek)
    t.end_fill()

# Hoofdcod
t.Screen().bgcolor('dark blue')

while True:
    willekeurig_punten = randint(2, 5) * 2 + 1
    willekeurig_grootte = randint(10, 50)
    willekeurig_kleur = (random(), random(), random())
    willekeurig_x = randint(-350, 300)
    willekeurig_y = randint(-250, 250)

    teken_ster(willekeurig_punten, willekeurig_grootte, willekeurig_kleur, willekeurig_x, willekeurig_y)
```

Rare regenboog (blz. 98)

```
import random
import turtle as t

def verkrijg_lijn_lengte():
    keuze = input('Voer de lengte voor de lijn in (lang, gemiddeld, kort): ')
    if keuze == 'lang':
        lijn_lengte = 250
    elif keuze == 'gemiddeld':
        lijn_lengte = 200
    else:
        lijn_lengte = 100
    return lijn_lengte

def verkrijg_lijn_dikte():
    keuze = input('Voer lijndikte in (superdik, dik, dun): ')
    if keuze == 'superdik':
        lijn_dikte = 40
    elif keuze == 'dik':
        lijn_dikte = 25
```

```

else:
    lijn_dikte = 10
return lijn_dikte

def in_venster():
    limiet_links = (-t.window_width() / 2) + 100
    limiet_rechts = (t.window_width() / 2) - 100
    limiet_boven = (t.window_height() / 2) - 100
    limiet_onder = (-t.window_height() / 2) + 100
    (x, y) = t.pos()
    binnen = limiet_links < x < limiet_rechts and limiet_onder < y < limiet_boven
    return binnen

def verplaats_turtle(lijn_lengte):
    pen_kleuren = ['red', 'orange', 'yellow', 'green', 'blue', 'purple']
    t.pencolor(random.choice(pen_kleuren))
    if in_venster():
        hoek = random.randint(0, 180)
        t.right(hoek)
        t.forward(lijn_lengte)
    else:
        t.backward(lijn_lengte)

lijn_lengte = verkrijg_lijn_lengte()
lijn_dikte = verkrijg_lijn_dikte()

t.shape('turtle')
t.fillcolor('green')
t.bgcolor('black')
t.speed('fastest')
t.pensize(lijn_dikte)

while True:
    verplaats_turtle(lijn_lengte)

```

Countdownkalender (blz. 110)

```

from tkinter import Tk, Canvas
from datetime import date, datetime

def ontvang_gebeurtenissen():
    lijst_gebeurtenissen = []
    with open('gebeurtenissen.txt') as bestand:
        for lijn in bestand:
            lijn = lijn.rstrip('\n')
            huidige_gebeurtenis = lijn.split(',')
            gebeurtenis_datum = datetime.strptime(huidige_gebeurtenis[1], '%d/%m/%y').date()
            huidige_gebeurtenis[1] = gebeurtenis_datum
            lijst_gebeurtenissen.append(huidige_gebeurtenis)
    return lijst_gebeurtenissen

```

```
def dagen_tussen_datums(datum1, datum2):
    tijd_tussen = str(datum1 - datum2)
    aantal_dagen = tijd_tussen.split(' ')
    return aantal_dagen[0]

root = Tk()
c = Canvas(root, width=800, height=800, bg='black')
c.pack()
c.create_text(100, 50, anchor='w', fill='orange', font='Arial 28 bold underline', \
              text='Mijn Countdownkalender')

gebeurtenissen = ontvang_gebeurtenissen()
vandaag = date.today()

verticale_ruimte = 100

for gebeurtenis in gebeurtenissen:
    gebeurtenis_naam = gebeurtenis[0]
    dagen_tot = dagen_tussen_datums(gebeurtenis[1], vandaag)
    weergave = 'Het is %s dagen tot %s' % (dagen_tot, gebeurtenis_naam)
    c.create_text(100, verticale_ruimte, anchor='w', fill='lightblue', \
                 font='Arial 28 bold', text=weergave)
    verticale_ruimte = verticale_ruimte + 30

root.mainloop()
```

Vraag het de expert (blz. 120)

```
from tkinter import Tk, simpledialog, messagebox

def lees_uit_bestand():
    with open('hoofdstad_data.txt') as bestand:
        for lijn in bestand:
            lijn = lijn.rstrip('\n')
            land, stad = lijn.split('/')
            de_wereld[land] = stad

def schrijf_naar_bestand(naam_land, naam_stad):
    with open('hoofdstad_data.txt', 'a') as bestand:
        bestand.write('\n' + naam_land + '/' + naam_stad)

print('Vraag het de expert - Hoofdsteden van de wereld')
root = Tk()
root.withdraw()
de_wereld = {}

lees_uit_bestand()

while True:
    vraag_land = simpledialog.askstring('Land', 'Typ de naam van een land:')

    if vraag_land in de_wereld:
```

```

resultaat = de_wereld[vraag_land]
messagebox.showinfo('Antwoord',
                    'De hoofdstad van ' + vraag_land + ' is ' + resultaat + '!')
else:
    nieuwe_stad = simpledialog.askstring('Vertel het me',
                                         'Ik weet het niet! ' +
                                         'Wat is de hoofdstad van ' + vraag_land + '?')
    de_wereld[vraag_land] = nieuwe_stad
    schrijf_naar_bestand(vraag_land, nieuwe_stad)

root.mainloop()

```

Geheime berichten (blz. 130)

```

from tkinter import messagebox, simpledialog, Tk

def is_even(getal):
    return getal % 2 == 0

def ontvang_even_letters(bericht):
    even_letters = []
    for teller in range(0, len(bericht)):
        if is_even(teller):
            even_letters.append(bericht[teller])
    return even_letters

def ontvang_oneven_letters(bericht):
    oneven_letters = []
    for teller in range(0, len(bericht)):
        if not is_even(teller):
            oneven_letters.append(bericht[teller])
    return oneven_letters

def wissel_letters(bericht):
    letter_lijt = []
    if not is_even(len(bericht)):
        bericht = bericht + 'x'
    even_letters = ontvang_even_letters(bericht)
    oneven_letters = ontvang_oneven_letters(bericht)
    for teller in range(0, int(len(bericht)/2)):
        letter_lijt.append(oneven_letters[teller])
        letter_lijt.append(even_letters[teller])
    nieuw_bericht = ''.join(letter_lijt)
    return nieuw_bericht

def ontvang_taak():
    taak = simpledialog.askstring('Taak', 'Wil je coderen of decoderen?')
    return taak

```

```
def ontvang_bericht():
    bericht = simplifiedialog.askstring('Bericht', 'Voer het geheime bericht in: ')
    return bericht

root = Tk()

while True:
    taak = ontvang_taak()
    if taak == 'coderen':
        bericht = ontvang_bericht()
        gecodeerd = wissel_letters(bericht)
        messagebox.showinfo('Ciphertekst van het geheime bericht is:', gecodeerd)
    elif taak == 'decoderen':
        bericht = ontvang_bericht()
        gedecodeerd = wissel_letters(bericht)
        messagebox.showinfo('Platte tekst van het geheime bericht is:', gedecodeerd)
    else:
        break
root.mainloop()
```

Schermdier (blz. 142)

```
from tkinter import HIDDEN, NORMAL, Tk, Canvas

def toggle_ogen():
    huidig_kleur = c.itemcget(oog_links, 'fill')
    nieuw_kleur = c.lijf_kleur if huidig_kleur == 'white' else 'white'
    huidige_status = c.itemcget(pupil_links, 'state')
    nieuwe_status = NORMAL if huidige_status == HIDDEN else HIDDEN
    c.itemconfigure(pupil_links, state=nieuwe_status)
    c.itemconfigure(pupil_rechts, state=nieuwe_status)
    c.itemconfigure(oog_links, fill=nieuw_kleur)
    c.itemconfigure(oog_rechts, fill=nieuw_kleur)

def knippen():
    toggle_ogen()
    root.after(250, toggle_ogen)
    root.after(3000, knippen)

def toggle_pupillen():
    if not c.scheel:
        c.move(pupil_links, 10, -5)
        c.move(pupil_rechts, -10, -5)
        c.scheel = True
    else:
        c.move(pupil_links, -10, 5)
        c.move(pupil_rechts, 10, 5)
        c.scheel = False
```

```
def toggle_tong():
    if not c.tong_uit:
        c.itemconfigure(tong_tip, state=NORMAL)
        c.itemconfigure(tong_hoofd, state=NORMAL)
        c.tong_uit = True
    else:
        c.itemconfigure(tong_tip, state=HIDDEN)
        c.itemconfigure(tong_hoofd, state=HIDDEN)
        c.tong_uit = False

def brutaal(event):
    toggle_tong()
    toggle_pupillen()
    verberg_blij(event)
    root.after(1000, toggle_tong)
    root.after(1000, toggle_pupillen)
    return

def toon_blij(event):
    if (20 <= event.x and event.x <= 350) and (20 <= event.y and event.y <= 350):
        c.itemconfigure(wang_links, state=NORMAL)
        c.itemconfigure(wang_rechts, state=NORMAL)
        c.itemconfigure(mond_blij, state=NORMAL)
        c.itemconfigure(mond_normaal, state=HIDDEN)
        c.itemconfigure(mond_sip, state=HIDDEN)
        c.blij = 10
    return

def verberg_blij(event):
    c.itemconfigure(wang_links, state=HIDDEN)
    c.itemconfigure(wang_rechts, state=HIDDEN)
    c.itemconfigure(mond_blij, state=HIDDEN)
    c.itemconfigure(mond_normaal, state=NORMAL)
    c.itemconfigure(mond_sip, state=HIDDEN)
    return

def verdrietig():
    if c.blij == 0:
        c.itemconfigure(mond_blij, state=HIDDEN)
        c.itemconfigure(mond_normaal, state=HIDDEN)
        c.itemconfigure(mond_sip, state=NORMAL)
    else:
        c.blij -= 1
        root.after(5000, verdrietig)

root = Tk()
c = Canvas(root, width=400, height=400)
c.configure(bg='dark blue', highlightthickness=0)
c.lijf_kleur = 'SkyBlue1'
```

```

lijf = c.create_oval(35, 20, 365, 350, outline=c.lijf_kleur, fill=c.lijf_kleur)
oor_links = c.create_polygon(75, 80, 75, 10, 165, 70, outline=c.lijf_kleur, fill=c.lijf_kleur)
oor_rechts = c.create_polygon(255, 45, 325, 10, 320, 70, outline=c.lijf_kleur, fill=c.lijf_kleur)
voet_links = c.create_oval(65, 320, 145, 360, outline=c.lijf_kleur, fill=c.lijf_kleur)
voet_rechts = c.create_oval(250, 320, 330, 360, outline=c.lijf_kleur, fill=c.lijf_kleur)

oog_links = c.create_oval(130, 110, 160, 170, outline='black', fill='white')
pupil_links = c.create_oval(140, 145, 150, 155, outline='black', fill='black')
oog_rechts = c.create_oval(230, 110, 260, 170, outline='black', fill='white')
pupil_rechts = c.create_oval(240, 145, 250, 155, outline='black', fill='black')

mond_normaal = c.create_line(170, 250, 200, 272, 230, 250, smooth=1, width=2, state=NORMAL)
mond_blij = c.create_line(170, 250, 200, 282, 230, 250, smooth=1, width=2, state=HIDDEN)
mond_sip = c.create_line(170, 250, 200, 232, 230, 250, smooth=1, width=2, state=HIDDEN)
tong_hoofd = c.create_rectangle(170, 250, 230, 290, outline='red', fill='red', state=HIDDEN)
tong_tip = c.create_oval(170, 285, 230, 300, outline='red', fill='red', state=HIDDEN)

wang_links = c.create_oval(70, 180, 120, 230, outline='pink', fill='pink', state=HIDDEN)
wang_rechts = c.create_oval(280, 180, 330, 230, outline='pink', fill='pink', state=HIDDEN)

c.pack()

c.bind('<Motion>', toon_blij)
c.bind('<Leave>', verberg_blij)
c.bind('<Double-1>', brutaal)

c.blij = 10
c.scheel = False
c.tong_uit = False

root.after(1000, knipperen)
root.after(5000, verdrietig)
root.mainloop()

```

Rups (blz. 158)

```

import random
import turtle as t

t.bgcolor('yellow')

rups = t.Turtle()
rups.shape('square')
rups.color('red')
rups.speed(0)
rups.penup()
rups.hideturtle()

blad = t.Turtle()

```

```
blad_vorm = ((0, 0), (14, 2), (18, 6), (20, 20), (6, 18), (2, 14))
t.register_shape('blad', blad_vorm)
blad.shape('blad')
blad.color('green')
blad.penup()
blad.hideturtle()
blad.speed(0)

spel_gestart = False
tekst_turtle = t.Turtle()
tekst_turtle.write('Druk SPATIE om te beginnen', align='center', font=('Arial', 16, 'bold'))
tekst_turtle.hideturtle()

score_turtle = t.Turtle()
score_turtle.hideturtle()
score_turtle.speed(0)

def buiten_venster():
    muur_links = -t.window_width() / 2
    muur_rechts = t.window_width() / 2
    muur_boven = t.window_height() / 2
    muur_onder = -t.window_height() / 2
    (x, y) = rups.pos()
    buiten = \
        x < muur_links or \
        x > muur_rechts or \
        y < muur_onder or \
        y > muur_boven
    return buiten

def game_over():
    rups.color('yellow')
    blad.color('yellow')
    t.penup()
    t.hideturtle()
    t.write('GAME OVER!', align='center', font=('Arial', 30, 'normal'))

def toon_score(huidige_score):
    score_turtle.clear()
    score_turtle.penup()
    x = (t.window_width() / 2) - 50
    y = (t.window_height() / 2) - 50
    score_turtle.setpos(x, y)
    score_turtle.write(str(huidige_score), align='right', font=('Arial', 40, 'bold'))

def plaats_blad():
    blad.ht()
    blad.setx(random.randint(-200, 200))
    blad.sety(random.randint(-200, 200))
    blad.st()
```

```
def start_game():
    global spel_gestart
    if spel_gestart:
        return
    spel_gestart = True

    score = 0
    tekst_turtle.clear()

    rups_snelheid = 2
    rups_lengte = 3
    rups.shapesize(1, rups_lengte, 1)
    rups.showturtle()
    toon_score(score)
    plaats_blad()

    while True:
        rups.forward(rups_snelheid)
        if rups.distance(blad) < 20:
            plaats_blad()
            rups_lengte = rups_lengte + 1
            rups.shapesize(1, rups_lengte, 1)
            rups_snelheid = rups_snelheid + 1
            score = score + 10
            toon_score(score)
        if buiten_venster():
            game_over()
            break

def naar_boven():
    if rups.heading() == 0 or rups.heading() == 180:
        rups.setheading(90)

def naar_onder():
    if rups.heading() == 0 or rups.heading() == 180:
        rups.setheading(270)

def naar_links():
    if rups.heading() == 90 or rups.heading() == 270:
        rups.setheading(180)

def naar_rechts():
    if rups.heading() == 90 or rups.heading() == 270:
        rups.setheading(0)

t.onkey(start_game, 'space')
t.onkey(naar_boven, 'Up')
t.onkey(naar_rechts, 'Right')
t.onkey(naar_onder, 'Down')
```

```
t.onkey(naar_links, 'Left')
t.listen()
t.mainloop()
```

Snap (blz. 168)

```
import random
import time
from tkinter import Tk, Canvas, HIDDEN, NORMAL

def volgende_vorm():
    global vorm
    global vorige_kleur
    global huidige_kleur

    vorige_kleur = huidige_kleur

    c.delete(vorm)
    if len(vormen) > 0:
        vorm = vormen.pop()
        c.itemconfigure(vorm, state=NORMAL)
        huidige_kleur = c.itemcget(vorm, 'fill')
        root.after(1000, volgende_vorm)
    else:
        c.unbind('q')
        c.unbind('p')
        if speler1_score > speler2_score:
            c.create_text(200, 200, text='Winnaar: Speler 1')
        elif speler2_score > speler1_score:
            c.create_text(200, 200, text='Winnaar: Speler 2')
        else:
            c.create_text(200, 200, text='Gelijkspel')
        c.pack()

def snap(event):
    global vorm
    global speler1_score
    global speler2_score
    geldig = False

    c.delete(vorm)
    if vorige_kleur == huidige_kleur:
        geldig = True

    if geldig:
        if event.char == 'q':
            speler1_score = speler1_score + 1
        else:
```

```
        speler2_score = speler2_score + 1
    vorm = c.create_text(200, 200, text='SNAP! Je hebt 1 punt gescoord!')
else:
    if event.char == 'q':
        speler1_score = speler1_score - 1
    else:
        speler2_score = speler2_score - 1
    vorm = c.create_text(200, 200, text='FOUT! Je hebt 1 punt verloren!')
c.pack()
root.update_idletasks()
time.sleep(1)

root = Tk()
root.title('Snap')
c = Canvas(root, width=400, height=400)

vormen = []

cirkel = c.create_oval(35, 20, 365, 350, outline='black', fill='black', state=HIDDEN)
vormen.append(cirkel)
cirkel = c.create_oval(35, 20, 365, 350, outline='red', fill='red', state=HIDDEN)
vormen.append(cirkel)
cirkel = c.create_oval(35, 20, 365, 350, outline='green', fill='green', state=HIDDEN)
vormen.append(cirkel)
cirkel = c.create_oval(35, 20, 365, 350, outline='blue', fill='blue', state=HIDDEN)
vormen.append(cirkel)

rechthoek = c.create_rectangle(35, 100, 365, 270, outline='black', fill='black', state=HIDDEN)
vormen.append(rechthoek)
rechthoek = c.create_rectangle(35, 100, 365, 270, outline='red', fill='red', state=HIDDEN)
vormen.append(rechthoek)
rechthoek = c.create_rectangle(35, 100, 365, 270, outline='green', fill='green', state=HIDDEN)
vormen.append(rechthoek)
rechthoek = c.create_rectangle(35, 100, 365, 270, outline='blue', fill='blue', state=HIDDEN)
vormen.append(rechthoek)

vierkant = c.create_rectangle(35, 20, 365, 350, outline='black', fill='black', state=HIDDEN)
vormen.append(vierkant)
vierkant = c.create_rectangle(35, 20, 365, 350, outline='red', fill='red', state=HIDDEN)
vormen.append(vierkant)
vierkant = c.create_rectangle(35, 20, 365, 350, outline='green', fill='green', state=HIDDEN)
vormen.append(vierkant)
vierkant = c.create_rectangle(35, 20, 365, 350, outline='blue', fill='blue', state=HIDDEN)
vormen.append(vierkant)
c.pack()

random.shuffle(vormen)

vorm = None
```

```

vorige_kleur = ''
huidige_kleur = ''
speler1_score = 0
speler2_score = 0

root.after(3000, volgende_vorm)
c.bind('q', snap)
c.bind('p', snap)
c.focus_set()

root.mainloop()

```

Memory (blz. 180)

```

import random
import time
from tkinter import Tk, Button, DISABLED

def toon_symbool(x, y):
    global eerste
    global vorigeX, vorigeY
    knoppen[x, y]['text'] = knop_symbolen[x, y]
    knoppen[x, y].update_idletasks()

    if eerste:
        vorigeX = x
        vorigeY = y
        eerste = False
    elif vorigeX != x or vorigeY != y:
        if knoppen[vorigeX, vorigeY]['text'] != knoppen[x, y]['text']:
            time.sleep(0.5)
            knoppen[vorigeX, vorigeY]['text'] = ''
            knoppen[x, y]['text'] = ''
        else:
            knoppen[vorigeX, vorigeY]['command'] = DISABLED
            knoppen[x, y]['command'] = DISABLED
            eerste = True

root = Tk()
root.title('Memory')
root.resizable(width=False, height=False)

knoppen = {}
eerste = True
vorigeX = 0
vorigeY = 0

knop_symbolen = {}

symbolen = [u'\u2702', u'\u2702', u'\u2705', u'\u2705', u'\u2708', u'\u2708',
            u'\u2709', u'\u2709', u'\u270A', u'\u270A', u'\u270B', u'\u270B',
            u'\u270C', u'\u270C', u'\u270F', u'\u270F', u'\u2712', u'\u2712',

```

```
u'\u2714', u'\u2714', u'\u2716', u'\u2716', u'\u2728', u'\u2728']

random.shuffle(symbolen)

for x in range(6):
    for y in range(4):
        knop = Button(command=lambda x=x, y=y: toon_symbool(x, y), width=3, height=3)
        knop.grid(column=x, row=y)
        knoppen[x, y] = knop
        knop_symbolen[x, y] = symbolen.pop()

root.mainloop()
```

Eiervanger (blz. 190)

```
from itertools import cycle
from random import randrange
from tkinter import Canvas, Tk, messagebox, font

canvas_breedte = 800
canvas_hoogte = 400

root = Tk()
c = Canvas(root, width=canvas_breedte, height=canvas_hoogte, \
background='deep sky blue')
c.create_rectangle(-5, canvas_hoogte - 100, canvas_breedte + 5, \
canvas_hoogte + 5, fill='sea green', width=0)
c.create_oval(-80, -80, 120, 120, fill='orange', width=0)
c.pack()

kleur_cyclus = cycle(['light blue', 'light green', 'light pink', 'light yellow', 'light cyan'])
ei_breedte = 45
ei_hoogte = 55
ei_score = 10
ei_snelheid = 500
ei_interval = 4000
moeilijkheidsgraad = 0.95

vanger_kleur = 'blue'
vanger_breedte = 100
vanger_hoogte = 100
vanger_start_x = canvas_breedte / 2 - vanger_breedte / 2
vanger_start_y = canvas_hoogte - vanger_hoogte - 20
vanger_start_x2 = vanger_start_x + vanger_breedte
vanger_start_y2 = vanger_start_y + vanger_hoogte

vanger = c.create_arc(vanger_start_x, vanger_start_y, \
vanger_start_x2, vanger_start_y2, start=200, extent=140, \
style='arc', outline=vanger_kleur, width=3)
```

```
spel_lettertype = font.nametofont('TkFixedFont')
spel_lettertype.config(size=18)

score = 0
score_tekst = c.create_text(10, 10, anchor='nw', font=spel_lettertype, fill='darkblue', \
                             text='Score: ' + str(score))

resterende_levens = 3
levens_tekst = c.create_text(canvas_breedte - 10, 10, anchor='ne', font=spel_lettertype, \
                              fill='darkblue', text='Levens: ' + str(resterende_levens))

eieren = []

def maak_ei():
    x = randrange(10, 740)
    y = 40
    nieuw_ei = c.create_oval(x, y, x + ei_breedte, y + ei_hoogte, fill=next(kleur_cyclus), width=0)
    eieren.append(nieuw_ei)
    root.after(ei_interval, maak_ei)

def beweeg_eieren():
    for ei in eieren:
        (ei_x, ei_y, ei_x2, ei_y2) = c.coords(ei)
        c.move(ei, 0, 10)
        if ei_y2 > canvas_hoogte:
            ei_gevallen(ei)

    root.after(ei_snelheid, beweeg_eieren)

def ei_gevallen(ei):
    eieren.remove(ei)
    c.delete(ei)
    verlies_een_leven()
    if resterende_levens == 0:
        messagebox.showinfo('Game Over!', 'Eindscore: ' \
                              + str(score))

    root.destroy()

def verlies_een_leven():
    global resterende_levens
    resterende_levens -= 1
    c.itemconfigure(levens_tekst, text='Levens: ' \
                    + str(resterende_levens))

def controleer_vangst():
    (vanger_x, vanger_y, vanger_x2, vanger_y2) = c.coords(vanger)
    for ei in eieren:
        (ei_x, ei_y, ei_x2, ei_y2) = c.coords(ei)
        if vanger_x < ei_x and ei_x2 < vanger_x2 and vanger_y2 - ei_y2 < 40:
            eieren.remove(ei)
            c.delete(ei)
            verhoog_score(ei_score)
    root.after(100, controleer_vangst)
```

```
def verhoog_score(punten):
    global score, ei_snelheid, ei_interval
    score += punten
    ei_snelheid = int(ei_snelheid * moeilijkheidsgraad)
    ei_interval = int(ei_interval * moeilijkheidsgraad)
    c.itemconfigure(score_tekst, text='Score: ' + str(score))

def beweeg_links(event):
    (x1, y1, x2, y2) = c.coords(vanger)
    if x1 > 0:
        c.move(vanger, -20, 0)

def beweeg_rechts(event):
    (x1, y1, x2, y2) = c.coords(vanger)
    if x2 < canvas_breedte:
        c.move(vanger, 20, 0)

c.bind('<Left>', beweeg_links)
c.bind('<Right>', beweeg_rechts)
c.focus_set()

root.after(1000, maak_ei)
root.after(1000, beweeg_eieren)
root.after(1000, controleer_vangst)
root.mainloop()
```