## Schleifen mit Turtle

### Aufgabe 1 - Wiederholte Bewegung für immer ohne Schleife

Wir wollen jedoch ein Konzept grob als Idee beschreiben. Die Rekursion. Dies ist eine Schachtelung einer **eigens geschriebenen** Funktion oder Prozedur. Schachtelung bedeutet hier, wir rufen in der Prozedur nochmals die gleiche Prozedur auf. Wenn wir diese Prozedur einmal aufrufen, ruft diese sich selbst immer und immer wieder auf und endet deshalb nie. Wir verzichten hier weiteres über die Rekursion zu sprechen, denn diese ist ein sehr forteschrittenes, gefährliches und deshalb selten verwendetes Konzept, welches hier nur einmal erwähnt wird. Es gibt uns aber die Change einmal selbst eine Funktion oder Prozedur zu definieren.

Verwende dazu folgende eigens geschriebeneProzeduren: \* def ich\_rufe\_mich\_selbst\_auf(): Wir definieren eine Prozedur welche sich immer und immer wieder selbst aufruft.

Diese Aufgabe hat die Gleiche Angabe wie [exercise 1, Aufgabe 1](../../exercise1-while_und_for_schleifen/angabe.md#aufgabe-1---wiederholte-bewegung-für-immer). Wir sehen zudem keinen Unterschied wenn wir das Programm laufen lassen zur Lösung mit der While-Schleife.

**Lösung:**

from turtle import \*  
from random import randint  
  
# --- Vorbereitung ---  
# Die Variablen für die Geschwindigkeit.  
geschwindigkeit\_am\_land = 1  
geschwindigkeit\_im\_wasser = 50 \* geschwindigkeit\_am\_land  
  
# Die Variablen für die Form.  
form\_am\_land = "turtle"  
form\_im\_wasser = "circle"  
  
shape(form\_am\_land)   
speed(geschwindigkeit\_am\_land)  
  
# --- Logik ---  
# Die Turtle nimmt den Faden aus der Tasche und legt ihn am Boden wenn sie losgeht.  
pendown()  
  
# Die Turtle drückt den Faden in den boden, fixiert diesen und macht einen Abdruck von sich selbst.  
stamp()  
  
breite = window\_width()  
hoehe = window\_height()  
  
halbe\_breite\_ohne\_komma = breite // 2   
halbe\_hoehe\_ohne\_komma = hoehe // 2  
  
# Wir erstellen eine neue Prozedur. Das wird Definition einer Funktion genannt und wird mit dem Keyword def gemacht.  
# Achtung! Wir enden mit einem Doppelpunkt, denn danach kommt eine Einrückung.  
def ich\_rufe\_mich\_selbst\_auf():  
 # Wir schreiben alles was zuerst in der While-Schleife war, in die Prozedur rein.  
 ziel\_in\_x = randint(-halbe\_breite\_ohne\_komma, halbe\_breite\_ohne\_komma)  
 ziel\_in\_y = randint(-halbe\_hoehe\_ohne\_komma, halbe\_hoehe\_ohne\_komma)  
  
 # Wir bewegen die Turtle und passen an wie diese dargestellt wird (Kreis oder Turtle).   
 if ziel\_in\_x > 0:  
 # Wir verwenden die Form für die rechte Seite. Diese ist 'turtle'.  
 shape(form\_am\_land)   
 speed(geschwindigkeit\_am\_land)  
 else:  
 # Wir verwenden die Form für die rechte Seite. Diese ist 'circle'.  
 shape(form\_im\_wasser)   
 speed(geschwindigkeit\_im\_wasser)  
  
 # Die Turtle nimmt den Faden aus der Tasche und legt ihn am Boden wenn sie losgeht.  
 pendown()  
  
 # Die Turtle bewegt sich an einen zufällig gewählten Ort.  
 goto(ziel\_in\_x, ziel\_in\_y)  
  
 # Die Turtle markier diesen Ort mit einem roten Punkt.  
 dot(25, "red")  
  
 # Die Turtle schneidet den Faden ab und gibt die Rolle mit dem Faden wieder in die Tasche. Sie legt ihn nicht mehr auf den Boden wenn sie losgeht.  
 penup()  
  
 # Die Turtle bewegt sich zur Ausgangsposition zurück.  
 goto(0, 0)  
  
 # Wichtig! Wir müssen uns am Ende der Prozedur selbst aufrufen!   
 # Damit ist ein Aufruf von ich\_rufe\_mich\_selbst\_auf gemeint.  
 # Wir rufen eine Prozedur ohne Parameter auf, indem wir () hinter den Namen der Prozedur schreiben.  
 ich\_rufe\_mich\_selbst\_auf()  
  
  
# Der start der Rekursion welche niemals abbrechen soll.  
# Wir haben nun ich\_rufe\_mich\_selbst\_auf erstellt (definiert), jedoch noch nicht aufgerufen.   
# Um das zu tun schreiben wir ich\_rufe\_mich\_selbst\_auf() nochmals hin.  
ich\_rufe\_mich\_selbst\_auf()  
  
# --- Abschluss ---  
# Hier steht nichts mehr... denn wir sind für immer oben gefangen.