# Programmieren mit Python

**Ein Minikurs für die Sek-Stufe I**

SchulArena.com

## Inhaltsverzeichnis



## *Beispiele aus dem Dossier:*

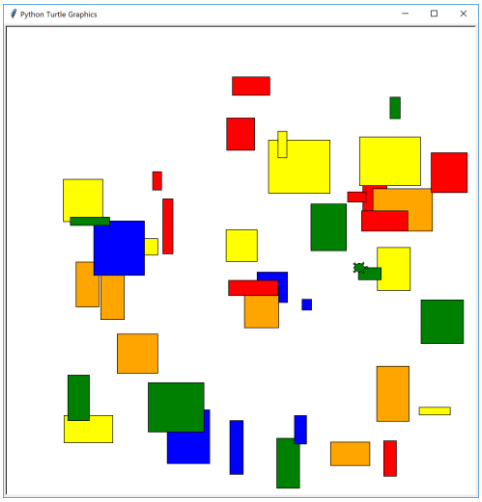
### 6.8 Aufgabe 8: Random Art

Hier soll Kunst nach Zufallsprinzip entstehen.

Es sollen Rechtecke in Zufallsgrössen mit Zufallsfarbfüllungen und an Zufallskoordinaten entstehen.

Eckdaten:

Rechtecke: Seiten von 10 bis 100 Pixel
Koordinaten x, y: zwischen -300 und 300
Farben: zufällige Auswahl aus 6 vordefinierten (siehe Aufgabe 3.2 als Hilfe)




Definiere (*def*:):
-Zufällige Koordinaten
-Zufällige Farbe
-Wie man ein Rechteck malt

Schleife:
Rufe die drei Definitionen in einer Schleife z.B. 40x auf

Einige Code-Hilfen:

```
from turtle import *
from random import randint
import random
shape("turtle")

def drawrectangle():
 length = randint(10, 100)
 height = randint(10, 100)
```

Erweiterung: Alle Farben, die es gibt, zufällig:

```
# define a random colour
def randomcolor():
 colormode(255)
 red = random.randint(0,255)
 blue = random.randint(0,255)
 green = random.randint(0,255)
 fillcolor(red,blue,green)
```

Beschrifte deinen Code gut und deutlich mit Kommentaren auf Englisch hinter Hashtags #.

Abspeichern: *random_art.py*

Erweiterung: Gleiche Aufgabe mit Kreisen, Dreiecken, Fünfecken usw.

siehe Zusatzaufgaben 5 + 7

### 8.2 Aufgabe 2: Zahlen-Ratespiel

Nun wollen wir ein kleines Zahlen-Ratespiel programmieren. Der Computer sagt, du sollst eine Zahl zwischen 0 und 100 erraten, welche das Programm zufällig auswählt. Nach jedem Rateversuch bekommst du eine Rückmeldung, ob die gesuchte Zahl grösser oder kleiner ist. Dies dauert so lange, bis die richtige Zahl erraten wird, was dann auch vom Computer bestätigt wird.

Dazu werden wir zuerst einige neue Befehle kennen lernen:

Da wir bei dieser Aufgabe nicht voraussagen können, wie oft das Programm durchlaufen werden muss, brauchen wir eine Schleife, welche die Bedingungen so lange prüft, bis die Zahl gefunden wurde:
*while True:*

hier Beginn der Schleife als Hilfe:

```
while True:
    zahl = float(input())
    if zahl > ergebnis:
        print("Die gesuchte Zahl ist kleiner.")
```

Dann gibt es drei Bedingungen. Die eine prüft, ob die Zahl, welche der User eingibt, zu klein ist. Eine weitere schaut, ob die eingegebene Zahl zu gross ist. Die dritte Bedingung prüft, ob die Zahl gefunden wurde, was zum Abbruch der Schleife führen muss:

*if zahl < ergebnis:*
*elif zahl > ergebnis:* *(steht für: else if)*
*elif zahl == ergebnis:*

Die Schleife unterbrechen wir mit: *break* nach der Bedingung „Ergebnis erreicht“.

Dann spielt hier auch noch der Zufall mit. Somit importieren wir zuerst:
*import random*

Die vom Computer ausgewählte Zahl definieren wir so:
*ergebnis = random.randint(0, 100)*

Denk daran, dass wir nach den Befehlen, welche mit einem Doppelpunkt enden, die folgenden Programmblöcke einrücken müssen.

Hier noch eine kleine Hilfe für die Eingabe: *zahl = int(input())*

Kommentare:

Erste Bedingung: *if*
Weitere Bedingungen: *elif*
*int* bei *zahl = int(input())* sagt dem Programm, dass es sich um eine ganze Zahl handelt. Statt *int* könntest du auch das dir bekannte *float* nehmen. Dann könntest du sogar mit Kommastellen arbeiten.

So, nun zeig mal, was du kannst. Nicht vergessen: Geht nicht, gibt's nicht! 😊

Speichern: *zahlen_raten.py*

siehe Zusatzaufgabe 3

Lösungen:

### 6.8 Random Art:

File Edit Format Run Options Window Help

```
# Code to draw a random-sized rectangle at random places
# with random colours

from turtle import *
from random import randint
import random
shape("turtle")

# define coordinates
def randomplace():
 up()
 x = randint(-300, 300)
 y = randint(-300, 300)
 goto(x,y)
 down()

# define a colour
def randomcolor():
 colors = ("red", "green", "blue", "yellow", "orange")
 fillcolor(random.choice(colors))

# draw a rectangle
def drawrectangle():
 length = randint(10, 100)
 height = randint(10, 100)
 begin_fill()
 forward(length)
 right(90)
 forward(height)
 right(90)
 forward(length)
 right(90)
 forward(height)
 right(90)
 end_fill()

# make 40 rectangles
for i in range(40):
 randomplace()
 randomcolor()
 drawrectangle()
```

An dem Beispiel kann man auch gut erkennen, wie wichtig das Einrücken bei den Codes ist, damit dem Programm klar ist, welche Elemente zusammengehören.

### 8.2 Zahlen-Ratespiel:

```
File  Edit  Format  Run  Options  Window  Help
import random

ergebnis = random.randint(0, 100)
print("Willkommen zum Zahlen-Ratespiel!")
print("Bitte errate die gesuchte Zahl; sie befindet sich zwischen 1 und 100: ")
while True:
    zahl = float(input())
    if zahl > ergebnis:
        print("Die gesuchte Zahl ist kleiner.")
    elif zahl < ergebnis:
        print("Die gesuchte Zahl ist größer.")
    elif zahl == ergebnis:
        print("Glückwunsch die von dir eingebene Zahl" , ergebnis ,  "stimmt mit der gesuchten Zahl überein.")
        break
```

Die Lösungen gibt es alle auch als Python-Dateien und als Videos, wo man das ausgeführte Programm sehen und beobachten kann.

***Beispiel eine Zusatzaufgabe:***

Die Zusatzaufgaben haben bewusst weniger Vorgaben und Hilfestellungen!

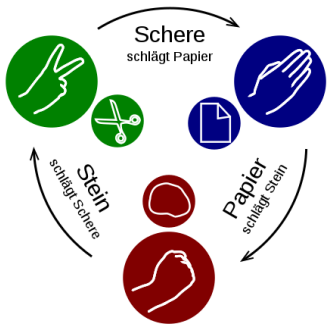
## Schere, Stein, Papier:


Du kennst das Spiel nicht? Hier nachlesen:
https://de.wikipedia.org/wiki/Schere,_Stein,_Papier

Du spielst gegen den Computer, der zufällig eine der drei Möglichkeiten wählt: Schere, Stein oder Papier

Du wählst zuerst, dann wird dir angezeigt, was der Computer gewählt hat und der Sieger wird ermittelt.

Das Spiel soll so lange gespielt werden, bis der User abbrechen will.




Tipps:

import: random & time
Variablen als Tuple: figuren("Schere", "Stein", "Papier")
spielen = True
weitere Variablen: spielerauswahl, spielerfigur, computerfigur
computerfigur = figuren[(random.randint(0,2))]

Es soll auch ein schönes „Titelbild“ geben mit verzögerter Anzeige (deshalb *import time*):

```
*******************************
* SCHERE | STEIN | PAPIER *
*******************************
```

Hilfe (Ausschnitt aus Code):

```
#Sieger oder Verlierer ermitteln
if spielerfigur == computerfigur:
    print("Es ist unentschieden. Der Computer wählte",computerfigur)
else:
    if spielerfigur == "Schere":
        if computerfigur == "Stein":
            print("Du hast verloren! Der Computer wählte", computerfigur)
        else:
            print("Gewonnen! Der Computer wählte", computerfigur)
```

Hilfe für nochmals spielen oder nicht:

```
#Nochmals spielen
time.sleep(1)
entscheidung = ""
while entscheidung not in ["j", "n"]:
    entscheidung = input("Willst du nochmals spielen? [j] Ja [n] Nein: ")
if entscheidung == "n":
    break
```