9. IF-ELSE

DU LERNST HIER...

wie man vorgehen muss, damit Programmblöcke nur unter gewissen Bedingungen ausgeführt werden. Du lernst auch, wie man Bedingungen *negiert* und wie man sie mit *Und*-, sowie *Oder*-Operatoren kombiniert.

MUSTERBEISPIEL

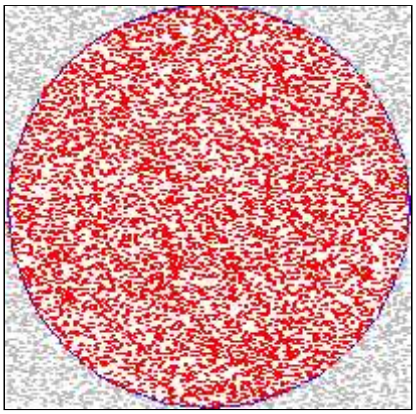
In deinem Programm erzeugst du mit dem Befehl *randin(-250, 250)* zwei Zufallszahlen x und y und lässt die Turtle an die Position (x, y) springen. Dort zeichnest einen rotern Punkt, wenn sich die Turtle innerhalb des Kreises mit dem Radius 250 befindet, sonst einen grauen Punkt. Diesen Vorgang wiederholst du 10 000 mal. Manchmal wird dies auch als Zufallsregen bezeichnet.

Befindet sich die Turtle innerhalb des Kreises, so gilt nach dem Satz von Pythagoras, dass ihr Abstandsquadrat *rsquare* zum Ursprung kleiner als 250 * 250 = 62500 ist.

Umgangssprachlich würdest du sagen:

"Falls das Radiusquadrat kleiner als 62500 ist, zeichne einen roten Punkt, sonst zeichne einen grauen Punkt."

Dies drückst du mit dem Keyword **if bedingung:** aus und rückst den nachfolgenden Programmblock ein. Dabei darfst du den Doppelpunkt nicht vergessen! Falls die Bedingung nicht erfüllt ist, wird der Programmblock ausgeführt, der unter der Zeile mit **else:** (wieder mit Doppelpunkt) steht.



Programm: **[► Online-Editor]** **[► WebTigerJython]**

```
from gturtle import *
from random import randint

makeTurtle()
hideTurtle()

repeat 10000:
    x = randint(-250, 250)
    y = randint(-250, 250)
    setPos(x, y)
    rsquare = x * x + y * y
    if rsquare < 62500:
       setPenColor("red")
       dot(4)
    else:
       setPenColor("gray")
       dot(4)
```

► In Zwischenablage kopieren

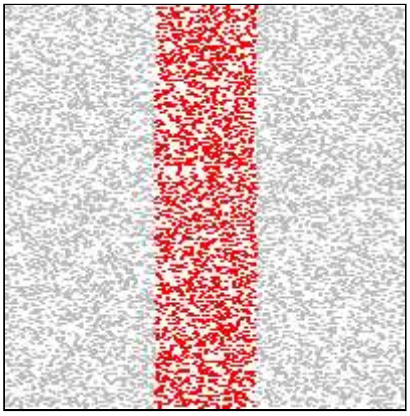
MERKE DIR...

Statt zu sagen, dass eine Bedingung erfüllt oder nicht erfüllt ist, sagt man auch, die Bedingung

sei **wahr** oder **falsch.** Sollen die grauen Punkte nicht gezeichnet werden, kannst du den else-Teil auch weglassen. Versuche es!

Falls du das Programm vorzeitig abbrechen willst, so klickst du auf den roten quadratischen Button.

BEDINGUNGEN MIT *UND* VERKNÜPFEN



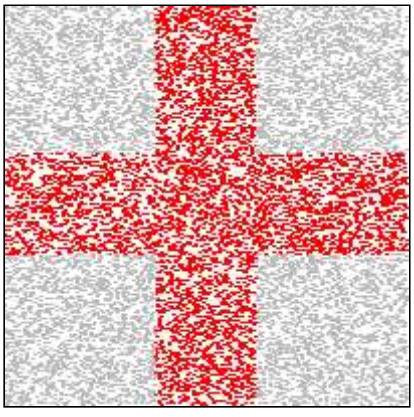
Wie auch in der Umgangssprache üblich, kannst du zwei Bedingungen mit **and** verknüpfen. Die so kombinierte Bedingung ist nur dann wahr, wenn beide Teilbedingungen wahr sind. Hier zeichnest du dann rote Punkte, wenn die x-Koordinate der Turtle *grösser als -100* ***und*** *kleiner als 100* ist. Es entsteht offenbar ein vertikaler roter Streifen.

Programm: **[► Online-Editor]** **[► WebTigerJython]**

```
from gturtle import *
from random import randint

makeTurtle()
hideTurtle()

repeat 10000:
    x = randint(-250,250)
    y = randint(-250,250)
    setPos(x, y)
    if x > -100 and x < 100:
       setPenColor("red")
       dot(4)
    else:
       setPenColor("gray")
       dot(4)
```

► In Zwischenablage kopieren

BEDINGUNGEN MIT *OR* VERKNÜPFEN



Ein horizontaler roter Streifen genügt der Bedingung *y > -100* and *y < 100*. Punkte, die entweder im vertikalen oder horizontalen oder in beiden Gebieten liegen, erfüllen beide Bedingungen. Umgangssprachlich sagst du, dass sie im *vertikalen oder (auch) horizontalen Streifen* liegen. Du verknüpfst die beiden Bedingungen mit **or**.

Programm: **[► Online-Editor]** **[► WebTigerJython]**

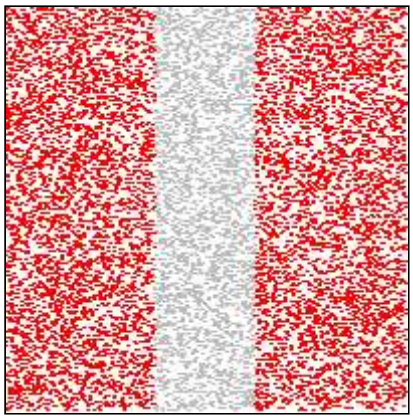
```
from gturtle import *
from random import randint

makeTurtle()
```

```
hideTurtle()

repeat 10000:
    x = randint(-250,250)
    y = randint(-250,250)
    setPos(x, y)
    if (x > -50 and x < 50) or (y > -50 and y < 50):
       setPenColor("red")
       dot(4)
    else:
       setPenColor("gray")
       dot(4)
```

► In Zwischenablage kopieren

EINE BEDINGUNG NEGIEREN

Mit dem Keyword *not* kannst du eine Bedingung negieren, das heisst, ihren Wahrheitswert umkehren. Punkte, die nicht im vertikalen Streifen liegen, genügen der Bedingung **not (x > -100 and x < 100)**, wie das folgende Programm zeigt:



Programm: **[► Online-Editor]** **[► WebTigerJython]**

```
from gturtle import *
from random import randint

makeTurtle()
hideTurtle()

repeat10000:
    x = randint(-250, 250)
    y = randint(-250, 250)
    setPos(x, y)
    if not (x > -100 and x < 100):
       setPenColor("red")
       dot(4)
    else:
       setPenColor("gray")
       dot(4)
```

► In Zwischenablage kopieren

MERKE DIR...

Die korrekte Klammerung bei Bedingungen ist sehr wichtig. Da *not* stärker bindet als *and* und *or*, musst du hier eine Klammer setzen. Am schwächsten bindet *or*.

Den grauen Streifen erhältst du logischerweise auch, wenn du verlangst, dass *x <= -50* oder *x >= 50* ist. Man kann also Bedingungen verschieden formulieren. Versuche es!

Für Zahlen gibt es folgende Vergleichsoperatoren:

<	kleiner
<=	kleiner oder gleich
==	gleich

>=	grösser oder gleich
>	grösser
!=	verschieden

Du musst dich insbesondere an die **Verdoppelung des Gleichheitszeichens** bei der Gleichheitsbedingung gewöhnen. Diese ist nötig, damit der Computer zwischen der Zuweisung und der Gleichheitsbedingung unterscheiden kann.

MEHRFACH-AUSWAHL

Um mehr als zwei Fälle zu unterscheiden, musst du im *else*-Teil wieder eine *if*-Bedingung einfügen. In Python kann man kürzer **elif** verwenden und schreibt:

```
if bed1:
    ...
elif bed2:
    ...
else:
    ...
```

Du siehst dies an folgendem Musterbeispiel, wie zufällig eine aus 5 Stiftfarben ausgewählt wird.

Programm: **[► WebTigerJython]** **[► Online-Editor]**

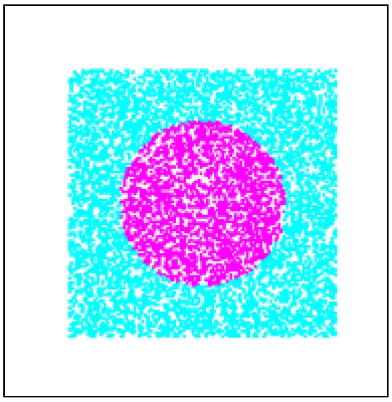
```
from gturtle import *
from random import randint

makeTurtle()
hideTurtle()
n = randint(1, 6)

if n == 1:
    setPenColor("red")
elif n == 2:
    setPenColor("yellow")
elif n == 3:
    setPenColor("magenta")
elif n == 4:
    setPenColor("green")
elif n == 5:
    setPenColor("blue")
else:
    setPenColor("black")
dot(100)
```

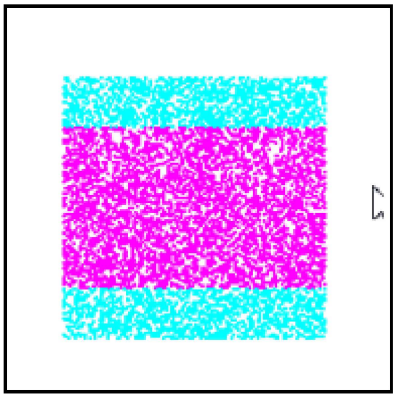
► In Zwischenablage kopieren

ZUM SELBST LÖSEN

1. Verwende Zufallszahlen zwischen -150 und 150, um folgende Grafiken zu erhalten.

   a) b)





2. Du hast gelernt, dass man mit folgendem Programm eine Spirale zeichnen kann.

```
from gturtle import *
makeTurtle()
hideTurtle()

s = 10
repeat 100:
    forward(s)
    right(90)
    s = s + 2
```

Verwende die if-else-Struktur, um eine zweifarbige Spirale zu zeichnen.

3. Verwende die Mehrfachauswahl, um eine vierfarbige Spirale zu zeichnen.