

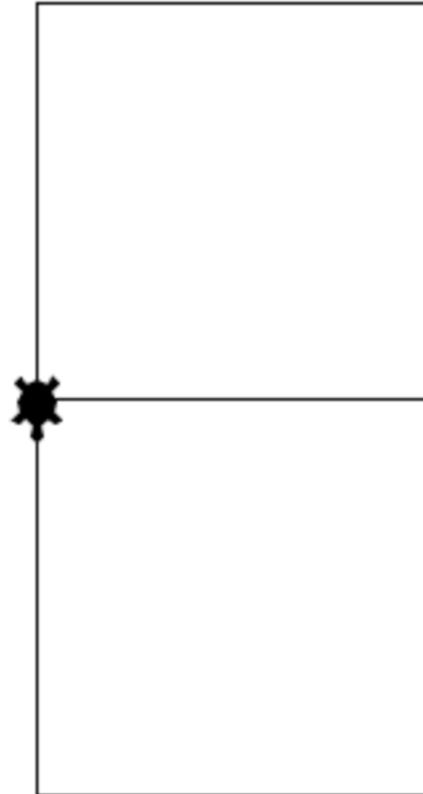
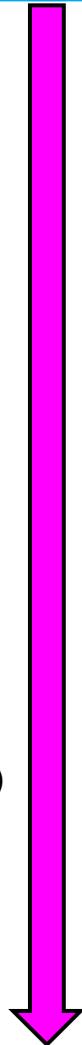
Programmieren I

Verzweigungen

Andreas Schärer, KSR

Code-Flow: Basic

```
from gturtle import *  
  
anton = Turtle()  
  
length = 150  
  
anton.left(90)  
anton.forward(length)  
anton.left(90)  
anton.forward(length)  
anton.left(90)  
anton.forward(length)  
anton.left(90)  
anton.forward(2*length)  
anton.left(90)  
anton.forward(length)  
anton.left(90)  
anton.forward(length)
```

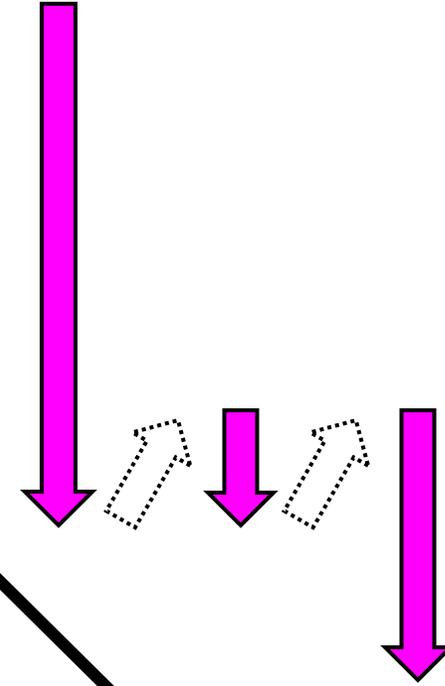


Im Normalfall wird Code von **oben nach unten**, Zeile für Zeile ausgeführt.

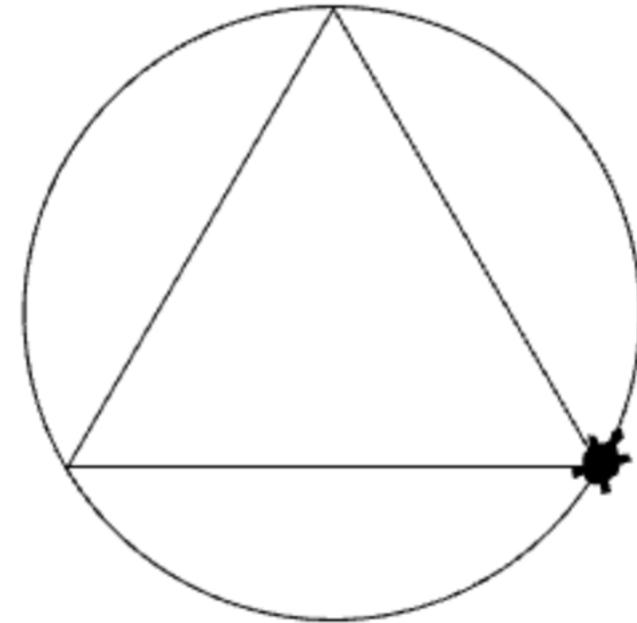
Code-Flow: Schleifen

Mit **Schleifen** (z.B. repeat) kann man einen **Codeblock mehrfach hintereinander ausführen**.

```
from gturtle import *  
  
berta = Turtle()  
  
berta.left(30)  
  
repeat 3:  
    berta.forward(200)  
    berta.left(120)  
  
berta.right(60)  
berta.leftArc(116, 360)
```



Der Codeblock, der wiederholt wird, muss **engerückt** sein!



Verzweigung: if-else

Eingabe wird in Variable `answer` gespeichert.

Benutzerin wird mit Spezialfunktion `input` nach **Eingabe** gefragt.

Beachte: Um Gleichheit zu überprüfen, muss **Doppelgleich**-Symbol `==` verwendet werden.

```
answer = input("Was gibt 37 + 5?")  
if answer == 42:  
    print("Korrekt!")  
else:  
    print("Falsch!")
```

Hier wird **Bedingung überprüft**. Falls (**if**) Bedingung, dass `answer` den Wert 42 hat, erfüllt ...

Falls (**else**) Bedingung *nicht* erfüllt, wird zugehörige Codeblock ausgeführt.

... wird zugehörige Codeblock (eingerückt!) ausgeführt.

Mit `print` Befehl kann man in Konsole eine **Ausgabe** machen. Solche Programme (ohne Turtle) nennt man **Konsolenprogramme**.

Code-Flow: Verzweigung



```
answer = input("Was gibt 37 + 5?")  
  
if answer == 42:  
    print("Korrekt!")  
else:  
    print("Falsch!")  
  
print(»Auf Wiedersehen")
```



In einer **Verzweigung (if-else)** wird eine **Bedingung überprüft**. Je nachdem, ob sie erfüllt wird oder nicht, wird unterschiedlicher Code ausgeführt.

Verzweigung: if-elif-else



elif für «else if»

Wird alles
übersprungen!

```
answer = input("Was gibt 37 + 5?")  
  
if answer == 42:  
    print("Korrekt!")  
elif answer == 40:  
    print("Fast, um 1 zu tief!")  
elif answer == 41:  
    print("Fast, um 1 zu hoch!")  
else:  
    print("Falsch!")  
  
print(»Auf Wiedersehen")
```

Sobald eine *Bedingung erfüllt* ist, werden die restlichen `elif` und `else` übersprungen.

Kann beliebig viele `elif` hintereinander haben.

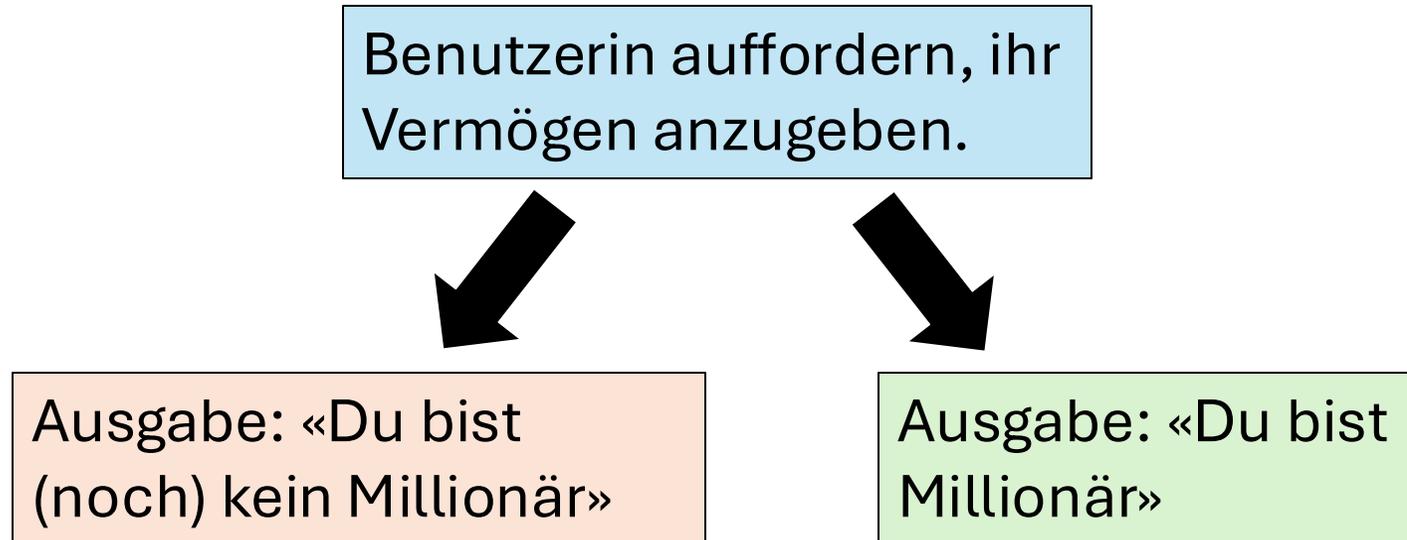
Vergleichsoperatoren

- In Beispielen oben haben immer auf Gleichheit überprüft mit Gleichheitsoperator ==
- Gibt aber noch weitere **Vergleichsoperatoren**:

Operator	Überprüft ob ...
$x == y$	x gleich y ist.
$x != y$	x <i>ungleich</i> y ist.
$x > y$	x grösser als y ist.
$x < y$	x kleiner als y ist.
$x >= y$	x grösser oder gleich y ist.
$x <= y$	x kleiner oder gleich y ist.

Beachte: '=' kommt immer *nach* <, >

Aufgabe: Wer ist Millionär?

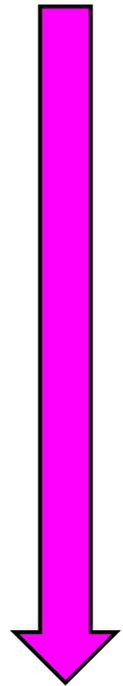


```
money = input("Wie viel Geld hast du?")

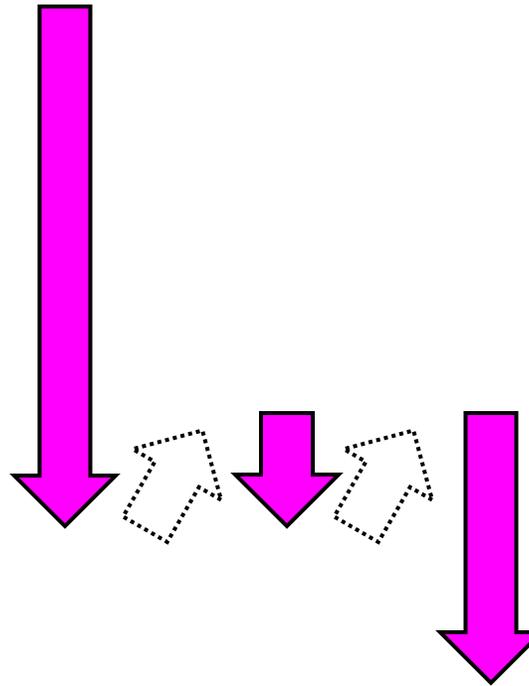
if money >= 1000000:
    print("Du bist Millionär!")
else:
    print("Du bist kein Millionär!")
```

Code Flow: Zusammenfassung

Basic



Schleife
(repeat)



Verzweigung
(if-...)

