Programmieren 2

Python Grundlagen – Informatik 1M

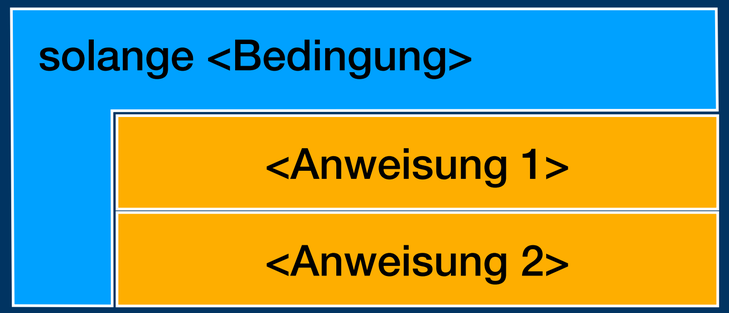
Von Struktogrammen zu Python

Anweisungen/Variablen, Schleifen und Verzweigungen

1. **Anweisung,**
   (meistens etwas mit Variablen)

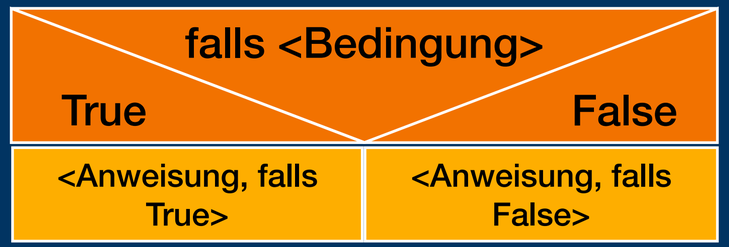

```
i = 0
text1 = “Hallo Welt”
Leonardo = Turtle()
Leonardo.forward(100)
```

2. **Schleifen**




```
while i < 7:
    print(text1)
    i = i + 1
```

3. **Verzweigung**




```
if i >= 4:
  print(i)
else:
  print(“Fehler!”)
```

Eingabe und Ausgabe

1. **Eingabe**

a = [Eingabe] (Zahl)

text1 = [Eingabe] (Text)

2. **Ausgabe**

Ausgabe a + b

Ausgabe text 1

```
a = input(“Erste Zahl eingeben”)
b = input(“Zweite Zahl eingeben”)
text1 = input(“Text eingeben”)

print(a + b)
print(text1)
```

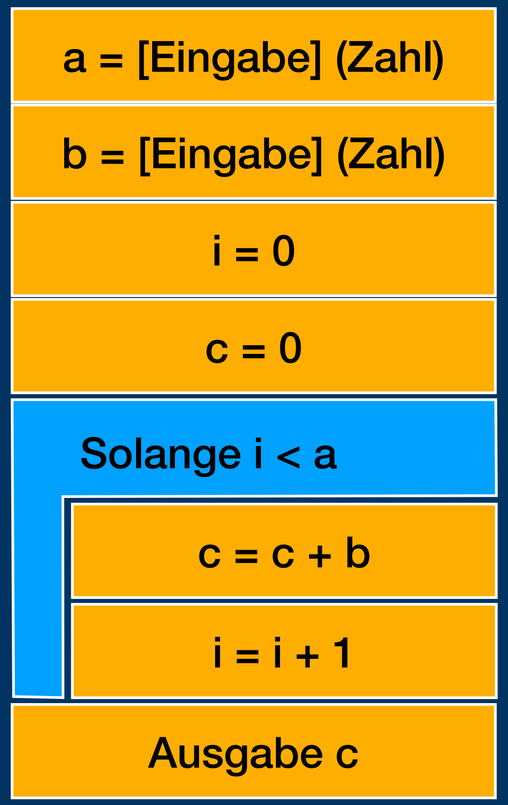
Ausprobieren:

1. TigerJython öffnen.
2. Code eingeben.
3. Speichern als input_output.py in **neuem Ordner**: …/Informatik/Programmieren2
4. Ausführen und Testen
   - Was, wenn anstelle einer Zahl ein Buchstabe/Text eingegeben wird?

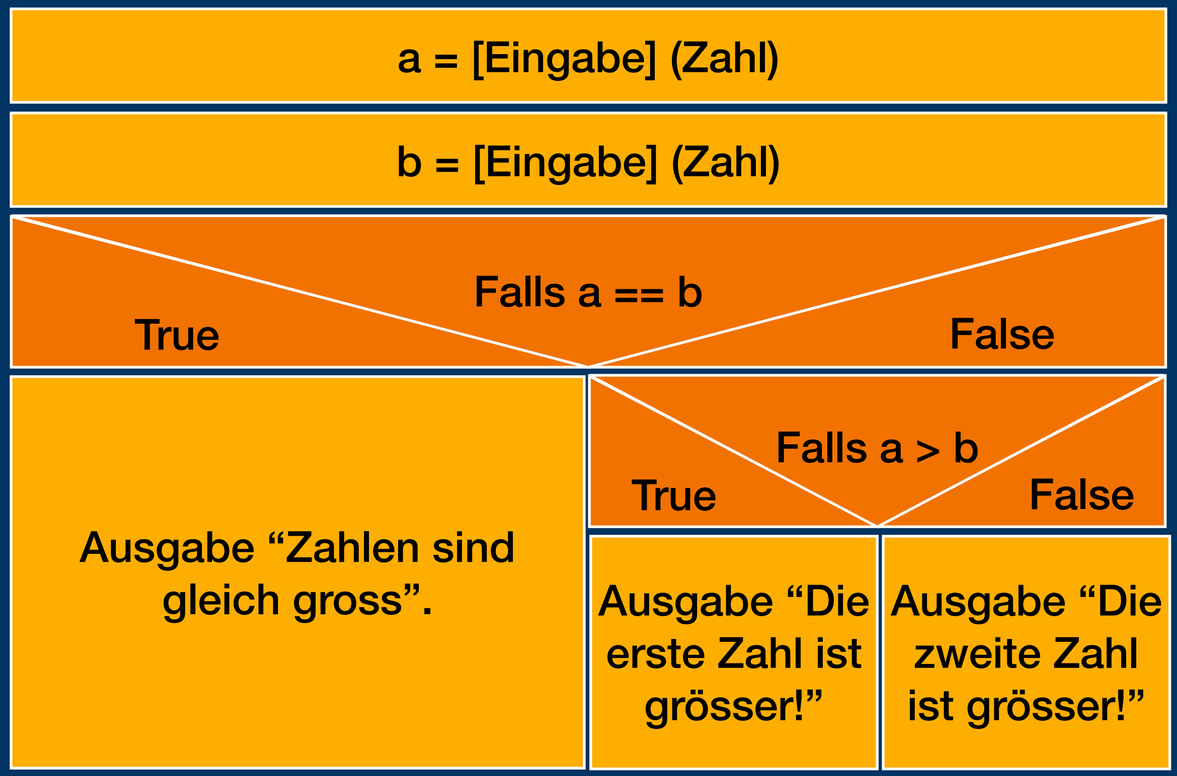
Übungen

Schreibe folgende Algorithmen in Python und teste sie. Für Hilfe siehe Wiki.

A) Multiplikation.py

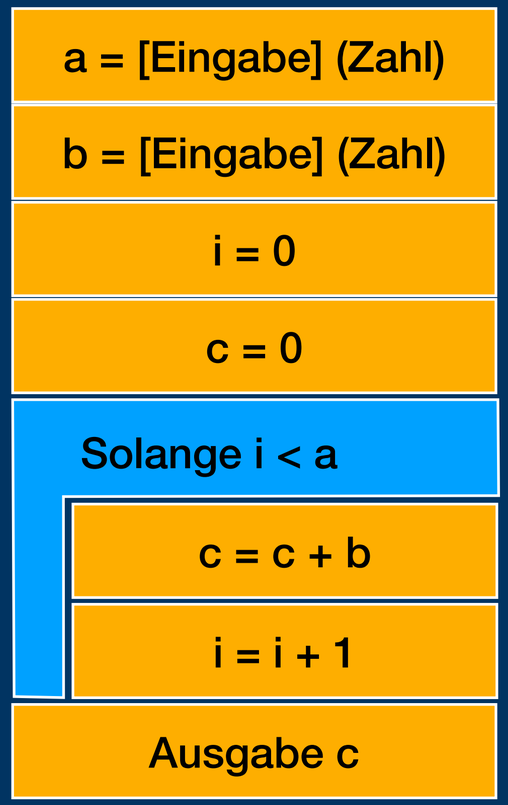


B) Vergleich.py



Lösungen

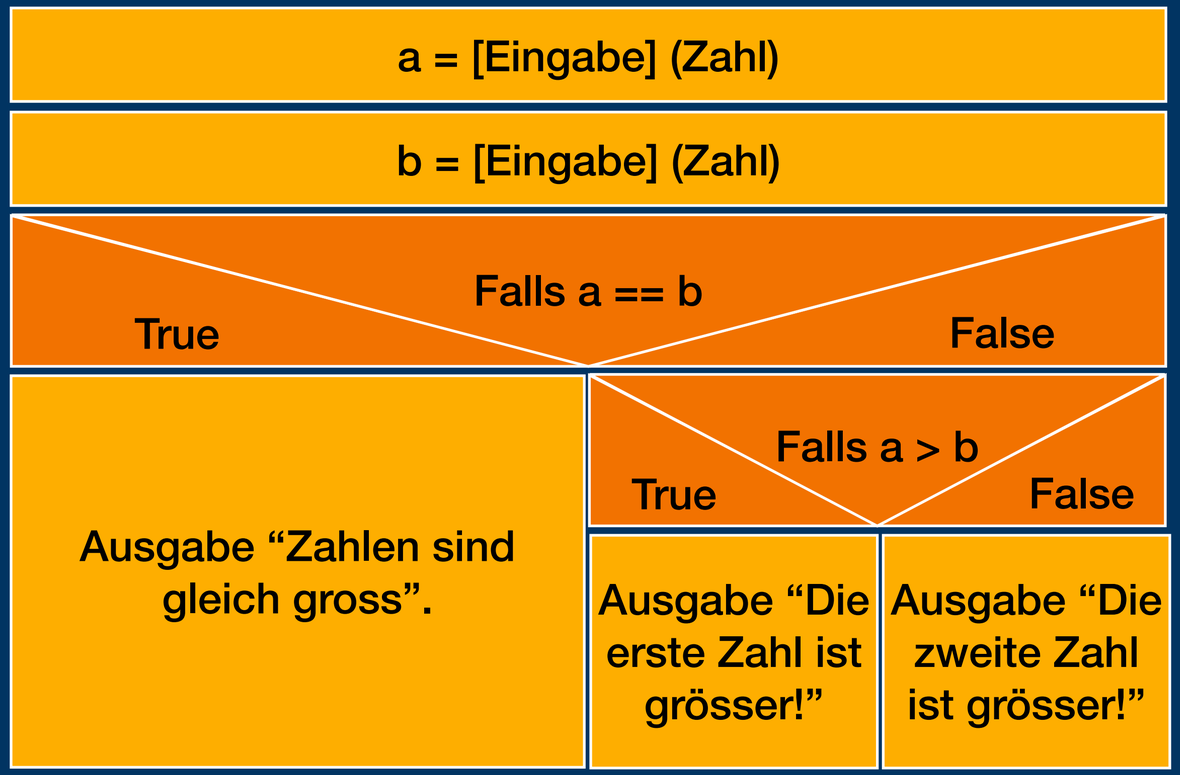
A) Multiplikation:




```
a = input(“Erste Zahl eingeben”)
b = input(“Zweite Zahl eingeben”)
i = 0
c = 0
while i < a:
  c = c + b
  i = i + 1
print(c)
```

Lösungen

B) Vergleich:




```
a = input(“Erste Zahl eingeben”)
b = input(“Zweite Zahl eingeben”)
i = 0
c = 0
if a == b:
  print(“Zahlen sind gleich gross”)
else:
  if a > b:
    print(“Die erste Zahl ist grösser”)
  else:
    print(“Die zweite Zahl ist grösser”)
```

```
…
if a == b:
  print(“Zahlen sind gleich gross”)
elif a > b:
  print(“Die erste Zahl ist grösser”)
else:
  print(“Die zweite Zahl ist grösser”)
```

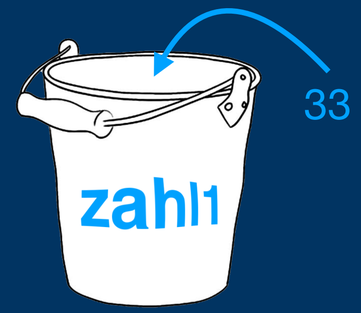
Variablen

Namen, Operationen, Datentypen

Variablen haben einen Namen und einen Wert

- **Variablen sind wie Behälter für Werte**
- **Sie können beliebige Namen haben, aber:**
  - Die Namen sollten *sprechend* sein, d.h. etwas darüber sagen, wozu die Variable da ist.
  - Keine Ziffer am Anfang.
  - Keine Zeichen wie / . : * etc.
- **Die Werte werden mit = zugewiesen**

```
zahl1 = 33
```




```
name = “Karl”
address = “Austrasse 17”
note_eng = 5.3
note_inf = 5.0
msg = “Es hat alles geklappt!”
```

Rechnen mit Variablen

Funktion	Python-Code
Addition	a+b
Subtraktion	a-b
Multiplikation	a*b
Division	a/b
Hoch (z.B. 2 hoch 5)	a**b
Wurzel (z.B. Wurzel von 2, sqrt für square-root) Achtung: braucht «import math»	math.sqrt(a)
Modulo (Rest der Ganzzahl-Division, Bsp. 17%5 = 2)	a%b

Variablen-Operationen – Was wird ausgegeben?

- **2er-Gruppen: Erst überlegen und notieren, dann ausprobieren.**

```
a = 6
b = 3
c = “Hey”

print(a+b) # 1. Ausgabe
print(a*b) # 2. Ausgabe
print(a/b) # 3. Ausgabe
print(a*c) # 4. Ausgabe
print(a%b) # 5. Ausgabe
```

```
9
18
2.0
HeyHeyHeyHeyHeyHey
0
```

Sonderfall:
“Gib a mal den Text c aus”
—> In diesem Fall kann Python eine Operation mit **Zahl UND Text** ausführen.

```
a = 6
b = 3
c = “Hey”

print(a+c) # 1. Ausgabe
```

```
TypeError: unsupported
operand type(s) for +:
'int' and 'str'
```

Aber **fast immer** entsteht ein **Problem**, wenn wir Zahlen mit Text verrechnen wollen.

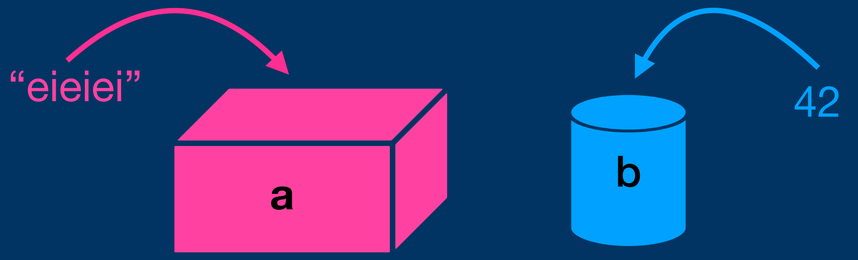
Datentypen ‘int’ und ‘str’

engl. **integer** = Ganzzahl

engl. **string** = Zeichenkette

- Der Typ (=Datentyp) einer Variable legt fest, welche *Art von Daten* darin gespeichert wird.
  - z.B, Ganzzahlen (int), Kommazahlen (float), Text (str) etc.
- (Variablen vom Typ ‘str’ brauchen mehr Speicherplatz als Variablen vom Typ ‘int’)

```
a = “eieiei”
b = 42
```



Datentypen ‘int’ und ‘str’

- **Was wird ausgegeben?**

```
zahl1 = 2
zahl2 = “4”

print(zahl1+3) # 1. Ausgabe
print(2*zahl1) # 2. Ausgabe
print(2*zahl2) # 3. Ausgabe
print(zahl2-zahl1) # 4. Ausgabe
```

```
5
4
44
TypeError…
```

- **Wird der Wert in Anführungsstriche gesetzt, heisst das für Python: Dies ist ein String str**
- **Und Text minus Zahl rechnen geht nicht.**

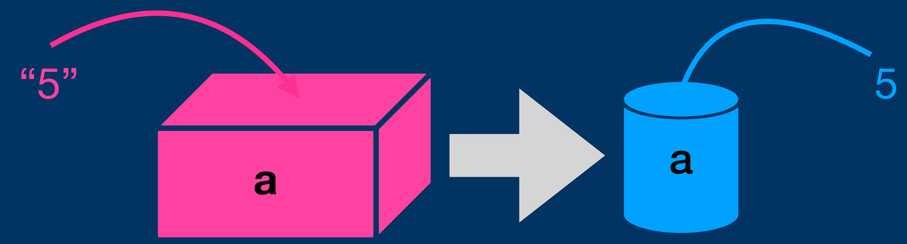
Typumwandlung mit `int()`

```
a = “5” # a ist nun ein string
b = 42 # b ist nun ein integer

a = int(a) # wandle string in integer

print(a + b)
```

```
47
```



Eingabe mit `input()`

- `input()` liefert die Eingabe der Benutzer:in
  - kann eine Aufforderung / eine Nachricht enthalten.
- `input()` liefert standardmässig einen String
- Die Funktionen `int()` bzw. `float()` konvertieren die Eingabe in eine Zahl:
  - int: integer – Ganzzahl
  - float: floating point – Fliesskommazahl

```
name = input(“Was ist dein Name?”)
```

```
age = int(input(“Wie alt bist du?”))
grade = float(input(“Was ist deine Note?”))
```

Formatierte Ausgabe

- Wenn wir Variablen nicht als blanke Zahlen sondern in Text eingebettet ausgeben wollen, dann können wir die Funktion `format()` nutzen:

```
age = int(input(“Wie alt bist du?”))
days = age * 365
hours = days * 24

print(“Du bist mindestens {0} Tage oder {1} Stunden alt”.format(days,hours))
```

- —> Ausprobieren.

Aufgaben

- Lese im Wiki “Programmieren II: Variablen, Verzweigungen, Schleifen” das erste Kapitel “Variablen & Mathematik”. Wenn dir etwas unklar ist: Frage nach!
- Löse die Aufgaben D1 bis D4.

Kleine Übung zu Variablen

- Schreibe ein Programm, das Folgendes kann:
- Benutzer:in kann nacheinander drei Noten (auf eine Kommastelle genau) eingeben.
- Das Programm gibt aus: “Der Notenschnitt ist …”.
  - —> Sprechende Variablen verwenden.
  - —> Speichern als Notenschnitt.py (im Ordner Programmieren2)

Verzweigungen

if, else, elif – Vergleichsoperatoren

if … else

- Allgemeiner Aufbau:

```
if BEDINGUNG:
    # Codeblock, falls BEDINGUNG erfüllt ist
else:
    # Codeblock, falls BEDINGUNG nicht erfüllt ist
```

- Beispiel:

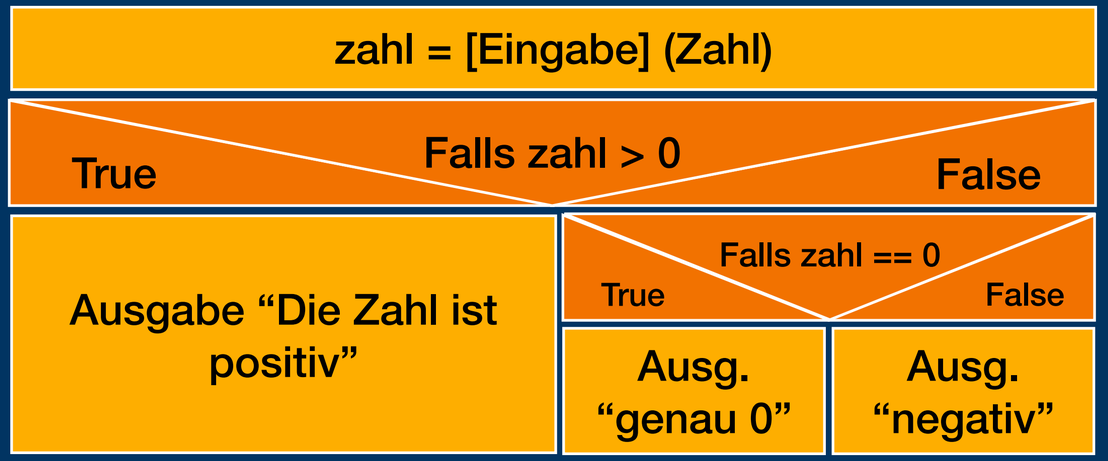
```
zahl = int(input("Gib eine Zahl ein."))
if zahl >= 0:
    print("Die Zahl ist positiv")
else:
    print("Die Zahl ist negativ")
```

Vergleichsoperatoren

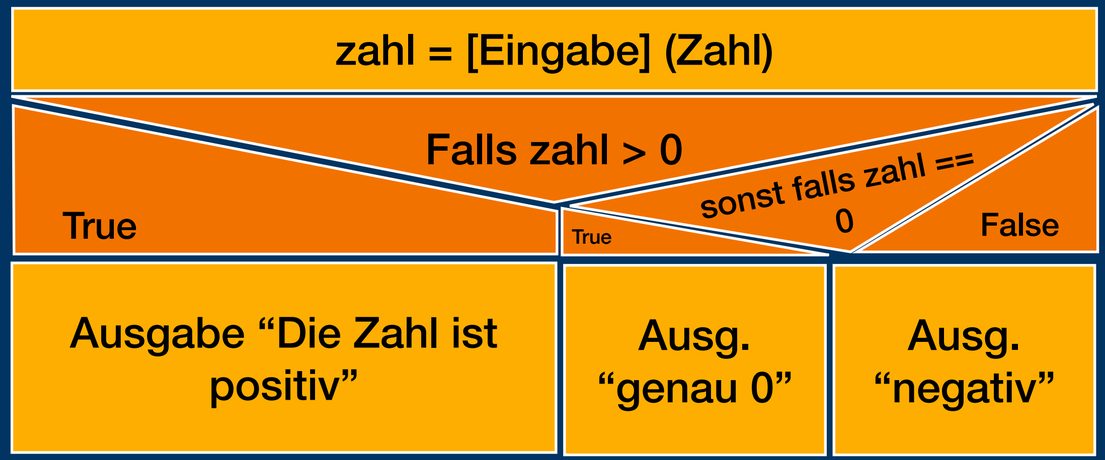
Operator	Beispiel	Erklärung
==	x == 4	x ist Zahl und hat Wert von genau 4
	s == "Hallo"	s ist String und hat genau den Inhalt „Hallo“
!=	x != 4	x ist NICHT eine Zahl vom Wert
>	x > 5	x ist Zahl grösser als 5
>=	x >= 5	x ist Zahl grösser gleich 5
<	x < 5	x ist Zahl kleiner als 5
<=	x <= 5	x ist Zahl kleiner gleich 5

if … else – elif ist gleich else: if

```
zahl = int(input(“Gib eine Zahl ein.”))
if zahl > 0:
    print(“Die Zahl ist positiv”)
else:
    if zahl == 0:
        print(“Die Zahl ist genau 0”)
    else:
        print(“Die Zahl ist negativ”)
```



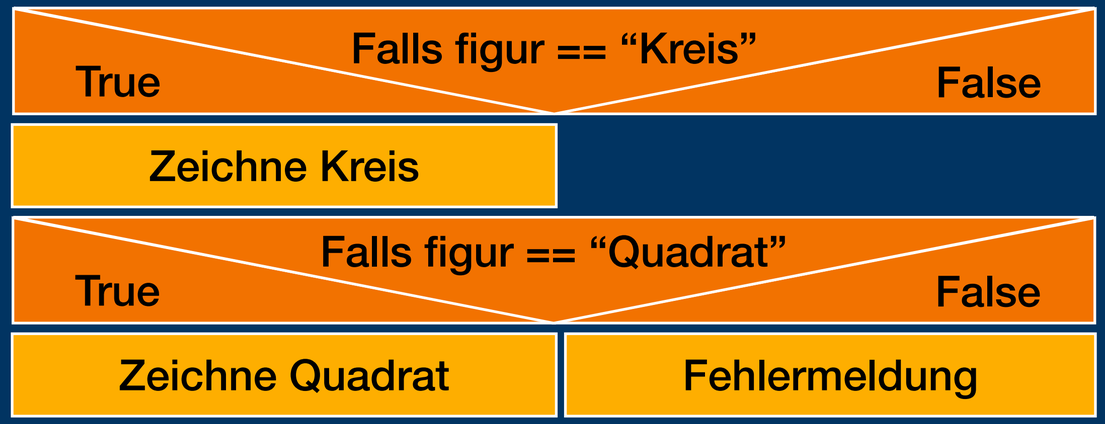

```
zahl = int(input(“Gib eine Zahl ein.”))
if zahl > 0:
    print(“Die Zahl ist positiv”)
elif zahl == 0:
    print(“Die Zahl ist genau 0”)
else:
    print(“Die Zahl ist negativ”)
```



if, else und elif – Unterschiede

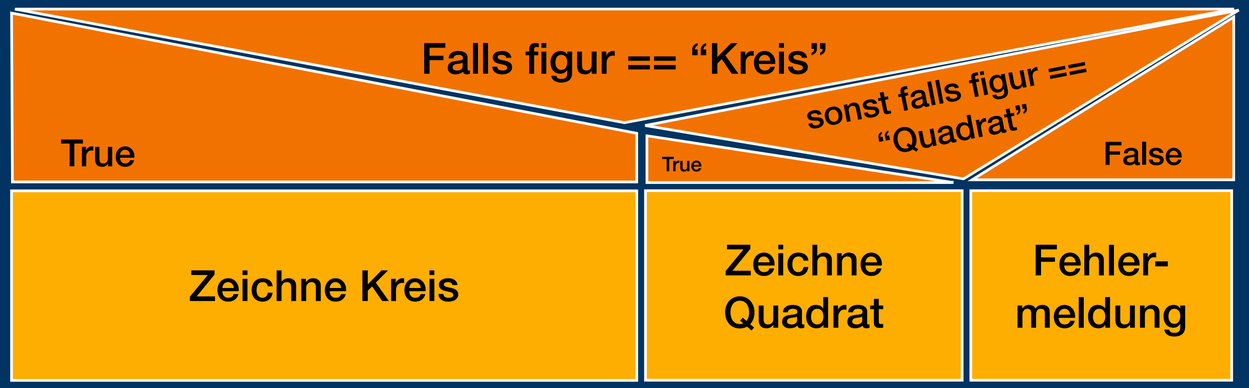
if-if-else: Zwei if-Verzweigungen nacheinander:
(bei der ersten ist else nicht angeführt, das ist oft so)

```
if figur == "Kreis":
    Michelangelo.rightArc(50,360)
if figur == "Quadrat":
    repeat 4:
        Michelangelo.forward(50)
        Michelangelo.right(90)
else:
    print("Gib entweder 'Quadrat' oder 'Kreis' ein")
```



if-elif-else: Eine if-Verzweigung mit drei Ästen.

```
if figur == "Kreis":
    Michelangelo.rightArc(50,360)
elif figur == "Quadrat":
    repeat 4:
        Michelangelo.forward(50)
        Michelangelo.right(90)
else:
    print("Gib entweder 'Figur' oder 'Kreis' ein")
```



Aufgaben

- Lese im Wiki “Programmieren II: Variablen, Verzweigungen, Schleifen” das zweite Kapitel “Logik: if-else”.
- Löse die Aufgaben E1 bis E3.

While-Schleife

die bessere repeat-Schleife

while …

- Allgemeiner Aufbau:

```
while BEDINGUNG:
    ... # Diese Code-Zeilen werden solange ausgeführt,
    ... # wie die BEDINGUNG TRUE (also erfüllt) ist.

... # Diese Code-Zeilen werden ausgeführt, sobald
... # wie die BEDINGUNG FALSE (also nicht mehr erfüllt) ist.
```

- Beispiel:

```
zahl = 1
while zahl >= 0:
  zahl = int(input(“Gib eine positive Zahl ein.”))

print(“Game Over: Du hast etwas eingegeben, was keine positive Zahl ist.”)
```

while True (Endlosschleife)

- Allgemeiner Aufbau:

```
while True:
    ... # Diese Code-Zeilen werden “endlos” wiederholt.

... # Diese Zeilen werden nicht ausgeführt – Ausnahme: break-Befehl in while...
```

- Beispiel Endlosschleife

```
while True:
    zahl1 = random.randint(0,10) # zufällige Zahl zwischen 0 und 10
    zahl2 = random.randint(0,10) # zufällige Zahl zwischen 0 und 10
    antwort = int(input("Was ergibt {} plus {}?".format(zahl1, zahl2)))
    if antwort != zahl1 + zahl2:
        break
print(“Game Over!”)
```

Aufgaben

- Lese im Wiki “Programmieren II: Variablen, Verzweigungen, Schleifen” das zweite Kapitel “While-Schleife”.
  - Probiere die Beispiele aus.
- Löse die Aufgaben F1 bis F5.