

Merkblatt - Vergleichsoperatoren

September 22, 2017

0.0.1 Vergleichsoperatoren

Um die Bedingungen in den if-else-Strukturen besser zu verstehen, schauen wir uns **Vergleichsoperationen** an. Das wird uns helfen eine Vielfalt an Bedingungen formulieren zu können.

0.0.2 Ungleichheitsoperatoren und Bools

```
In [1]: if 6 < 5:
        print("JA")
```

Warum kommt es zu keiner Ausgabe mit print()? Schauen wir uns doch Ausdrücke mit Ungleichheitszeichen (<,>) im Detail an:

```
In [5]: print(6 < 5)
        print(5 < 6)
```

False
True

True und **False** (die Großschreibung beachten) sind weitere *feststehende Ausdrücke* in Python. Neben Strings (Zeichenketten), Ganzzahlen (Integer) und Fließkommazahlen (Floats) bilden sie einen weiteren Datentyp - den **Bool**.

```
In [7]: b = False
        print(b)
```

False

Genauer wird eine if-Bedingung also nur dann ausgeführt, wenn nach dem if ein Bool mit dem Wert True steht.

```
In [8]: result = 5 < 6
        if result:
            print("5 ist kleiner als 6")
```

5 ist kleiner als 6

```
In [10]: print(5 < 6)
```

True

0.0.3 Der Gleichheitsoperator

Neben Ungleichheiten können wir natürlich auch Gleichheiten abfragen, mit ==

```
In [11]: print(5 == 5)
         print(5 == 4)
```

```
True
False
```

```
In [19]: if 5 == 5:
         print("5 ist 5")
```

```
5 ist 5
```

Mittels des Gleichheitsoperators können wir auch die Zustände *grösser gleich* (>=) und *kleiner gleich* (<=) abfragen

```
In [18]: print(5 < 5)
         print(5 <= 5)
         print(5 >= 5)
```

```
False
True
True
```

0.0.4 Strings vergleichen

Wir können nicht nur Zahlen miteinander vergleichen, sondern auch Strings.

```
In [15]: word = "Hallo"
         print(word == "Hallo")
         print(word == "Welt")
```

```
True
False
```

0.0.5 Der Ungleichheitsoperator

Auf Ungleichheit checken wir mit dem Zeichen !=

```
In [16]: word = "Hallo"
         print(word != "Hallo")
         print(word != "Welt")
```

```
False
True
```

Auch Zahlen kann man auf Ungleichheit hin miteinander vergleichen:

```
In [3]: zahl = 4  
        print(zahl != 4)  
        print(zahl != 5.5)
```

False

True

0.0.6 Spiel doch jetzt ein wenig herum mit dem was du gelernt hast:

- formuliere eigene Vergleichsoperationen mit den Operatoren aus dieser Lektion :-)

```
In [ ]:
```