

Python - Kapitel 1

Datentypen:

Was sind Datentypen ?

Datentypen sind in unserem Fall erstmal nur Typen von Werten mit denen wir arbeiten.
Mehr ist für unsere Zwecke erstmal nicht wichtig

Wir verwenden folgende Datentypen:

Integer: Mit Integers können wir ganze Zahlen (also -1,0,1,2,...565754,...) benutzen.
Wie erstellen wir einen **Integer** in Python ?

```
unserInt = 5
```

Float: Mit einem Double können wir Zahlen mit Komma darstellen (also 1.2, 6.7).
Wie erstellen wir einen Float in Python ?

```
unserFloat = 1.5
```

String: Mit Strings können wir jeden möglichen Text speichern. Wichtig ist dabei, dass dieser Text
immer in den "" steht.
Wie erstellen wir einen **String** in Python ?

```
unserString = "Hallo"
```

Wie funktioniert das Erstellen ?

```
unserInt = 5
```

1. Wir erstellen eine variable indem wir irgendwo ein neues Wort hinschreiben.
2. Das „=“ Zeichen ist hier eine Zuweisung. Es gilt der Satz „unserInt **hat den Wert** 5“
3. Das hinter der Zuweisung ist der Wert mit dem es zugewiesen wird. Hier 5.

Boolean: Mit einem Boolean können wir nur zwei Werte darstellen. Entweder „True“ oder „False“

```
unserErsterBool = True
```

```
unserZweiterbool = False
```

Kommentare:

In jeder Programmiersprache gibt es Kommentare. Kommentare sind dafür da, dass ihr Text schreiben könnt, der aber bei der Ausführung einfach ignoriert wird. Das hilft dabei den Code übersichtlicher zu machen und um Merkhilfen zu notieren.

```
# Es gibt auch Kommentare
```

1. Für Kommentare über eine Zeile muss einfach nur ein # davor gesetzt werden.

Bedingungen:

Für Programme sind Bedingungen immens wichtig. Mit Bedingungen wird gesteuert und auf Eingaben reagiert. (Diese Erklärung ist nicht so wichtig.):

Hier haben wir eine einfache if - Verzweigung. Nur wenn unsere Bedingung gilt soll etwas gemacht werden.
(natürlich funktioniert dieser Code nicht. Statt Bedingung muss wirklich eine Bedingung gelten.)

```
if Bedingung:  
    print("Die Bedingung ist erfüllt.")
```

Boolean: Wir haben bei unseren Typen noch einen vergessen, nämlich den Boolean. Der Boolean kann nur die Werte **False** (für nein) und **True** (für ja) annehmen.

-> Jede Bedingung wertet entweder zu False oder zu True aus.

Was heißt das für uns ?

Wir wissen damit wie unsere if - Verzweigung funktioniert und das der Code links auch klappt. Wir können also alles als if Bedingung verwenden was als Ergebnis einen dieser Booleans zurück gibt.

Python - Kapitel 1

Wie schreiben wir denn unsere Bedingungen:
Mit Vergleichen ?

```
if unserInt < 6:  
    print("Unsere Zahl ist kleiner als 6")
```

Jetzt überprüft unser Programm ob unsere Variable x kleiner als fünf ist.

Bei Ja -> führt das Programm alles aus was eingerastet ist, das heißt alles was erst nach ein paar Leerzeichen unter if beginnt. **Drückt dafür einfach Tabulator auf eurer Tastatur.**

Bei Nein -> springt an die nächste Stelle, die nicht eingerastet ist.

Wir kennen jetzt also den Satz „Wenn das passiert mache“
Den Satz können wir erweitern mit „Wenn das passiert mache, Sonst mache wenn das passiert, sonst mache“

Was macht das Programm diesmal ?

```
if Bedingung:  
    print("Die Bedingung ist erfüllt.")  
elif ZweiteBedingung:  
    print("nur die zweite Bedingung ist erfüllt.")  
else:  
    print("Keine der Bedingungen ist erfüllt.")
```

Wichtig: Es gilt **Bedingung = False** und **ZweiteBedingung = False** hier.

Was macht dieses Programm?

1. Es guckt bei dem 1. **If** ob dies gilt.
2. Da **Bedingung** False ist gilt das nicht und das Programm geht zu elif
3. Da **ZweiteBedingung** auch False ist springt das Programm an else
4. Else wird immer ausgeführt, wenn das Programm hier ankommt.
5. Es wird geschrieben „Keine der Bedingungen ist erfüllt.“

Python - Kapitel 1

Aufgabe 1

a) Trage ein welche Werte v,w,x,y,z am Ende haben werden...

```
v = 20
w = 3
x = 3
y = 4
z = 10

if x < v:
    x = x + v
    z = 4
elif x > 5:
    y = y + x
else:
    w = 3
    v = 3
```

v = _____

w = _____

x = _____

y = _____

z = _____

b)

```
v = 1
w = 2
x = 3
y = 4
z = 5

if x > 5:
    x = x + v
    z = 4
elif x > 3:
    y = y + x
    v = v * 2
    z = z * 2
else:
    w = 3
    v = 3

    if (x == y) & (x == v):
        x = v * w

y = 3
z = 3
```

v = _____

w = _____

x = _____

y = _____

z = _____

c)

```

v = 1
w = 2
x = 3
y = 4
z = 5

if x < 5:
    if x < 4:
        if x < 3:
            if x < 2:
                if x < 1:
                    x = x * 1
                else:
                    x = x + 1
            else:
                x = x + 2
        else:
            x = x + 3
    else:
        x = x + 4

v = x + 1
w = x + 2
y = x + 3
z = x + 4

```

v = _____

w = _____

x = _____

y = _____

z = _____