

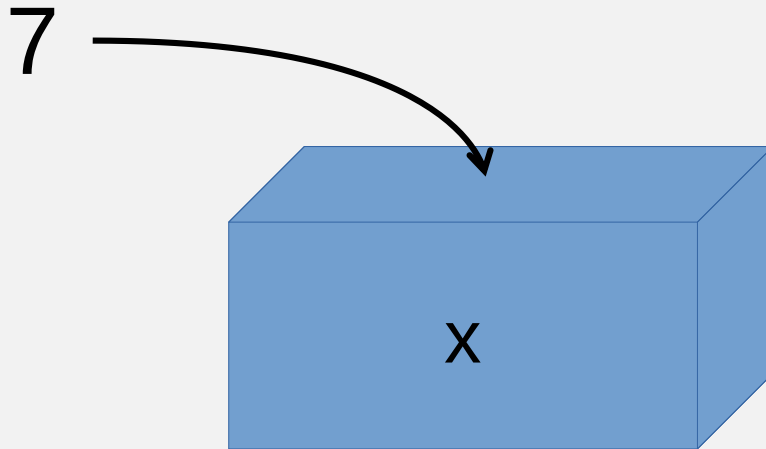
# DATEN, DATENTYPEN EINGABE, AUSGABE

#Grundlagen

# VARIABLEN

- können ihren Wert ändern (sind variabel)  
Spezialfall: Konstanten
- werden als Platzhalter verwendet
- können Informationen speichern
- haben einen Namen, einen Wert und einen Typ
- haben einen Gültigkeitsbereich

# IDEE



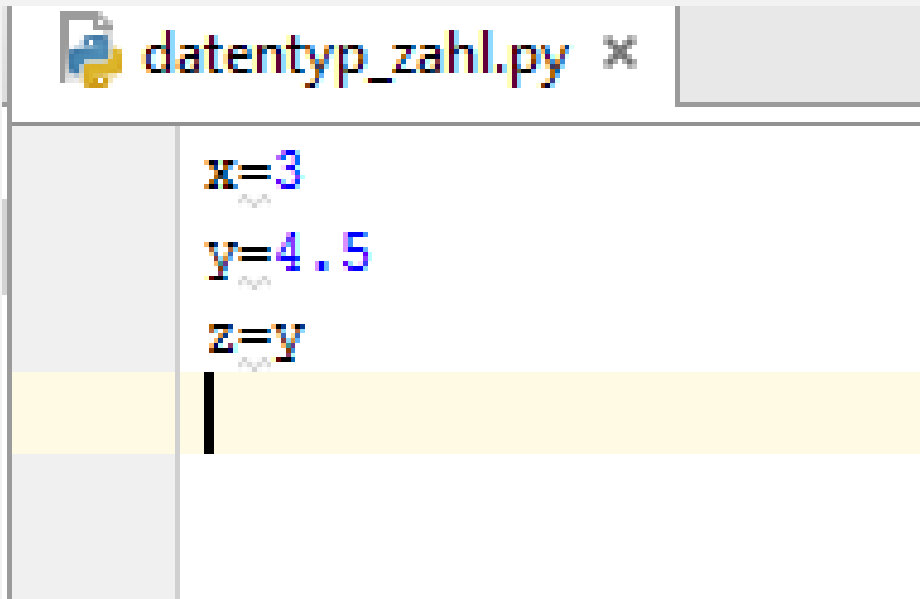
In Python:

`x=7`

# ÜBERSICHT DATENTYPEN (UV)

- Zahlen
  - Ganzzahlig (Integer)
  - Dezimalbrüche (Float)
- Sequentielle Datentypen
  - Text (Strings)
  - Listen
    - In einer Liste kann eine Folge beliebiger Objekte gespeichert werden, zum Beispiel Strings, Integers, Float-Zahlen aber auch Listen und Tupel selbst

# DATENTYP ZAHL



```
datentyp_zahl.py x
x=3
y=4.5
z=y
|
```

## Allgemein:

Name\_der\_Variable=Wert

## Hinweis:

= Zuweisungsoperator (kein  
Gleichheitszeichen)



# BERECHNUNGEN

Operation	Zeichen	Beispiel
Addition	+	3+5
Subtraktion	-	3-5
Division	/	3/5
Multiplikation	*	3*5
Potenzieren	**	3**5
Ganzzahlige Division	//	16//3
Rest der ganzzahligen Division (modulo)	%	16%3

# EINFACHE EINGABE

- Eingabe mittels input()
  - input() liefert immer Zeichenkette
  - gegebenenfalls muss eine Typumwandlung in das gewünschte Datenformat durchgeführt werden
    - int() – Zeichenkette nach Integer
    - float() – String nach Gleitkommazahl
    - str() – Zahl nach Zeichenkette

```
eingabe=input ("Deine Zahl?")  
eingabe2=int (input ("Deine Zahl?"))  
eingabe3=float (input ("Deine Zahl?"))
```

# EINFACHE AUSGABE

- Ausgabe mittels print()

```
print(23)
print("Hallo")
eingabe=input("Dein Name?")
print("Hallo", eingabe)
```



# AUFGABE I - PROGRAMM

- Schreibe ein Programm, das zwei Zahlen einliest und die Summe sowie das Produkt ausgibt. Formatiere die Ausgabe sinnvoll:  
Das Produkt von ... und ... ist ...

```
#Das ist ein Kommentar
#Eingabe
zahl1=int(input("Zahl 1?"))
zahl2=int(input("Zahl 2?"))
#Verarbeitung
ergebnis=zahl1 + zahl2
#Ausgabe
print("Die Summe von",zahl1, "und ", zahl2, "ist: ", ergebnis)
```

# ZEICHENKETTEN

- Zeichenketten sind Folge gleichartiger Elemente (Zeichen)
- Zugriff auf einzelne Zeichen:

```
zeichenkette="Hallo Kurs"
```

```
print (zeichenkette[0])  
print (zeichenkette[6])  
print (zeichenkette[-1])
```

0	1	2	3	4	5	6	7	8	9
H	a	l	l	o		K	u	r	s
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

# SLICING

- Welche Auswirkung hat der folgende Zugriff?

zeichenkette[1:3]

zeichenkette[4:8]

zeichenkette[:7]

zeichenkette[3:]

- → Ausgabe von Teilzeichenketten von/ab/bis bestimmten Index

# LÄNGEN, VERKETTUNG

- Länge mittels len()

```
print(len(zeichenkette))
```

- Verkettung mit Operator +

```
vorname="Thomas"  
nachname="Müller"
```

```
print(vorname + nachname)  
print(vorname + " " + nachname)
```

# LISTEN

- Sequenz gleichartiger Objekte
  - Liste von Zahlen
  - Liste von Zeichen (Zeichenketten)
  - Liste von Listen

# BEISPIELE

```
liste1=[1,2,3,4,5]  
liste2=["grün", "rot", "blau"]  
liste3=[[1], [1,2], [1,2,3]]
```

```
print(liste1[0])  
print(liste2[1])  
print(liste3[2])
```

1

rot

[1, 2, 3]

# LISTENOPERATIONEN

- → Handout
- Aufgabe:
  - Lesen Sie 5 Zahlen ein und speichern Sie die Werte in einer Liste
  - Berechnen Sie das arithmetische Mittel der Werte und geben Sie die Liste und das arithmetische Mittel aus.