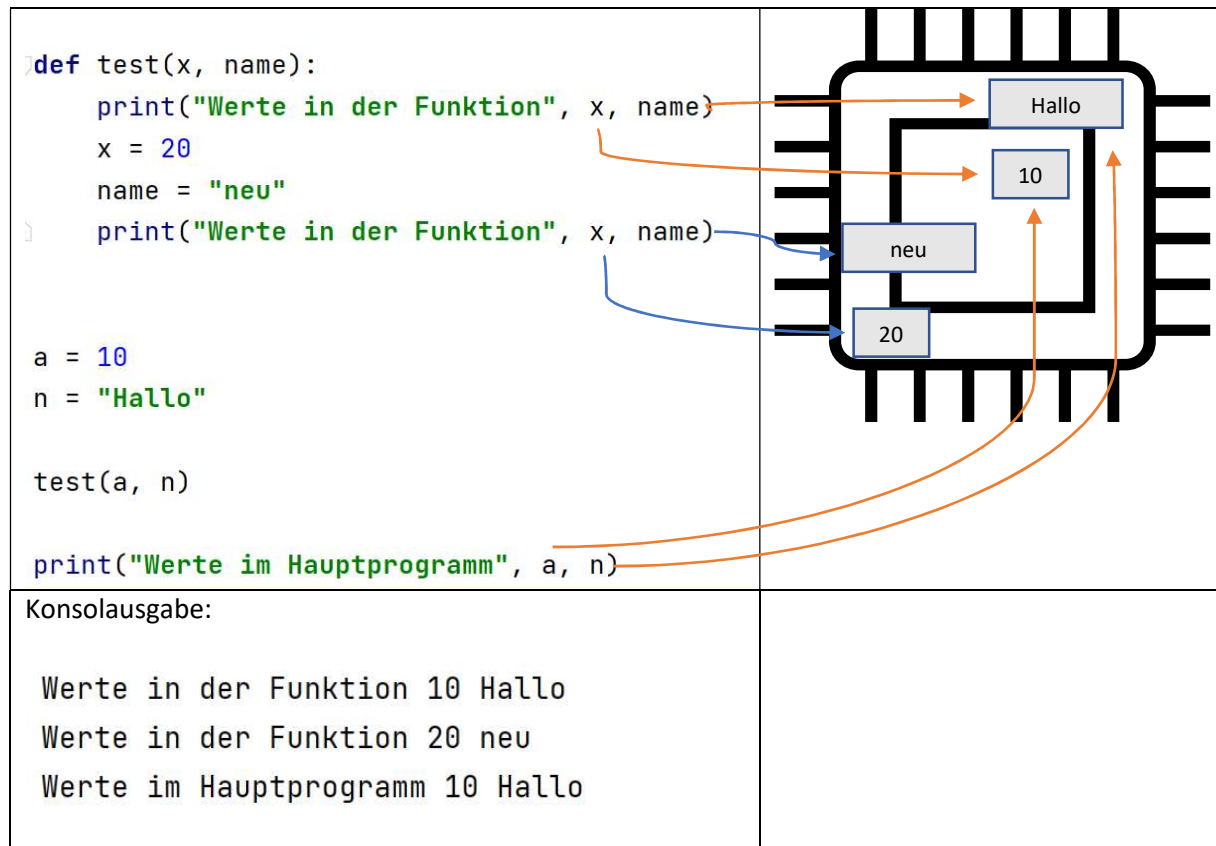


9 Methoden, Prozeduren und Funktionen

- Definition:
 - Prozeduren
 - Prozeduren geben keinen Wert an den Aufrufer zurück
 - Prozeduren werden in Python wie Funktionen definiert
 - es wird *None* zurückgegeben
 - Funktionen
 - Funktionen geben ein Ergebnis an den Aufrufer zurück
 - Funktionen werden mit dem Schlüsselwort **def** definiert
 - Methoden
 - Methoden lesen oder verändern Attributwerte eines Objektes
 - Methoden sind in Klassen gekapselt
 - Methoden werden wie Funktionen mit dem Schlüsselwort **def** definiert
- wiederkehrende Programmabschnitte lassen sich auslagern
- sie sind dann nur noch aufzurufen
- Funktionen übernehmen keinen, einen oder mehrere Werte und arbeiten dann mit diesen Werten
- Funktionen können Werte zurückgeben
- Funktionen können in Funktionen aufgerufen werden
- Vorteile
 - Steigerung der Effizienz bei der Programmierung durch Wiederverwendbarkeit von Funktionen
 - Vermeidung von Fehlern durch Aufruf von geprüften Funktionen
 - Verbesserung der Übersichtlichkeit des Quellcodes
 - Grundlage des modularen Programmaufbaus

9.1 Parameter einer Funktion

- eine Funktion kann beliebig viele mit Kommata getrennte Parameter haben
- beim Aufruf der Funktion werden Verweise auf Werte übergeben
- in der Funktion zeigen die Parameter auf die selben Werte im Speicher
- wird einem Parameternamen in der Funktion ein neuer Werte zugewiesen, wird im Speicher ein neuer Wert angelegt – der Name im Hauptprogramm hat mit dieser Änderung nichts zu tun



- die Reihenfolge der Parameter muss bei Aufruf der Funktion und der Übergabe von Werten eingehalten werden
- wird die Reihenfolge nicht eingehalten, kann es zu Fehlern und ungewollten Effekten kommen
- es gibt die Möglichkeit, die Übergabe der Parameter unabhängig von der Reihenfolge zu machen – das geschieht mit den Schlüsselwortparametern

9.2 Rückgabewert einer Funktion

- Funktionen in Python geben immer einen Wert an die aufrufende Stelle zurück
 - auch wenn es nicht explizit angegeben ist, dann wird der Wert *None* (für *keinen Wert*) zurückgegeben
 - eine Rückgabe selbst wird in der Funktion mit dem Schlüsselwort **return** eingeleitet
 - die Funktion sollte so aufgebaut sein, dass die Rückgabe am Ende der Funktionsanweisungen erfolgt – damit der Ablauf der Funktionsanweisungen klarer strukturiert ist
-
- Beispiele:

```
# Defintion der Funktion rueckgabe
def rueckgabe(x):
    if x < 100:
        rueckgabewert = 0
    else:
        rueckgabewert = x + 50
    return rueckgabewert

# Aufruf der Funktion rueckgabe mit dem Wert 150
# die Funktion liefert den Wert 200 zurück
a = rueckgabe(150)

print("Rückgabewert von der Funktion rueckgabe: ", a)
```

Rückgabewert von der Funktion rueckgabe: 200

9.2.1 Mehrere Rückgabewerte einer Funktion

- Funktionen in Python können mehrere Rückgabewerte liefern
- die Funktion liefert dann ein Tupel mit Werten zurück
- die Rückgabe selbst wird in der Funktion mit dem Schlüsselwort **return** eingeleitet
- Auswirkung, falls die Werte der Parameter in der Funktion verändert werden:
 - für Zahlen (int, float) und Zeichenketten (string):
 - bei Veränderung wird eine Kopie des Objektes angelegt
 - eine Veränderung der Kopie hat keine Auswirkung auf das Original
 - für Listen und Dictionaries:
 - es wird eine Referenz auf das Objekt übergeben
 - Veränderungen wirken sich direkt auf das Objekt aus

- Beispiele:

```
# Funktion mit Parametern
# es lassen sich beliebig viele Parameter definieren

def fkt(a, b, c , d, e, f):
    print(f"Werte in der Funktion {a}, {b}, {c}, {d}, {e}, {f}")
    a = 20
    b = 25.0
    c = "Hello"
    d[1] = "one"
    e[0] = 0
    # f[0:1] = 0 führt zu dem Fehler TypeError: 'tuple' object does not
    support item assignment
    print(f"Werte in der Funktion {a}, {b}, {c}, {d}, {e}, {f}")

i = 10
f = 12.5
s = "Hallo"
d = {1: "eins", 2: "zwei", 3: "drei"}
l = [1, 2, 3]
t = (1, 2, 3)

print(f"vor dem Aufruf {i}, {f}, {s}, {d}, {l}, {t}")

fkt(i, f, s, d, l, t)

print(f"nach dem Aufruf {i}, {f}, {s}, {d}, {l}, {t}")

vor dem Aufruf 10, 12.5, Hallo, {1: 'eins', 2: 'zwei', 3: 'drei'}, [1,
2, 3], (1, 2, 3)

Werte in der Funktion 10, 12.5, Hallo, {1: 'eins', 2: 'zwei', 3:
'drei'}, [1, 2, 3], (1, 2, 3)

Werte in der Funktion 20, 25.0, Hello, {1: 'one', 2: 'zwei', 3: 'drei'},
[0, 2, 3], (1, 2, 3)

nach dem Aufruf 10, 12.5, Hallo, {1: 'one', 2: 'zwei', 3: 'drei'}, [0,
2, 3], (1, 2, 3)
```

Aufgabe

1. BMI-Rechner

Es ist eine Funktion zur Berechnung des BMI zu erstellen. Der BMI errechnet sich nach der Formel

$$BMI = \text{Körpergewicht in kg} / (\text{Körpergröße in m} * \text{Körpergröße in m})$$

Die Funktion ist mit verschiedenen Werten zu testen.

2. Umfang und Fläche eines Rechtecks

Zur Berechnung des Umfangs und der Fläche eines Rechtecks sind zwei Funktionen zu definieren. Die Funktionen sind mit verschiedenen Werten zu testen.

3. Kartenspiel

Die Verteilung der Karten ist in eine Funktion auszulagern. Als Parameter sind die Anzahl der Mitspieler und die Anzahl der Karten, die jeder Mitspieler erhalten soll mit anzugeben.