

## 9.7 Rekursive Funktionen

- die Idee eines rekursiven Algorithmus ist, einen Algorithmus durch sich selbst zu beschreiben
- der Algorithmus muss an einer definierten Stelle beendet werden, sonst ist der Algorithmus unendlich
- z.B. Fakultätsberechnung:
  - für alle  $x > 0$ :  $x! = x * (x-1)$
  - für  $x = 0$ :  $0! = 1$
- Beispiel:

*# Definition der rekursiven Funktion zur Fakultätsberechnung*

```
def compfact(x):  
    if x > 0:  
        return x * compfact(x-1)  
    else:  
        return 1
```

*# Berechnung der Fakultäten für die Zahlen 0 bis 9*

```
for i in range(0,10):  
    print(f"{i}! = {compfact(i)}")
```

```
0! = 1  
1! = 1  
2! = 2  
3! = 6  
4! = 24  
5! = 120  
6! = 720  
7! = 5040  
8! = 40320  
9! = 362880
```

## Aufgabe

### 1. Countdown

Es ist eine Funktion zu definieren, die durch Aufruf der Funktion selbst von einem Startwert bis 0 herunterzählt und diese Werte auf der Konsole ausgibt.

### 2. Messreihe

Aus einer Messreihe wurden 100 ganzzahlige Werte in einer Liste gespeichert. Für die Messwerte sollen verschiedene statistische Kenndaten ermittelt werden:

- a. Berechnung des Minimums der Messwerte
- b. Berechnung des Maximums der Messwerte
- c. Berechnung des Medians der Messwerte

Hinweis: der Wert aus der Mitte der sortierten Liste

- d. Berechnung der Spannweite der Messwerte

Hinweis: der Abstand zwischen dem kleinsten und dem größten Wert

- e. Berechnung der mittleren Abweichung der Messwerte

Hinweis: für alle Werte ist der positive Abstand zum Mittelwert zu bilden  
diese Abstände werden aufsummiert und durch die Anzahl der  
Werte geteilt

Beispiel für 3 Werte:

Werte            3, 7, 2

Mittelwert      $(3+7+2) / 3 = 4$

mittlere Abw.    $(|3-4|+|7-4|+|2-4|) / 3 = (1+3+2) / 3 = 2$

- f. Berechnung des Wertes, der am häufigsten vorkommt

Hinweis zur Erzeugung der 100 Messwerte:

```
# Standardmodul random einbinden  
import random
```

```
messwerte = random.choices(range(0,1000), k=100)
```