# Lektion 11 – Funktionen mit Rückgabewert

## Nachschlagen bei…

W3Schools: Functions

## Anweisungen

Studiere das genannte Kapitel in W3Schools und die folgenden Anweisungen. Beantworte die Fragen in diesem Dokument und kopiere den Code der Aufgaben hierher.

## Erläuterungen

Oft ist es praktisch, wenn eine Funktion einen Wert zurückgibt, der dann in einer Variablen gespeichert werden kann. So etwas kennen wir schon.

x=input("x =") Hier gibt input() einen String zurück, der in x gespeichert wird.

print(str(z)+"! =", y) Hier verwandelt str() die Zahl z in einen String und gibt diesen zurück, sodass er mit "! =" zusammengehängt werden kann.

## Beispiele

### Beispiel 1 - Addition mit Rückgabewert

def addieren(a, b):

 a=float(a)

 b=float(b)

 return a+b

c=addieren(input("a = "), input("b = "))

print("summe =",c) *Probier aus, was dieser Code macht*

### Beispiel 2 - Größenvergleich

def maximum(a, b):

 if a>b:

 return a

 return b

print(maximum(3,5))

print(maximum(6,2))

print(maximum(7,7)) *Probier aus, was dieser Code macht*

Auffällig bei Beispiel 2 ist, dass das erwartete „else“ fehlt. Das ist deshalb nicht nötig, weil die Funktion beim ersten Auftreten von „return“ beendet wird.

## Fragen

Mit welchem Schlüsselwort wird in einer Funktion ein Rückgabewert erzeugt?

Welche Ausgaben erzeugen die vorherigen beiden Beispiele?

## Aufgaben

1. Erstelle eine Datei addieren\_liste\_func.py. Diese enthält eine Funktion addieren, der man eine beliebig lange Zahlenliste übergeben kann (etwa so: s=addieren(3,5,7,9)). Im Unterschied zu Lektion 10 wird die Summe als Rückgabewert ausgespuckt, sodass sie in der Variablen s aufgefangen werden kann, die dann mit print(s) ausgegeben wird.
2. Erstelle eine Datei fakultaet\_func.py. Definiere die Funktion fakultaet(z) so, dass innerhalb der Funktion die Fakultät von z berechnet und dann mit return ausgeworfen wird. Berechne sodann die Fakultät von 4 und 7 und gib beide aus. Etwa so:
y=fakultaet(4)
print("4! =", y)
print("7! =", fakultaet(7))
(Hier wird einmal der Funktionswert in die Variable y geschrieben, einmal direkt ausgegeben. Du kannst es jedoch machen, wie du möchtest.)

## Fleißaufgabe

1. Erstelle eine Datei vektoraddition\_ret.py. Diese enthält eine Funktion vadd(a,b) der man zwei dreidimensionale (Verbesserung: beliebigdimensionale) Vektoren (als Listen) übergeben kann. Die Summe wird jedoch diesmal als Rückgabewert ausgespuckt, der einer Variablen übergeben wird. Etwa so:
c=vadd(a,b)
print(a,"+",b,"=",c)
Der Codeschnipsel zeigt nur den Aufruf der Funktion und die Ausgabe. Die Definition der Funktion und die Belegung von a und b sind deine Aufgabe.