

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 26 09:32:31 2017

@author: christianehelzel
"""
```

```
| ' on Thu Oct 26 09:32:31 2017@author: christianehelzel'
```

1 Mehr zu Funktionen:

1.1 Beliebige Anzahl von Parametern

Durch '*' kann man eine beliebige Anzahl von Parametern an eine Funktion übergeben.

```
def arithmetic_mean(*args):
    su = 0
    num=0
    for x in args:
        su += x
        num +=1
    return su/num
```

Damit kann man die SELBE Funktion mit einer verschiedenen Anzahl an Uebergabeparametern aufrufen:

```
print ("Mittelwert: {0:5.2f}".format(arithmetic_mean(1,2,3)))
print ("Mittelwert: {0:5.2f}".format(arithmetic_mean(1,2,3,4,5)))
```

```
Mittelwert:  2.00
Mittelwert:  3.00
```

1.2 Keyword Argumente

Einigen Funktionsparametern wird ein 'default' Wert zugewiesen. Beim Aufruf der Funktion können wir diese Argumente weglassen, falls wir die Standardwerte verwenden möchten.

```
def trajectory(x,v0,y0,theta=45,g=9.81):
    import numpy as np

    theta = theta*np.pi/180
    return x*np.tan(theta) - g*x**2 / (2*v0**2 * np.cos(theta)**2) + y0
```

Bei diesem Aufruf ist theta=10 und g=4,56

```
print(trajectory(10, 20, 2, 10, 4.56))
```

| 3.17554782073

Hier werden die 'default' Werte für 'theta' und 'g' verwendet

```
print(trajectory(10, 20, 2))
```

| 9.5475

Man auch nur einen bestimmten Wert übergeben

```
print(trajectory(10, 20, 2, g=4.56))
```

| 10.86

1.3 Funktion als Argument einer Funktion

```
def diff1st(f, x, h=1E-3):  
    r = (f(x+h)-f(x-h))/(2*h)  
    return r  
  
def g(t):  
    return t**(-6)  
  
t = 1.2  
d1g = diff1st(g, t)  
print("Die Approximation der 1. Ableitung ist: %5.2f" % d1g)
```

| Die Approximation der 1. Ableitung ist: -1.67