

# Python\_34

September 15, 2018

In [5]: *#Erster Aufgabenteil*

```
#Pakete importieren
import numpy as np
from uncertainties import ufloat
from uncertainties import unumpy as unp

#Eingabe von Messwerten mit Fehlern
un=0.3
ut=0.05
n1=unp.uarray([0,10,20,30,40,50],[un])
T1=unp.uarray([0,0.1,0.2,0.2,0.1,0.2],[ut])
n2=unp.uarray([0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100],[un])
T2=unp.uarray([0,0,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.1,0.2,0.1,0,0.1,0,0.1,0.1,0.2,0],
c_Cu=380
m_Kal=0.098
C_Kal=m_Kal*c_Cu
C_T=5
m_w=ufloat(0.050,0.002)
m=5
g=9.81
d=ufloat(0.04765,0.0003)
T1_tot=sum(T1)
n1_tot=ufloat(50,un)
T2_tot=sum(T2)
n2_tot=ufloat(100,un)

#Berechnen von C
C1=m*g*n1_tot*np.pi*d/T1_tot
C2=m*g*n1_tot*np.pi*d/T2_tot

#Definition von Funktion zur Berechnung von c_w
def c_w(C_1,C_2,m_1,C_3):
    return (C_3-C_1-C_2)/m_1

#c_w Berechnen
c_w1=c_w(C_Kal,C_T,m_w,C1)
```

```

c_w2=c_w(C_Kal,C_T,m_w,C2)

#Ausgabe von Ergebnissen
print('c_w fuer erste Messung =',c_w1)
print('c_w fuer zweite Messung =',c_w2)
print('c_w Mittelwert =',(c_w1+c_w2)/2)

#Berechne gewichtete arithmetische Mittelwert
gew1=1/((c_w1.s)*(c_w1.s))
gew2=1/((c_w2.s)*(c_w2.s))
wrt1=c_w1.n
wrt2=c_w2.n
gam=(gew1*wrt1+gew2*wrt2)/(gew1+gew2)
gam_dev=1/(np.sqrt(gew1+gew2))
gam_res=ufloat(gam,gam_dev)

#Ausgabe von Ergebnissen bei gewichteten arithmetischen Mittelwert
print('c_w Gewichteter arithmetische Mittelwert =',gam_res)
print('Bestwert =',gam)
print('Standardunsicherheit =',gam_dev)

def tstd(x_0,y,u):
    return abs(x_0-y)/u

print('Abweichung des gewichteten Mittelwerts vom Literaturwert =',tstd(4187,4182,554),')

c_w fuer erste Messung = (8.3+/-1.4)e+03
c_w fuer zweite Messung = (3.5+/-0.6)e+03
c_w Mittelwert = (5.9+/-0.8)e+03
c_w Gewichteter arithmetische Mittelwert = (4.2+/-0.6)e+03
Bestwert = 4187.196493065609
Standardunsicherheit = 553.9562198129233
Abweichung des gewichteten Mittelwerts vom Literaturwert = 0.009025270758122744 sigma

```

In [11]: #Zweiter Aufgabenteil

```

#Pakete importieren
import numpy as np
from uncertainties import ufloat
from uncertainties import unumpy as unp

#Werte
U=ufloat(14.9,0.02)
I=ufloat(1.5,0.018)
c_Cu=380
m_Kal=0.098
C_Kal=m_Kal*c_Cu

```

```

dT1=4.5
dt1=180
dT2=19.45
dt2=900
m_w1=ufloat(116.94,0.003)
m_w2=ufloat(113.42,0.003)

#Definiere Funktion zur Berechnung der Waermekapazitaet des Wassers
def e_c_w(u,i,dt,dT,C_K,m_w):
    return ((u*i*dt/dT)-C_K)/m_w

#Berechnen von c_w
c_w1=e_c_w(U,I,dt1,dT1,C_Kal,m_w1)
c_w2=e_c_w(U,I,dt2,dT2,C_Kal,m_w2)

#Ausgabe von Ergebnissen
print('c_w bei erster Messreihe =',c_w1)
print('c_w bei zweiter Messreihe =',c_w2)
print('Mittelwert =',(c_w1+c_w2)/2)

#Berechne gewichtete arithmetische Mittelwert
gew1=1/((c_w1.s)*(c_w1.s))
gew2=1/((c_w2.s)*(c_w2.s))
wrt1=c_w1.n
wrt2=c_w2.n
gam=(gew1*wrt1+gew2*wrt2)/(gew1+gew2)
gam_dev=1/(np.sqrt(gew1+gew2))
gam_res=ufloat(gam,gam_dev)

#Ausgabe von Ergebnissen bei gewichteten arithmetischen Mittelwert
print('c_w Gewichteter arithmetische Mittelwert =',gam_res)
print('Bestwert =',gam)
print('Standardunsicherheit =',gam_dev)

def tstd(x_0,y,u):
    return abs(x_0-y)/u

print('Abweichung des gewichteten Mittelwerts vom Literaturwert =',tstd(7930,4182,0.07))

```

```

c_w bei erster Messreihe = 7.33+/-0.09
c_w bei zweiter Messreihe = 8.79+/-0.11
Mittelwert = 8.06+/-0.10
c_w Gewichteter arithmetische Mittelwert = 7.93+/-0.07
Bestwert = 7.930564484485937
Standardunsicherheit = 0.07073838595137996
Abweichung des gewichteten Mittelwerts vom Literaturwert = 53542.85714285714 sigma

```