

RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Steinhoff

**ATTEMPTED TASKS ARE Umdrehen.py
Rechnen.py**

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
import sys

dateiname = sys.argv[1]

with open (dateiname, "r") as dateiname:
    inhalt = dateiname.readlines()

print(inhalt[::-1])
```

THE MASTER SOLUTION IS

```
import sys

text = []
source = sys.argv[1]
try:
    with open(source,"r") as quelle:
        text = quelle.readlines()
    for i in range(len(text)-1,-1,-1):
```

```

        for c in range(len(text[i])-1,-1,-1):
            sys.stdout.write(text[i][c])
    print()
except:
    print("Fehler beim Laden der Datei")

```

THE INPUT DATA CONFIG FOR THIS FILE IS:

```

{
    "params": {
        "param1": "file-path"
    }
}

```

TEST RUN START HERE

chosen input

"'Umdrehen.py' *Das ist ein relativer Dateipfad, der auf eine Datei mit diesem Namen in dem aktuellen Arbeitsverzeichnis zeigt. Diese muss also in jedem Fall und unabhaengig von individuellem Betriebssystem, Ordnerstruktur etc. existieren, nicht-leer und lesbar sein (da es die selbe Datei ist die wir gerade ausfuehren."

your output

```

celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py
['print(inhalt[::-1])', '\n', '\tinhalt = dateiname.readlines()\n', 'with open (dateiname, "r") as dateiname:\n', '\n', 'dateiname = sys.argv[1]\n', '\n', 'import sys\n']
celinemueller@MacBook-Air-Buro sheet07 % 

```

expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py /Users/celinemueller/Desktop/codeTestEnv
ironment/Handins/Steinhoff/sheet07/Umdrehen.py
)]1-::[tlahni(tnirp

)(senildaer.emanietad = tlahni
:emanietad sa )"r" ,emanietad( nepo htiw

]1[vgra.sys = emanietad

sys tropmi
celinemueller@MacBook-Air-Buro sheet07 %
```

note that I gave the complete path to YOUR solution, to enable the comparing of the outputs.

COMMENTS ON YOUR SOLUTION:

Wie du sehen kannst, hast du alle Zeilen in ein Array gelesen, und das Array dann verkehrt herum ausgegeben. Dabei wurde `\n` nicht zu Zeilenumbrüchen, da die Zeilen nicht als Zeilen, sondern noch im Array ausgegeben wurden, weiterhin wurde die Reihenfolge der Buchstaben nicht umgedreht.

Wenn du `[::-1]` verwenden willst, musst du die Originaldatei als EINEN STRING einlesen (`.read()` anstatt `.readlines()`), ansonsten musst du über die Elemente des Arrays iterieren (diese sind dann Zeilen), und dann bei jeder Zeile nochmal über ihre Charaktere iterieren, um die umgedrehte Ausgabe zu ermöglichen.

STARTING TASK Rechnen.py

THE ATTEMPTED SOLUTION IS

```
import sys

def durchschnitt(zahlen):
    summe = 0
    for i in range(len(zahlen)):
        summe += zahlen[i]
```

```

    durchschnitt = summe // len(zahlen)
    return durchschnitt

def minimum(zahlen):
    minimum = zahlen[0]
    for i in range (len(zahlen)):
        if zahlen[i] < minimum:
            minimum = zahlen[i]
    return minimum

def maximum(zahlen):
    maximum= zahlen[0]
    for i in range (len(zahlen)):
        if zahlen[i] > maximum:
            maximum = zahlen[i]
    return maximum

zahlen = [float(arg) for arg in sys.argv[1::]]

durchschnitt = durchschnitt(zahlen)
print(durchschnitt)
minimum = minimum(zahlen)
print(minimum)
maximum = maximum(zahlen)
print(maximum)

```

THE MASTER SOLUTION IS

```

import sys

def avg(zahlen):
    return (sum(zahlen)/len(zahlen))

def min(zahlen):
    min = zahlen[0]
    for i in zahlen:

```

```

        if i < min: min = i
    return min

def max(zahlen):
    max = zahlen[0]
    for i in zahlen:
        if i > max: max = i
    return max

z = []
for zeile in sys.stdin:
    z += [float(zeile)]

print("Durchschnitt:", avg(z))
print("Minimum:", min(z))
print("Maximum:", max(z))

```

THE INPUT DATA CONFIG FOR THIS FILE IS:

```

{
  "stream": [
    {
      "type": "float",
      "repeatable": true,
      "min_repeats": 1,
      "max_repeats": 10
    }
  ]
}

```

TEST RUN START HERE

chosen input

```
[  
1,  
2.3,  
5.5,  
11  
]
```

your output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py  
Traceback (most recent call last):  
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Steinhoff/sheet07/Rechnen.py", line 26, in  
    <module>  
      durchschnitt = durchschnitt(zahlen)  
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Steinhoff/sheet07/Rechnen.py", line 7, in  
    durchschnitt  
      durchschnitt = summe // len(zahlen)  
ZeroDivisionError: integer division or modulo by zero  
celinemueller@MacBook-Air-Buro sheet07 %
```

expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py  
1  
2.3  
5.5  
11  
Durchschnitt: 4.95  
Minimum: 1.0  
Maximum: 11.0
```

!Wichtig! Beachte dass, wenn ich die Eingabe über die Standardeingabe manuell mache, ich ein geeignetes Symbol senden muss, um zu signalisieren, dass die Eingabe vorbei ist (das heißt EOF und ist meistens ctrl-d, recherchiere ansonsten für dein System), ansonsten gibt es einen

Fehler:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py
1
2.3
5.5
11

Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Mastersolutions/sheet07/Rechnen.py", line 20
, in <module>
    z += [float(zeile)]
           ~~~~~
ValueError: could not convert string to float: '\n'
celinemueller@MacBook-Air-Buro sheet07 %
```

COMMENTS ON YOUR SOLUTION:

Die Fehlermeldung entsteht, weil du versucht hast, die zu berechnen Zahlen aus den Programmparametern auszulesen, anstatt sie, wie in Aufgabenstellung gefordert, über den Eingabestrom einzulesen.

Wenn du die Zeile

```
zahlen = [float(arg) for arg in sys.argv[1::]]
```

durch die Zeile

```
zahlen = [float(arg) for arg in sys.stdin]
```

ersetzt, ergibt sich folgende Ausgabe:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py
1
2.3
5.5
11
4.0
1.0
11.0
celinemueller@MacBook-Air-Buro sheet07 %
```

Diese ist fast korrekt, die Unterschiede zur Musterlösungsausgabe ergeben sich daraus, dass du die Integer-Operation `//` anstatt die Float-Operation `/` verwendet hast.
