

RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Hemmer-Dudszus

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

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
import sys

f = open("datei.txt", "r")
print(f.readlines([-1]))
f.close()

sys.stdout.writelines(f)
```

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
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Umdrehen.py", line
3, in <module>
    f = open("datei.txt", "r")
FileNotFoundError: [Errno 2] No such file or directory: 'datei.txt'
celinemueller@MacBook-Air-Buro sheet07 %
```

expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py /Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Umdrehen.py

)f(seniletirw.tuodts.sys

)(esolc.f
))1-[(senildaer.f(tnirp
)"r" ,"txt.ietad"(nepo = f

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 dein Namen **einer** Quelldatei (datei.txt) in das Programm "hard-gecodet", anstatt, wie in der Aufgabenstellung gefordert, den Namen **irgendeiner, vorher nicht definierter** Quelldatei per **Programmparameter** zu übergeben.

Wenn du deine zweite Zeile durch die Zeile

```
f = open(sys.argv[1], "r")
```

ersetzt, ändert sich die Ausgabe wie folgt:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Umdrehen.py", line
  4, in <module>
    print(f.readlines([-1]))
              ~~~~~~^~~~~~
TypeError: argument should be integer or None, not 'list'
celinemueller@MacBook-Air-Buro sheet07 %
```

Die nächste Fehlermeldung ergibt sich daraus, dass du versucht hast, die Liste, die `.readlines()` zurück gibt, mit einem Index in eckigen Klammern ALS FUNKTIONSPARAMETER anzusprechen, was keine gültige Python-Syntax ist. Du hast also den Index innerhalb der Klammern geschrieben. Die Klammer steht für einen Funktionsaufruf, der wie gesagt eine Liste zurück gibt, diese steht also erst zu Verfügung wenn der Befehl fertig gelaufen ist (nachdem die Klammer geschlossen wurde).

Wenn du deine dritte Zeile durch die Zeile

```
print(f.readlines())
```

ersetzt, ändert sich die Ausgabe wie folgt:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py
['import sys\n', '\n', 'f = open(sys.argv[1], "r")\n', 'print(f.readlines(-1))\n', 'f.close()\n', '\n', 'sys.stdout.writelines(f)\n']
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Umdrehen.py", line 7, in <module>
    sys.stdout.writelines(f)
    ~~~~~^
ValueError: I/O operation on closed file.
celinemueller@MacBook-Air-Buro sheet07 %
```

Was heißt dieser fehler? Du beziehst dich in dem `print()` statement auf `f`, was du als Variable für die geöffnete Datei festgelegt hast. Diese hast du aber schon in Zeile 4 wieder geschlossen. Es empfiehlt sich, während die Datei noch geöffnet ist, ihren Inhalt in eine Variable zu speichern, die dann auch noch genutzt werden kann, wenn die Datei selbst wieder geschlossen wurde.

Wenn du deine dritte Zeile durch die Zeile

```
zeilen = f.readlines()
```

ersetzt, ändert sich die Ausgabe wie folgt:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py
import sys
f = open(sys.argv[1], "r")
zeilen = f.readlines(-1)
f.close()
sys.stdout.writelines(zeilen)
celinemueller@MacBook-Air-Buro sheet07 %
```

Wie du sehen kannst, wird jetzt deine Datei korrekt eingelesen und wieder ausgegeben, ohne aber etwas umzudrehen. Bitte vergleiche mit der Musterlösung um zu verstehen wie du die Aufgabe umsetzen kannst.

STARTING TASK Rechnen.py

THE ATTEMPTED SOLUTION IS

```
import sys

def durchschnitt(x):
    for element in x:
```

```

        summe += element
schnitt = summe / len(x)
print(schnitt)

def minimum(x):
    sort(x)
    mi = x[0]
    print(mi)

def maximum(x):
    sort(x)
    ma = x[len(x)-1]
    print(ma)

zahlen = sys.stdin.readlines

for zahlen in sys.stdin:
    array = []

durchs = durchschnitt(array)
mini = minimum(array)
maxi = maximum(array)

print("Durchschnitt:", durchs)
print("Minimum:", mini)
print("Maximum:", maxi)

```

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  
1  
2.3  
5.5  
11  
Traceback (most recent call last):  
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Rechnen.py",  
    line 25, in <module>  
      durchs = durchschnitt(array)  
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Hemmer-Dudszus/sheet07/Rechnen.py",  
    line 6, in durchschnitt  
      schnitt = summe / len(x)  
               ^^^^^  
UnboundLocalError: cannot access local variable 'summe' where it is not associated with a value  
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:

Eine Variable ist immer nur innerhalb (und unterhalb) des Scopes gültig, in dem sie initialisiert wurde. Du hast die Variable "summe" erst IN der for-schleife zum ersten mal benutzt (also initialisiert), d. h. , dass sie dir ausserhalb der For-schleife nicht zur verfügung steht.

Wenn du innerhalb der Funktion als ersten Befehl

```
def durchschnitt(x):
    summe = 0
    # ...
```

einsetzt, kannst du die Variable "summe" sowohl in der for-schleife als auch darunter anwenden.

Es entsteht dann aber eine weiter, andere Fehlermeldung. Diese kommt durch die Zeile

```
for zahlen in sys.stdin:
    array = []
```

zustande. Diese besagt, dass für jede Zeile (Eingabe) im Standard-Eingabestrom ein neues, leeres Array mit dem Namen "array" erzeugt werden kann. Dies liest natürlich nicht die Zeilen EIN , die übergeben werden. Bitte schaue dir nochmal an, welche Möglichkeiten es gibt, Zeilen

über den Eingabestrom einzulesen (und dabei auch in einer Variable zu speichern.) Schau dir vor allem `sys.stdin` und `.readlines()` nochmal an.
