

05e_Miniuebungen_Loesungen

0.1 Mini-Aufgaben zur Überprüfung des Verständnis: Fehlersuche

0.1.1 Exceptions

Schreiben Sie eine idiotensichere Einlesefunktion von Benutzereingaben, die solange `input()` aufruft, bis entweder ein `int` oder ein `float` eingegeben wurde, und geben Sie den eingelesenen Wert zurück.

Hinweis: Für den Test auf `int` hilft eine analoge Exception wie in den Beispielen dieses Abschnitts nicht. Erklären Sie warum das so ist. Lesen Sie sich dann schlau, was die Funktion `s.isnumeric()` für eine `str` Variable tut.

```
[4]: # Hilfsfunktion für float-Test
def is_float(n):
    try:
        float(n)
        return True
    except ValueError:
        return False

# Hilfsfunktion für int-Test
def is_int(n):
    # Wenn wir hier analog eine Konvertierung in int probieren würden,
    # würden wir für einen float alle Nachkommastellen verlieren!
    return n.isnumeric()

# Einlesefunktion
def input_int_float():
    while True:
        n = input("Geben Sie einen int oder float ein:")
        if is_int(n):
            return int(n)
        elif is_float(n):
            return float(n)
        else:
            print("Sorry, nochmal versuchen bitte")

val = input_int_float()
print("Finaler Wert:", val, type(val))
```

Geben Sie einen int oder float ein:3
Finaler Wert: 3 <class 'int'>

0.1.2 Exceptions bei Dateioperationen

Im Verlauf dieser Lerneinheit haben wir einige .csv Dateien angelegt. Versuchen Sie die folgenden Dateien zu öffnen, und verhindern Sie durch das Fangen einer geeigneten Exception den Programm-Abbruch, falls die Datei nicht gefunden wurde. Geben Sie jeweils aus, ob die Datei gefunden wurde oder nicht.

```
[8]: def probe_file(filename):  
    try:  
        f = open(filename, mode='r')  
        print("file", filename, "found")  
        f.close()  
    except FileNotFoundError:  
        print("file", filename, "not found")  
  
probe_file("DEXUSEU.csv")  
probe_file("blablaba.csv")  
probe_file("f_evaluation.csv")
```

```
file DEXUSEU.csv found  
file blablaba.csv not found  
file f_evaluation.csv found
```

0.1.3 Assertions

Programmieren Sie das vorherige Beispiel noch einmal, mit assertions. Mit der Funktion `os.path.isfile()` aus dem Modul `os` können Sie überprüfen, ob eine Datei existiert.

```
[10]: import os  
  
def probe_file(filename):  
    if os.path.isfile(filename):  
        print("file", filename, "found")  
    else:  
        print("file", filename, "not found")  
  
probe_file("DEXUSEU.csv")  
probe_file("blablaba.csv")  
probe_file("f_evaluation.csv")
```

```
file DEXUSEU.csv found  
file blablaba.csv not found  
file f_evaluation.csv found
```

0.1.4 Nutzung eines Debuggers

Turnen Sie das Beispiel aus dem Demovideo zum Debugging mit Spyder nach. Rekapitulieren Sie dann andere Szenarien, im Dunstkreis der bisherigen Übungen und Miniübungen, bei denen die Nutzung eines Debuggers hilfreich gewesen wäre.

0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Göddeke <https://www.ians.uni-stuttgart.de>,
Universität Stuttgart

Version vom April 2023

Lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz



Veröffentlicht auf <https://zoerr.de>, (alle Rechte am Logo vorbehalten)



Gefördert durch die Stiftung Innovation in der Hochschullehre. (alle Rechte am Logo vorbehalten)



Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie.

[]: