

Python

Sommerschule Spracherwerb

try - except - finally

- Error Handling

- Wie kann das Programm weiterlaufen, auch wenn Programmteile auf die wir keinen Einfluss haben einen Fehler produzieren?

-> Exceptions

<https://docs.python.org/3/tutorial/errors.html>

try:

■ **f = open("meinText.txt","r")**

except OSError as err:

■ **print("Fehler:",err)**

finally:

■ **print("Try-Except fertig.")**

```
try:  
    f = open("meinText.txt","r")  
except OSError as err:  
    print("Fehler:",err)  
finally:  
    print("Try-Except fertig.")
```

```
try:  
    f = open("meinText.txt","r")  
except OSError as err:  
    print("Fehler:",err)  
finally:  
    print("Try-Except fertig.")
```

```
try:  
    f = open("meinText.txt","r")  
except OSError as err:  
    print("Fehler:",err)  
else:  
    print("Keine Exception")  
finally:  
    print("Try-Except fertig.")
```

```
i = 1
while i <= 100:
    print(i)
    i = i + 1
```

```
i = 1
while True:
    print(i)
    i = i + 1
    if i >= 100:
        break
```


while

Solange eine Bedingung wahr ist, wird der Befehlsblock ausgeführt.

Kann mit **break** vorzeitig beendet werden.

import random

<https://docs.python.org/3/library/random.html>

```
from random import randint
```

<https://docs.python.org/3/library/random.html>

Exkurs: Module

Importieren von Programmbibliotheken

Built-in oder extern

beispiel:

```
import random
```

```
from random import randint
```

Übung 6: Nummernspiel

Erstelle ein Spiel, in dem:

zufällig eine Zahl zwischen 0 und 100 ausgesucht wird

Der Nutzer einen Wert zwischen 0 und 100 eingibt

Informationen über den Eingabewert werden preisgegeben

Ist (viel) größer/kleiner als der gesuchte Wert

Das Spiel läuft solange, bis die Zahl gefunden wurde oder der Nutzer “end” eingibt.

Wenn die Zahl gefunden wurde, soll die Anzahl der benötigten *Guesses* angezeigt werden.

Die *Guesses* sollen in einer Liste gespeichert werden.

```
for i in range(100):  
    print(i)
```

```
for i in range(100):  
    if i % 2 == 0:  
        continue  
    print(i)
```

Übung 7: Fibonacci-Folge

Erstelle eine Liste der Fibonacci-Folge mit einer for- und eine while-Schleife für die ersten 100 Zahlen der Folge.

Fibonacci-Folge: 0 1 1 2 3 5 8 13 ...

<https://de.wikipedia.org/wiki/Fibonacci-Folge>


```
for item in [1, 5, 42, -7]:  
    print(item % 2 == 0)
```

```
try:  
    f = open("datei.txt","r")  
    lines = f.readlines()  
    f.close()  
except:  
    print("Fehler")
```

<https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>

```
with open("datei.txt", "w") as f:  
    f.write("wichtiger Text")
```

<https://docs.python.org/3/tutorial/inputoutput.html#reading-and-writing-files>

Übung 8: Speichern der Daten

Verändere den Code vom Nummernspiel so, dass alle Guesses am Ende des Spiels in einer Datei gespeichert werden.

Jede Zeile soll dabei einen Spielablauf wiedergeben

Die Werte eines Spielablaufs sollen per “;” getrennt werden

Was musst du tun, damit ein Spielablauf am Ende einer Datei angehängt wird?

<https://docs.python.org/3/library/functions.html#open>

Übung 9: Sortieren

Lade dir aus Github (<https://github.com/Zadest/python-3>) den Ordner herunter.

Erstelle ein Skript, das:

- Alle Dateiname im Ordner "Data" als Liste in Python ausgibt.

- Sortiere diese Dateinamen anhand der Endungen in jeweils eine Liste

- Kopiere die Dateien anhand der Dateiendung in den entsprechenden Unterordner

Hinweis:

- Importiere die os - Bibliothek. Schaue dir besonders **os.listdir** und **os.rename** an

<https://docs.python.org/3/library/os.html>