

Exceptions mit Python

Lernen mit Fehlern umzugehen

Einführung

Bitte sehen Sie sich das folgende Programm an:

```
1 # 11-intro.py
2 # insert
3 print("Please input a number")
4 z = input()
5
6 # casting
7 num = int(z)
8
9 #output
10 print("You have entered the number", num, "correct?")
11 print("End")
```

Einführung

Solange der Benutzer im vorherigen Beispiel **eine Zahl eingibt**, ist alles in Ordnung. **Das Programm läuft bis zum Ende durch.**

Doch was passiert, wenn wir etwas eingeben, das nicht zu einer Zahl umgewandelt werden kann, z. B. "Hallo"

Einführung

Wir erhalten einen von vielen Fehler-Kategorien, in diesem Fall einen sogenannten **ValueError**.

Wir werden nun unser Programm mit **try & except** auf diesen Fehler vorbereiten.

Einführung



```
1 # 11-intro-improved
2
3 print("Please input a number")
4 z = input()
5
6 # casting
7 try:
8     num = int(z)
9     # output
10    print("You have entered the number", num, "correct?")
11
12 except:
13    print("You have not entered a number")
14
15 print("End")
```

Einführung



```
1 # 11-intro-improved
2
3 print("Please input a number")
4 z = input()
5
6 # casting
7 try:
8     num = int(z)
9     # output
10    print("You have entered the number", num, "correct?")
11
12 except:
13    print("You have not entered a number")
14
15 print("End")
```

try Anweisung

Die Anweisung **try** leitet eine **Ausnahmebehandlung** ein.

Ähnlich wie bei der if-Anweisung gibt es **verschiedene Zweige**, die das Programm durchlaufen kann.

Das Programm versucht, die Anweisungen durchzuführen, **die eingerückt hinter try stehen.**

try Anweisung

Falls die Eingabe **erfolgreich** ist, wird der **except-Zweig ignoriert**.

Ähnlich wie das Verhalten bei **if und else**.

except Anweisung

Ist die try-Anweisung jedoch **nicht erfolgreich**, handelt es sich um einen Fehler, nun wird dieser Fehler mit der except-Anweisung abgefangen.

Dies führt dazu, dass alles, was innerhalb der except-Zweigs steht, ausgeführt wird.

except Anweisung

Anschliessend läuft das Programm (**auch im Fehlerfall**) ohne Abbruch zu Ende. Dies, weil der Fehler zwar auftrat, aber **abgefangen** wurde.

Wichtig

Bitte vergessen Sie nicht den **Doppelpunkt** nach den verschiedenen Zweigen (gleich wie bei if-elif-else).



Eingabe wiederholen

Nun wollen wir uns noch ein Beispiel ansehen, bei dem **while** und die **Fehlerbehandlung** kombiniert werden.

Eingabe wiederholen



```
1 #11-retry-if-error.py
2
3 error = 1
4
5 # retry if wrong
6 while error == 1:
7     print("Please enter a number")
8     z = input()
9
10    # casting
11    try:
12        num = int(z)
13        print("Success! You have entered the number:", num, "correct")
14        error = 0
15
16    except:
17        print("You have not entered a number")
```

Ende

Das war alles für dieses Kapitel
