Python-Ausnahmen: Eine Einführung

Inhalt

- Ausnahmen versus Syntaxfehler
- Auslösen einer Ausnahme
- Die `AssertionError` Ausnahme
- Der `try...except` Block: Umgang mit Ausnahmen
- Die `else` Klausel
- Aufräumen nach der `finally` Verwendung
- Zusammenfassen

Ein Python-Programm wird beendet, sobald es auf einen Fehler stösst. In Python kann ein Fehler ein Syntaxfehler oder eine Ausnahme sein. In diesem Artikel erfahren Sie, was eine Ausnahme ist und wie sie sich von einem Syntaxfehler unterscheidet. Danach erfahren Sie, wie Sie Ausnahmen auslösen und Behauptungen aufstellen. Dann werden Sie mit einer Demonstration des `try...except` Blocks abschliessen.



Ausnahmen versus Syntaxfehler

Syntaxfehler treten auf, wenn der Parser eine falsche Anweisung erkennt. Beachten Sie folgendes Beispiel:

```
>>> print( 0 / 0 ))
  File "<stdin>", line 1
    print( 0 / 0 ))
                  ^
SyntaxError: invalid syntax
```

Der Pfeil zeigt an, wo der Parser auf den **Syntaxfehler gestossen ist** . In diesem Beispiel war eine Klammer zu viel. Entfernen Sie es und führen Sie Ihren Code erneut aus:

```
>>> print( 0 / 0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: integer division or modulo by zero
```

Dieses Mal ist ein **Ausnahmefehler aufgetreten** . Dieser Fehlertyp tritt immer dann auf, wenn syntaktisch korrekter Python-Code zu einem Fehler führt. Die letzte Zeile der Meldung gibt an, auf welche Art von Ausnahmefehler Sie gestossen sind.

Anstatt die Nachricht anzuzeigen `exception error`, gibt Python an, welche Art von Ausnahmefehler aufgetreten ist. In diesem Fall war es eine `ZeroDivisionError`. Python verfügt über verschiedene eingebaute Ausnahmen sowie die Möglichkeit, selbst definierte Ausnahmen zu erstellen.

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/archiv/m320/learningunits/lu04/theorie/lu4-kapitel_2/start

Last update: **2026/08/31 08:12**