

LU08e - Repetition Syntaxfehler und Logikfehler

Was ist ein Syntaxfehler?

Ein **Syntaxfehler** tritt auf, wenn der Code gegen die Regeln der Programmiersprache verstößt. Python versteht den Code in diesem Fall nicht und bricht die Ausführung ab.

Typische Beispiele für Syntaxfehler

- **Fehlende oder falsche Klammern:**

```
print('Hello World' # Fehlende schließende Klammer
```

- **Falsche Einrückung:**

```
def greet():
    print('Hello')
        print('World') # Einrückung stimmt nicht
```

- **Tippfehler im Code:**

```
priint('Hello') # Tippfehler im Funktionsnamen
```

Tipps zur Vermeidung

1. Nutze eine **IDE** (z. B. PyCharm), die Syntaxfehler direkt anzeigt.
2. Überprüfe deinen Code sorgfältig, bevor du ihn ausführst.
3. Lies die Fehlermeldungen genau durch. Sie geben Hinweise auf die Problemstelle.

Was ist ein Logikfehler?

Ein **Logikfehler** tritt auf, wenn der Code zwar korrekt ausgeführt wird, aber nicht das gewünschte Ergebnis liefert. Diese Fehler sind schwieriger zu finden, da sie keine Fehlermeldung erzeugen.

Beispiele für Logikfehler

- **Falsche Berechnung:**

```
def calculate_average(numbers):
```

```
return sum(numbers) # Vergessen, durch die Anzahl der Zahlen zu teilen
```

- **Endlosschleifen:**

```
x = 0
while x < 10:
    print(x) # 'x' wird nie erhöht, Schleife endet nie
```

- **Falsche Bedingungen:**

```
age = 20
if age > 18:
    print('You are a minor') # Logikfehler: Sollte 'You are an adult' sein
```

Debugging-Strategien für Logikfehler

1. Verwende einen **Debugger**, um den Programmablauf zu analysieren.
2. Prüfe die Werte von Variablen während der Programmausführung.
3. Nutze **Testfälle**, um die Logik deines Codes zu überprüfen.

Debugger nutzen

Ein **Debugger** ist ein Tool, das hilft, Fehler im Programm zu finden, indem es Einblicke in die Programmausführung gibt.

Wichtige Funktionen eines Debuggers

1. Breakpoints setzen:

Pausiert das Programm an einer bestimmten Stelle, damit du den aktuellen Zustand prüfen kannst.

1. Variablen prüfen:

Zeigt den aktuellen Wert von Variablen an.

1. Schrittweise Ausführung:

Führt den Code Zeile für Zeile aus, um den Ablauf zu verfolgen.

[m319-E2F](#), [m319-E2E](#)



Kevin Maurizi

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**



Permanent link:

<https://wiki.bzz.ch/de/modul/m319/learningunits/lu08/repetition>

Last update: **2025/06/23 07:45**