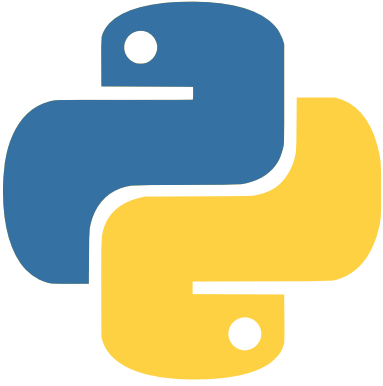
Universität Stuttgart
Projekt digit@L – BOOST. SKILLS. SUPPORT.



Programmierkurs Python

Fehlersuche und Fehlerbehandlung: Fehlertypen

Dominik Göddeke

IANS SimTech

Stiftung Innovation in der Hochschullehre

Fehlersuche und Fehlerbehandlung: Fehlertypen

Fehlersuche und Fehlerbehandlung

- **Fehler passieren** bei der Programmierung, das ist völlig normal

Fehlersuche und Fehlerbehandlung

- **Fehler passieren** bei der Programmierung, das ist völlig normal
- In diesem Abschnitt: verschiedene **Fehlertypen**

Fehlersuche und Fehlerbehandlung

- **Fehler passieren** bei der Programmierung, das ist völlig normal
- In diesem Abschnitt: verschiedene **Fehlertypen**
- Dann: **Exceptions** zur Fehlerbehandlung im Programm

Fehlersuche und Fehlerbehandlung

- **Fehler passieren** bei der Programmierung, das ist völlig normal
- In diesem Abschnitt: verschiedene **Fehlertypen**
- Dann: **Exceptions** zur Fehlerbehandlung im Programm
- Schließlich: Hilfe bei der **Fehlersuche, Debugging**

Fehlertypen

Fehlertypen

- **Syntaxfehler**: Interpreter bricht Ausführung des Programms ab

Fehlertypen

- **Syntaxfehler**: Interpreter bricht Ausführung des Programms ab
- Grund: **Code nicht lauffähig**

```
h=5; print(h
```

Fehlertypen

- **Syntaxfehler**: Interpreter bricht Ausführung des Programms ab
- Grund: **Code nicht lauffähig**

```
h=5; print(h
```

- Nur selbst reparierbar, denn geschuldet unserer Unfähigkeit oder Schludrigkeit beim Programmieren

Fehlertypen

- **Syntaxfehler**: Interpreter bricht Ausführung des Programms ab
- Grund: **Code nicht lauffähig**

```
h=5; print(h
```

- Nur selbst reparierbar, denn geschuldet unserer Unfähigkeit oder Schludrigkeit beim Programmieren
- Vernünftige IDEs „meckern“ über solche Fehler, bspw. durch Unterkringelung

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft
- **Beispiel im Code**: pq-Formel absichlich falsch

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft
- **Beispiel im Code**: pq-Formel absichlich falsch
- Boshafte Fehler, insb. bei **Weitergabe unserer Programme**

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft
- **Beispiel im Code**: pq-Formel absichlich falsch
- Boshafte Fehler, insb. bei **Weitergabe unserer Programme**
  - Versprechen an Benutzer*innen: Programm tut, was die Dokumentation verspricht

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft
- **Beispiel im Code**: pq-Formel absichlich falsch
- Boshafte Fehler, insb. bei **Weitergabe unserer Programme**
  - Versprechen an Benutzer*innen: Programm tut, was die Dokumentation verspricht
  - Benutzer*innen schauen selten in Quellcode

Fehlertypen

- **Logische Fehler**: Fehler in Berechnungen oder im Programmablauf
- Führen zu fehlerhaften Ausgaben, obwohl das Programm syntaktisch fehlerfrei durchläuft
- **Beispiel im Code**: pq-Formel absichlich falsch
- Boshafte Fehler, insb. bei **Weitergabe unserer Programme**
  - Versprechen an Benutzer*innen: Programm tut, was die Dokumentation verspricht
  - Benutzer*innen schauen selten in Quellcode
  - Oder haben Sie sich den Quellcode des Moduls `csv` angesehen?

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development
- Nicht: komplette Programmierung eines Codes, dann Testung

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development
- Nicht: komplette Programmierung eines Codes, dann Testung
- Sondern Tests für kleine Komponenten des Codes direkt bei der Entwicklung

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development
- Nicht: komplette Programmierung eines Codes, dann Testung
- Sondern Tests für kleine Komponenten des Codes direkt bei der Entwicklung
- Konzept: **unit-tests**

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development
- Nicht: komplette Programmierung eines Codes, dann Testung
- Sondern Tests für kleine Komponenten des Codes direkt bei der Entwicklung
- Konzept: **unit-tests**
- Bei jeder Änderung und Erweiterung: komplette Ausführung der so entstehenden Testsammlung

Fehlertypen

- **Vermeidung/Minimierung logischer Fehler**: test-driven development
- Nicht: komplette Programmierung eines Codes, dann Testung
- Sondern Tests für kleine Komponenten des Codes direkt bei der Entwicklung
- Konzept: **unit-tests**
- Bei jeder Änderung und Erweiterung: komplette Ausführung der so entstehenden Testsammlung
- So **Verifikation**, dass immer noch alles wie vorgesehen funktioniert

Fehlertypen

- **Laufzeitfehler**: Programmabbrüche, die zur Laufzeit auftreten

Fehlertypen

- **Laufzeitfehler**: Programmabbrüche, die zur Laufzeit auftreten
- Beispiel: komplexe Berechnung führt zu Wert `4` von `some_variable`

```
some_other_variable = 1/(4-some_variable)
```

Fehlertypen

- **Laufzeitfehler**: Programmabbrüche, die zur Laufzeit auftreten
- Beispiel: komplexe Berechnung führt zu Wert `4` von `some_variable`

```
some_other_variable = 1/(4-some_variable)
```

- Pyton meldet `ZeroDivisionError: division by zero`

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**
- Ausgesprochen nützlich: Programm-gesteuerte Reaktion auf Fehler

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**
- Ausgesprochen nützlich: Programm-gesteuerte Reaktion auf Fehler
- Ohne Unterbrechung des Programms

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**
- Ausgesprochen nützlich: Programm-gesteuerte Reaktion auf Fehler
- Ohne Unterbrechung des Programms
- Wichtig bei großen Programmen, kein Abbruch nach mehreren Stunden Rechenzeit

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**
- Ausgesprochen nützlich: Programm-gesteuerte Reaktion auf Fehler
- Ohne Unterbrechung des Programms
- Wichtig bei großen Programmen, kein Abbruch nach mehreren Stunden Rechenzeit
- Kennen wir schon: Kombination von `input()` mit `while` Schleife

Fehlertypen

- Laufzeitfehler **im Programm „abfangbar“**
- Ausgesprochen nützlich: Programm-gesteuerte Reaktion auf Fehler
- Ohne Unterbrechung des Programms
- Wichtig bei großen Programmen, kein Abbruch nach mehreren Stunden Rechenzeit
- Kennen wir schon: Kombination von `input()` mit `while` Schleife
- Mit Exceptions aber **viel** mächtiger

Codebeispiele

Impressum, Danksagung und Quellen

Stiftung Innovation in der Hochschullehre

Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, https://stiftung-hochschullehre.de

SimTech

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

---

Autor: Dominik Göddeke, IANS, Universität Stuttgart

ZOERR

Veröffentlicht auf dem Zentralen OER Repositorium Baden-Württemberg, https://www.zoerr.de