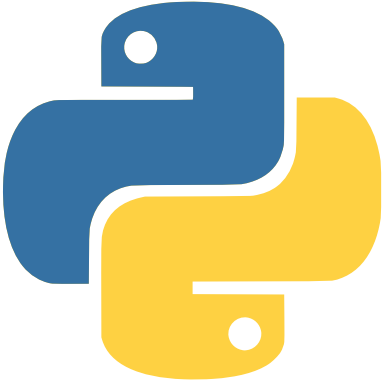
Universität Stuttgart
Projekt digit@L – BOOST. SKILLS. SUPPORT.



Programmierkurs Python

Fehlersuche und Fehlerbehandlung: Exceptions

Dominik Göddeke

IANS SimTech

Stiftung Innovation in der Hochschullehre

Fehlersuche und Fehlerbehandlung: Exceptions

Exceptions

- Laufzeitfehler heißen auch **Exceptions**

Exceptions

- Laufzeitfehler heißen auch **Exceptions**
- Gewissermaßen **Ausnahmen im regulären Programmablauf**

Exceptions

- Laufzeitfehler heißen auch **Exceptions**
- Gewissermaßen **Ausnahmen im regulären Programmablauf**
- In diesem Abschnitt: Verständnis ohne objektorientierte Programmierung

Exceptions

- Laufzeitfehler heißen auch **Exceptions**
- Gewissermaßen **Ausnahmen im regulären Programmablauf**
- In diesem Abschnitt: Verständnis ohne objektorientierte Programmierung
- Ausführliche Liste der **Exception-Hierarchie** unter `https://docs.python.org/3/library/exceptions.html` und im Jupyter Notebook

Exceptions

- Laufzeitfehler heißen auch **Exceptions**
- Gewissermaßen **Ausnahmen im regulären Programmablauf**
- In diesem Abschnitt: Verständnis ohne objektorientierte Programmierung
- Ausführliche Liste der **Exception-Hierarchie** unter `https://docs.python.org/3/library/exceptions.html` und im Jupyter Notebook
- **Ausblick nächste Videos**: effizient kombinierbar mit Debuggern

Abfangen von Exceptions

Abfangen von Exceptions

- **Sprachfeature**: `try/except`

Abfangen von Exceptions

- **Sprachfeature**: `try/except`
- Ausführung des Programms trotz Laufzeitfehler

Abfangen von Exceptions

- **Sprachfeature**: `try/except`
- Ausführung des Programms trotz Laufzeitfehler
- **Guter Programmierstil**: abfangen aller laut Dokumentation infragekommenden Exceptions

Abfangen von Exceptions

- **Sprachfeature**: `try/except`
- Ausführung des Programms trotz Laufzeitfehler
- **Guter Programmierstil**: abfangen aller laut Dokumentation infragekommenden Exceptions
- **Syntax** für einen `try/except` Block

```
try:
    ... Code der eventuell eine Exception "wirft" ...
except:
    ... Code der ausgeführt wird, wenn eine Exception aufgetreten ist ...

... Code der anschließend "normal" ausgeführt wird
```

Abfangen von Exceptions

- **Sprachfeature**: `try/except`
- Ausführung des Programms trotz Laufzeitfehler
- **Guter Programmierstil**: abfangen aller laut Dokumentation infragekommenden Exceptions
- **Syntax** für einen `try/except` Block

```
try:
    ... Code der eventuell eine Exception "wirft" ...
except:
    ... Code der ausgeführt wird, wenn eine Exception aufgetreten ist ...

... Code der anschließend "normal" ausgeführt wird
```

- `except` Block nur dann ausgeführt, wenn ein Fehler auftritt

Abfangen von Exceptions

- **Sprachfeature**: `try/except`
- Ausführung des Programms trotz Laufzeitfehler
- **Guter Programmierstil**: abfangen aller laut Dokumentation infragekommenden Exceptions
- **Syntax** für einen `try/except` Block

```
try:
    ... Code der eventuell eine Exception "wirft" ...
except:
    ... Code der ausgeführt wird, wenn eine Exception aufgetreten ist ...

... Code der anschließend "normal" ausgeführt wird
```

- `except` Block nur dann ausgeführt, wenn ein Fehler auftritt
- Fehler vor dem `try` Block: nicht behandelt, weiterhin Programmabbruch

Codebeispiel

Abfangen bestimmter Exceptions

Abfangen bestimmter Exceptions

- Obige einfachste Form des `try/except` Block: Abfangen **aller Fehler**

Abfangen bestimmter Exceptions

- Obige einfachste Form des `try/except` Block: Abfangen **aller Fehler**
- Unabhängig vom Typ des Fehlers

Abfangen bestimmter Exceptions

- Obige einfachste Form des `try/except` Block: Abfangen **aller Fehler**
- Unabhängig vom Typ des Fehlers
- Erweiterung `except` Block um **Typ des abzufangenden Fehlers**

Codebeispiele

Abfangen mehrerer Exception-Typen

Abfangen mehrerer Exception-Typen

- `except` Blöcke kaskadierbar zur **Reaktion auf verschiedene Fehler**

Abfangen mehrerer Exception-Typen

- `except` Blöcke kaskadierbar zur **Reaktion auf verschiedene Fehler**
- Erster zutreffender Block wird ausgeführt

Abfangen mehrerer Exception-Typen

- `except` Blöcke kaskadierbar zur **Reaktion auf verschiedene Fehler**
- Erster zutreffender Block wird ausgeführt
- **Syntax**

```
try:
    ... Fehleranfälliger Code
except ErrorClass1:
    ... Reaktion auf ErrorClass1 ...
    pass   # damit signalisieren wir, dass der Fehler behandelt wurde
except ErrorClass2:
    ... Reaktion auf ErrorClass2 ...
    pass
except (ErrorClass3, ErrorClass4):
    ... Reaktion auf ErrorClass3 und ErrorClass4 ...
    pass
except BaseException as error:
    ... alle anderen Fehler ...
    pass
```

Abfangen mehrerer Exception-Typen

- **Guter Programmierstil**: Reaktion auf alle erwarteten Exceptions separat

Abfangen mehrerer Exception-Typen

- **Guter Programmierstil**: Reaktion auf alle erwarteten Exceptions separat
- Aber: **Exceptions „teuer“**, also nicht übertreiben

Abfangen mehrerer Exception-Typen

- **Guter Programmierstil**: Reaktion auf alle erwarteten Exceptions separat
- Aber: **Exceptions „teuer“**, also nicht übertreiben
- **Kompromiss**: primär bei der Validierung von Eingaben

Abfangen mehrerer Exception-Typen

- **Guter Programmierstil**: Reaktion auf alle erwarteten Exceptions separat
- Aber: **Exceptions „teuer“**, also nicht übertreiben
- **Kompromiss**: primär bei der Validierung von Eingaben
- Dateioperationen sind auch Eingabe (!)

Codebeispiele

Schlussbemerkungen

Schlussbemerkungen

- Fehler können immer auftreten

Schlussbemerkungen

- Fehler können immer auftreten
- Robuste Programme beinhalten Fehlerbehandlungen

Schlussbemerkungen

- Fehler können immer auftreten
- Robuste Programme beinhalten Fehlerbehandlungen
- Basierend auf Antizipierung der Fehlerarten

Schlussbemerkungen

- Fehler können immer auftreten
- Robuste Programme beinhalten Fehlerbehandlungen
- Basierend auf Antizipierung der Fehlerarten
- Eigene Hierarchie an Fehlertypen definierbar $\rightarrow$ später

Schlussbemerkungen

- Fehler können immer auftreten
- Robuste Programme beinhalten Fehlerbehandlungen
- Basierend auf Antizipierung der Fehlerarten
- Eigene Hierarchie an Fehlertypen definierbar $\rightarrow$ später
- Blick in die Dokumentation der Standardbibliothek und externer Bibliotheken: zahlreiche eigene Fehlertypen

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