

# Fehler und Fehlermeldungen

---

Claas de Boer, Tilman Hinnerichs

4. Februar 2021

Python-Grundlagen

## Der Kurs in Kürze

---

- Datentypen: `int`, `str`, `float`, ...
- Kontrollstrukturen: `if`, `else`, `elif`
- Container: `list`, `set`, `tuple`
- Mapping: `dict`
- Objektorientierung: `class`, `init`
- Funktionen: `def f(x)`, `lambda f: ...`
- Built-Ins: `map`, `filter`, `zip`
- Dateien: `open`, `close`, `with open(...)` as `f`

## Heute: Fehler & Fehlermeldungen

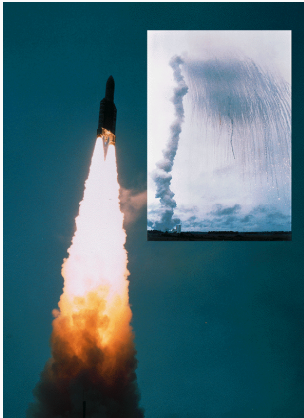
---

## Softwarefehler: Ariane 5

Fehler sind menschlich, warum sollte mein Programm keine haben?

# Softwarefehler: Arienne 5

Fehler sind menschlich, warum sollte mein Programm keine haben?



- Erster Flug der **Arienne 5**
- 4 Satelliten an Bord
- Nachfolgerakete der Arienne 4
- **RUD** nach 37 Sekunden
- 16-Bit signed Integer Overflow (g-Sensor)

Talk: Softwarefehler in der Raumfahrt

Wenn Python bei der Ausführung eures Programmes einen Fehler feststellt, wird ein **Stack Trace** ausgegeben und das Programm abgebrochen.

Wenn Python bei der Ausführung eures Programmes einen Fehler feststellt, wird ein **Stack Trace** ausgegeben und das Programm abgebrochen.

## Der **Stack Trace**

- enthält alle Funktionsaufrufe, die getätigt wurden bevor der Fehler auftrat,
- endet mit der eigentlichen Fehlermeldung



## Beispiel: NameError

```
[claas slides/session_10]€ python example.py
Traceback (most recent call last):
  File "example.py", line 5, in <module>
    say('Michael')
  File "example.py", line 3, in say
    print('Hello, ' + nam)
NameError: name 'nam' is not defined
```

```
1 # example.py
2 def say(name):
3     print('Hello, ' + nam)
4
5 say('Michael')
```

# Wie liest man einen Stack Trace?

```
[claas slides/session_10]€ python example.py
Traceback (most recent call last):
  File "example.py", line 5, in <module>
    say('Michael')
  File "example.py", line 3, in say
    print('Hello, ' + nam)
NameError: name 'nam' is not defined
```

- **Blau:** Der ausgelöste Fehler, letzte Zeile des Stack Traces
- **Grün:** Die Fehlernachricht (Error Message)
- **Orange:** Die Datei, die Zeile, und der Modulname bzw. die Funktion in welcher der Fehler aufgetreten ist
- **Rot:** Der ausgeführte Code, der den Fehler verursachte

Die beliebtesten Stack Traces:  
Nummer 3 wird dich überraschen!

---

## Beispiel: AttributeError

```
[claas slides/session_10]€ python example2.py
Traceback (most recent call last):
  File "example2.py", line 3, in <module>
    my_num.param
AttributeError: 'int' object has no attribute 'param'
```

```
1 # example2.py
2 my_num = 1
3 my_num.param
```

## Beispiel: ImportError

```
[claas slides/session_10]€ python example3.py
Traceback (most recent call last):
  File "example3.py", line 2, in <module>
    import xyz
ModuleNotFoundError: No module named 'xyz'
```

```
1 # example3.py
2 import xyz
```

## Beispiel: SyntaxError

```
[claas slides/session_10]€ python example4.py
File "example4.py", line 2
    def say(name)
        ^
SyntaxError: invalid syntax
```

```
1 # example4.py
2 def say(name)
3     print('Hello, ' + name)
4
5 say('Michael')
```

## Beispiel: KeyError

```
[claas slides/session_10]€ python example5.py
Traceback (most recent call last):
  File "example5.py", line 6, in <module>
    print(plz["Dresden Weißer Hirsch"])
KeyError: 'Dresden Weißer Hirsch'
```

```
1 # example5.py
2 plz = {
3     "Dresden Alberstadt": "01099",
4     "Dresden Plauen": "01187",
5     "Dresden Südvorstadt-Ost": "01217"
6 }
7 print(plz["Dresden Weißer Hirsch"])
```

## Beispiel: IndexError

```
[claas slides/session_10]€ python example6.py
Traceback (most recent call last):
  File "example6.py", line 7, in <module>
    print(tram_lines[10])
IndexError: list index out of range
```

```
1 # example6.py
2 tram_lines = [
3     "(7) Weixdorf - Betr. Hof Trachenberge",
4     "(12) A.-Wolf-Platz - Leutowitz",
5     "(46) Dresden-Niedersedlitz - Betr. Hof Trachenberge"
6 ]
7 print(tram_lines[10])
```



## Weitere Fehlermeldungen

- **TypeError**: Operation an ungeeignetem Typ (Bsp: `len(13)`)
- **ValueError**: Operation bekommt richtigen Typ, aber falschen Wert
- **KeyError**: Key nicht im Mapping (`dict`) vorhanden
- **IndexError**: Sequenzzugriff auf Element außerhalb des vorhandenen Index
- **IOError**: I/O Operation schlägt fehl (Bsp: Dateien)
- **ZeroDivisionError**: Division durch Null
- ...

## Fehler abfangen

---

# Try, Try, and Try Again

Mit dem `try/except`-Statement können Fehler behandelt werden.

```
1 # example2.py
2 my_num = 1
3 my_num.param
```

```
1 try:
2     my_num = 1
3     my_num.param
4 except AttributeError:
5     # falls im try-block ein AttributError auftritt,
6     # do this
7     print("Error: Property does not exist")
```

# Try, Try, and Try Again - Again

`try/except` kann als **Catch-All** verwendet werden.

```
1 try:
2     func()
3 except Exception as e:
4     print("An error occurred!")
5     print(e)
```

Oder um spezifische Fehler zu behandeln.

```
1 try:
2     func()
3 except ZeroDivisionError as zde:
4     # handle zde
5 except ValueError as ve:
6     # handle ve
7 except KeyError, IndexError as e:
8     # handle KeyError & IndexError
```

# Raise Exceptions from their Grave!

Vergleichbar mit **Built-Ins**, könnt ihr auf eine große Menge an Exceptions direkt zugreifen.

Eine Exception auslösen könnt ihr mit dem **raise**-Statement.

```
1 def series(x):  
2     if x < 0:  
3         raise ValueError  
4     return compute_series_for(x)
```

# Aufgaben

1. Implementiere eine Funktion `caesar_cipher`, die für einen String `plaintext` und eine Zahl `n` den um `n` caesar-verschlüsselten String `secret` berechnet.
2. Erweitere die Funktion `caesar_cipher` so, dass ein `TypeError` ausgelöst wird, wenn die Funktion mit einem Parameter des falschen Typs aufgerufen wird. Passe die Error Message an, sodass klar wird, welcher der Parameter den Fehler ausgelöst hat.
3. **Bonus:** Implementiere die Funktion `vigenere_cipher`, welche die Vigenère-Chiffre implementiert.

Vielen Dank, dass ihr so zahlreich erschienen  
seid!

Habt eine schöne vorlesungsfreie Zeit.