



Alexander Jenke, Theodor Straube

## Übung1: Einführung

Programmierkurs Python // Mittwoch, 30. Oktober 2019

### ToC

- Organisatorisches
- IDEs
- Hello World
  - Interpreter
  - print()
  - Funktionen
- Simple Calculator
  - Import
  - Main-Boilerplate
  - input()
  - int()





# Organisatorisches Tutoren

## **Alexander Jenke** alexander.jenke@tu-dresden.de

- Master Informatik, 3. Sem
- Seit 2015 an der TU
- Schwerpunkte:
  - Rechnernetze
  - Künstliche Intelligenz (Deep Learning)



#### **Theodor Straube**

theodor.straube@mailbox.tu-dresden.de

- Diplom Informatik, 9. Sem
- Seit 2015 an der TU
- Schwerpunkte:
  - Nicht festgelegt ;)

Hier bitte ein Bild einfügen

 Folien & Übungsaufgaben unter <a href="https://github.com/AlexanderJenke/python-lessons">https://github.com/AlexanderJenke/python-lessons</a>





## **Organisatorisches**

#### **Termine**

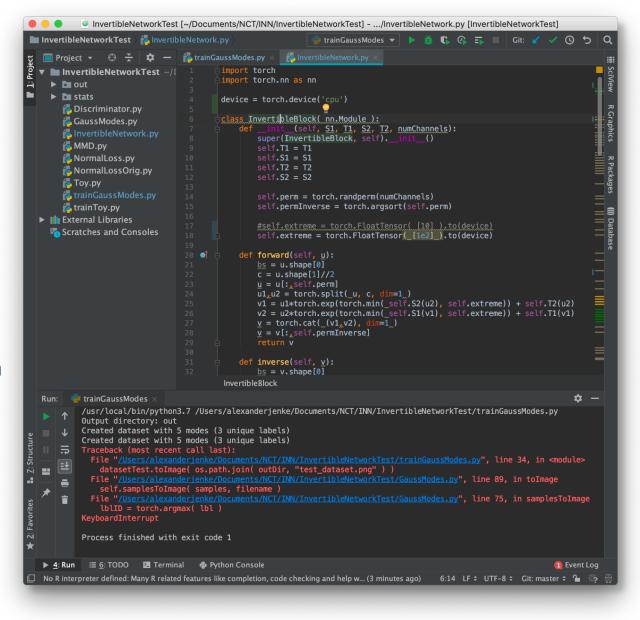
- Zeitraum: 28.10.2019 09.02.2020
- Weihnachtsferien: 22.12.2019 05.01.2020
- Buß- und Bettag: 20.11.2019 ⇒ Ersatz-Termin
- 2 Wochen pro Themenblock
  - Erste Woche Einführung und Aufgaben
  - Zweite Woche Fragen und Lösungen
- PyTorch-Einführung gegen Ende des Semesters (Deep Learning Framework)





## **IDEs - PyCharm**

- Integrierter Python Interpreter
- Debugger, Coverage Measurement, Profiler, uvm.
- Autovervollständigung
- Gratis Lizenz als Informatikstudent der TU Dresden <a href="https://www.jetbrains.com/shop/eform/students">https://www.jetbrains.com/shop/eform/students</a>







#### **Hello World**

 Der Python Interpreter wird direkt in der Konsole mittels python3 gestartet.

```
alexanderjenke@MacBook-Pro python3

Python 3.7.0 (default, Oct 2 2018, 09:18:58)

[Clang 10.0.0 (clang-1000.11.45.2)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

- Python-Skripte haben die Endung: .py
- Python-Skripte können mittels python3 script.py ausgeführt werden
- Python-Skripte können mittels python3 –i script.py interaktiv ausgeführt werden
- Ausgaben auf die Konsole erfolgen mit der print() –Funktion
- Eigene Funktionen definieren:

```
def name_der_funktion():
    do_some_stuff()
```





## **Simple Calculator**

- Funktionen einer Datei im selben Verzeichnis können mittels import skript aufrufbar gemacht werden
- Code innerhalb der main-boilerplate wird nur ausgeführt, wenn das Skript ausgeführt wird.

```
if __name__ == '__main__':
    do_some_stuff()
```

- Eingaben auf der Konsole können mitteln input() gelesen werden.
- Mit var = input() können sie in eine Variable namens var gespeichert werden.
- Strings können mit + zusammengefügt werden.
- Strings können mit int() zu Integern umgewandelt werden.





#### Fibonacci Reihe

 Beim Aufruf von Funktionen können Variablen übergeben werden

```
def foo(bar):
    print(bar)

foo("Hello World!")
```

 Das If-Statement führt abhängig von Bedingungen unterschiedlichen Code aus

```
if bar == "Hello World":
    print("Hallo Welt")

elif bar > 0:
    print(bar - 1)

else:
    foo(bar)
```

 While-Schleifen wiederholen Code, solange eine Bedingung erfüllt ist

```
i = 0
while i < 5:
    print(i)
    i += 1</pre>
```

For-Schleifen führen Code n-mal aus

```
for i in range(5):
    print(i)
```

• Logische Ausdrücke:

```
== gleich
```

!= ungleich

> größer

< kleiner





### nächste Woche

- PEP8
- Facts about Python
- String Operationen
- mehr zu Funktionen
- list comprehensions



