FH | JOANNEUM
University of Applied Sciences

Python in a Day
Programmieren leicht gemacht

Maierbrugger, Raab

FH JOANNEUM
University of Applied Sciences

DIGITAL INNOVATION HUB SÜD

Digital Innovation Hub Süd

- Der *Digital Innovation Hub Süd (DIH SÜD)* ist ein **Kompetenznetzwerk**, das **Klein- und Mittelbetriebe (KMU)** bei der **digitalen Transformation** mit Expertise, Vernetzung und Infrastruktur unterstützt.
- Services:
  - Durchführung von Informationsveranstaltungen
  - Aktivitäten der Innovations- und Technologieberatung
  - **Durchführung von Qualifizierungsmaßnahmen (Schulungen)**
  - Begleitung bei der Entwicklung von Innovationen

Regionen und Netzwerk

- Den KMUs steht dabei das regionale Forschungs- und Innovationssystem folgender Bundesländer zur Verfügung: Steiermark, Kärnten, Burgenland, Osttirol
- Das Netzwerk des DIH Süd umfasst **Hochschulen**, **Forschungszentren** und **Inkubatoren** aus den genannten Regionen.

Finanzierung

Themenbereiche von DIH Süd

- Produktions- und Fertigungstechnologien
- Digitale Sicherheit
- **Daten & Künstliche Intelligenz**
- Digitale Geschäftsmodelle & -prozesse
- Nachhaltigkeit & Kreislaufwirtschaft
- Arbeit der Zukunft & Humanressourcen

VORSTELLUNG

FH JOANNEUM
University of Applied Sciences

Mag.ª Marie-Christine Maierbrugger MSc

Education

- Lehramtstudium Mathematik und Englisch (KFU Graz)
- Masterstudium "Data Science and Artificial Intelligence" (FH JOANNEUM Graz)

Berufliche Aktivität

- Lehrerin an verschiedenen Schulen (Sekundarstufe)
- Hochschullektorin an der FH JOANNEUM

FH JOANNEUM
University of Applied Sciences

Mag. Raphaele Raab MSc

Ausbildung

- Lehramtstudium Mathematik
  (KFU Graz)
- Masterstudium
  "Data and Information Science"
  (FH JOANNEUM Graz)

Berufliche Aktivität

- Hochschullektor an der FH JOANNEUM

Fokus in Forschung und Lehre

Scripting, Machine Learning, Data Preprocessing, Mathematik

Kursplan

Vormittag: 09:00 – 12:30

Pause nach Bedarf

Inhalte:

- Setup
- Programmier Basics
- Pythonic code
- Funktionen lesbarer machen
- Type Checking
- Modules

Nachmittag: 13:30 – 17:00

Pause nach Bedarf

Inhalte:

- Datenmanagement
- Machine Learning
- Datenvisualisierung
- Python Utilities

SETUP

Voraussetzungen für Python Programmierung

- Python Installation - kompatibel mit den Packages die man verwenden will
- Virtuelle Umgebung – isolierte Umgebung für jedes Projekt
- Python Packages (Libraries)
- Dependency management – Abhängigkeiten der Packages untereinander verwalten
- Publishing – Code soll möglichst einfach weitergegeben werden können
- Eine Entwicklungsumgebung (IDE)

Visual Studio Code

- Entwicklungsumgebung (IDE)
- Für Windows, Linux, MacOS
- https://code.visualstudio.com/

VS Code installieren

Visual Studio Code

- Settings
  - Theme
  - Auto save
- Extensions installieren
  - Python
  - Jupyter
  - Ruff

UV – Python Package Manager

- Kann alle der genannten Voraussetzungen managen
- Einfach zu benutzen
- Schneller als andere Package Manager (pip, anaconda, poetry)
- Hat eine sehr gute Dokumentation:
  - https://docs.astral.sh/uv/

UV installieren

UV – Wichtige Commands

- uv --help
- uv python list
- uv python install 3.8 / uv python uninstall 3.8
- uv python pin <python-version>
- uv init … erstellt ein environment im momentanen Folder
- uv init <project_name> … neuer Folder project_name
- uv add <package> <package> <package>
- uv sync

UV – Konfigurations-Files

- Konfigurationsfiles beinhalten die dependencies eines environments
- Mithilfe dieser Files kann eine Umgebung verlässlich wiederhergestellt werden
  - `pyproject.toml`
    definiert die breiten requirements und enthält Python Projekt Metadaten
  - `uv.lock`
    Enthält die exakten, aufgelösten dependency versionen
- Diese Files (vor allem `uv.lock`) sollten nicht manuell verändert werden!

RUFF

- Ruff:
  - Code Formatter und Linter
  - Konfigurieren mittels (*ruff.toml* file) – oder default rules
  - Rules: https://docs.astral.sh/ruff/rules/
  - Ausführliches Beispiel für ruff.toml: https://github.com/CoreyMSchafer/dotfiles/blob/master/settings/ruff.toml
  - RUFF als Formatter und Linter aktivieren:
    - Strg + Shift + p
    - „User Settings JSON“

RUFF

Formatter und Linter testen

test_ruff.py > ...

```python
import os
import pandas as pd
import numpy as np


p = {1: 'this is a test for my formatter',2: 'it will surely manage to clean this mess',3: 'I would hatefor my code to be so ugly',
}
def function1(a, b):
    return np.sum(a + b)

def function(c):
    return pd.DataFrame(c)
g = 'another variable'
```

Jupyter - Notebooks

- Erlauben die Kombination von Markdown Text und ausführbarem Python Code
- Verwenden die virtual environment als kernel
- Man kann einzelne Zellen ausführen (oder mehrere auf einmal)