

Python 3 - Fortgeschrittene Konzepte und Anwendungen

Python-Umgebungen und Jupyter Notebooks

Hussam Alafandi

February 3, 2025

Einführung

Ziele des Tages

- Verstehen, warum virtuelle Umgebungen nützlich sind.
- Erstellung und Verwaltung von virtuellen Umgebungen.
- Installation und Verwaltung von Python-Paketen.
- Einführung in Jupyter Notebooks.
- Arbeiten mit Jupyter-Zellen und Markdown.

Überblick der Tools

- **Python:** Die Programmiersprache, die wir heute nutzen.
- **venv:** Erstellen isolierter Umgebungen für Projekte.
- **pip:** Paketmanager für Python-Bibliotheken.
- **Conda:** Alternative zu venv, speziell für wissenschaftliche Pakete.
- **Jupyter Notebook:** Interaktive Umgebung für Python-Code und Visualisierungen.

Warum virtuelle Umgebungen?

- Vermeidung von Versionskonflikten zwischen Projekten.
- Reproduzierbare Entwicklungsumgebung.
- Einfache Verwaltung von Abhängigkeiten.

Virtuelle Umgebungen erstellen

Erstellen einer virtuellen Umgebung mit venv

Schritte:

1. Erstellen der Umgebung:

```
1 python -m venv my_env
```

2. Aktivieren der Umgebung:

```
1 source my_env/bin/activate # Mac/Linux  
2 my_env\Scripts\activate # Windows
```

3. Deaktivieren der Umgebung:

```
1 deactivate
```

Erstellen einer virtuellen Umgebung mit Conda

Schritte:

1. Erstellen der Umgebung:

```
1 conda create --name my_env python=3.10
```

2. Aktivieren der Umgebung:

```
1 conda activate my_env
```

3. Deaktivieren der Umgebung:

```
1 conda deactivate
```

Verwalten von Abhängigkeiten

Installieren von Paketen mit pip

- Installation eines Pakets mit pip:

```
1 pip install notebook
```

- Anzeigen installierter Pakete:

```
1 pip list
```

- Erstellung einer Liste installierter Pakete (Abhängigkeiten):

```
1 pip freeze > requirements.txt
```

- Installation der gespeicherten Abhängigkeiten:

```
1 pip install -r requirements.txt
```

Installieren von Paketen mit Conda

- Installation eines Pakets mit Conda:

```
1 conda install numpy pandas matplotlib
```

- Anzeigen installierter Pakete:

```
1 conda list
```

- Erstellung einer Umgebungsdatei für Wiederverwendung:

```
1 conda env export > environment.yml
```

- Installation der gespeicherten Umgebung:

```
1 conda env create -f environment.yml
```

Einführung in Jupyter Notebooks

Was ist Jupyter Notebook?

- Eine interaktive Umgebung für Python-Code.
- Unterstützt Code, Markdown, Diagramme und mehr.
- Perfekt für Datenanalyse und wissenschaftliches Rechnen.

Starten von Jupyter Notebook

- Sicherstellen, dass Jupyter installiert ist:

```
1 pip install notebook
```

- Starten von Jupyter:

```
1 jupyter notebook
```

Jupyter Notebooks in VS Code und Google Colab

Jupyter Notebooks in VS Code

- Visual Studio Code unterstützt Jupyter Notebooks nativ.
- Installation der Jupyter-Erweiterung in VS Code
- Vorteil: Integrierte Debugging- und Entwicklungsumgebung.

Jupyter Notebooks in Google Colab

- Google Colab ist eine cloudbasierte Jupyter-Notebook-Umgebung.
- Kein Setup erforderlich – funktioniert direkt im Browser.
- Zugriff auf GPUs und TPUs für schnellere Berechnungen.
- Laden eines Notebooks in Colab:
 - Direkt eine ‘.ipynb’-Datei hochladen.
 - Öffnen aus Google Drive oder GitHub.
- Beispiel für das Ausführen eines Codes in Colab:

```
1 print("Hello Google Colab!")
```

Fragen?