4 Erstellung einer virtuellen Environment

Ein virtuelles Environment (VENV) dient der isolierten Verwaltung von Python-Abhängigkeiten für ein spezifisches Projekt. Dies ist besonders vorteilhaft, um Konflikte zwischen verschiedenen Projekten zu vermeiden, da jedes Environment eigene Abhängigkeiten und Pakete verwaltet, unabhängig von anderen Python-Installationen auf dem System.

Step 1: **Eingabeaufforderung (CMD) öffnen**

Drücken Sie Win + R, geben Sie cmd ein und bestätigen Sie mit Enter.

(HINWEIS: Mit **dir** können Sie sich immer alle Dateien im aktuellen Ordner anzeigen lassen)

Step 2: **Zum gewünschten Speicherort navigieren**

Falls das virtuelle Environment (VENV) in einem bestimmten Verzeichnis erstellt werden soll, navigieren Sie dorthin. Beispiel:

cd "C:\Users\<your User>\pythonEnv"

Falls das Verzeichnis noch nicht existiert, erstellen Sie es mit folgendem Befehl und wechseln anschließend in den Ordner:

1. **mkdir "C:\Users\<your User>\pythonEnv "**
2. **cd " C:\Users\<your User>\pythonEnv "**

Step 3: **Virtuelles Environment im aktuellen Verzeichnis erstellen**

Führen Sie den folgenden Befehl aus, um ein neues virtuelles Environment mit dem Namen myFirstEnv zu erstellen:

py -3.13 -m venv myFirstEnv

(HINWEIS: Falls eine andere oder mehrere Python-Versionen auf Ihrem System installiert sind, müssen Sie die jeweilige Version entsprechend anpassen, für die Sie das virtuelle Environment erstellen möchten. Hier am Beispiel Python 3.13)

Nach der Ausführung wird ein neuer Ordner „myFirstEnv“ im aktuellen Verzeichnis erstellt. Dieser Ordner enthält alle notwendigen Dateien für die virtuelle Environment, einschließlich der Interpreter-Binaries, Bibliotheken und Konfigurationsdateien.

Step 4: **In den Environment-Ordner navigieren und aktivieren**

Wechseln Sie in den Scripts-Ordner des erstellten virtuellen Environments:

cd "C:\Users\<your User>\pythonEnv\myFirstEnv\Scripts"

Führen Sie anschließend den folgenden Befehl aus, um das virtuelle Environment zu aktivieren:

activate

Nach der Aktivierung sehen Sie im Befehlszeilen-Prompt den Namen des Environments in Klammern, z. B.:

(myFirstEnv) C:\Users\<your User>\pythonEnv\myFirstEnv\Scripts>

Dies zeigt an, dass das virtuelle Environment erfolgreich aktiviert wurde.

Step 5: **Pakete im aktivierten Environment installieren**

Nach der erfolgreichen Aktivierung des virtuellen Environments erscheint (myFirstEnv) vor der Eingabezeile, was darauf hinweist, dass das Environment aktiv ist. Innerhalb dieses Environments können nun alle für das Projekt notwendigen Pakete installiert werden.
Beispielhafte Installationen:

pip install numpy

pip install pandas

...

Um eine Übersicht der bereits installierten Pakete und deren Versionen zu erhalten, kann folgender Befehl verwendet werden:

pip list

(Falls Sie eine Requirements.txt Datei zur Verfügung gestellt bekommen, können Sie dieses Installieren wie es unter Moodle „Environments In- und Export erklärt“ wird.

Step 6: **Spyder-Kernels für die Nutzung mit Spyder installieren**

Falls Spyder als Entwicklungsumgebung genutzt werden soll, ist es notwendig, die Spyder-Kernels zu installieren. Dies kann mit folgendem Befehl erfolgen:

pip install spyder-kernels==3.0.*