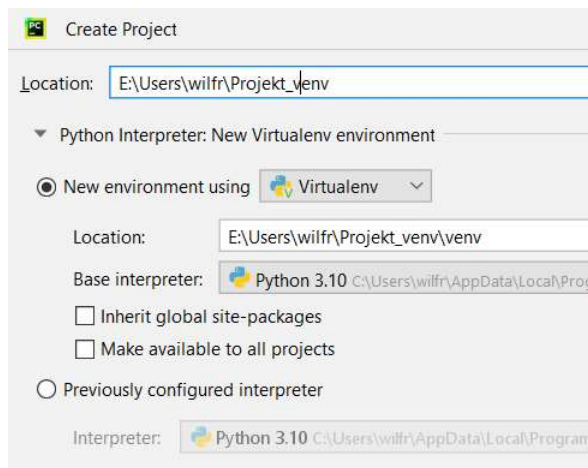


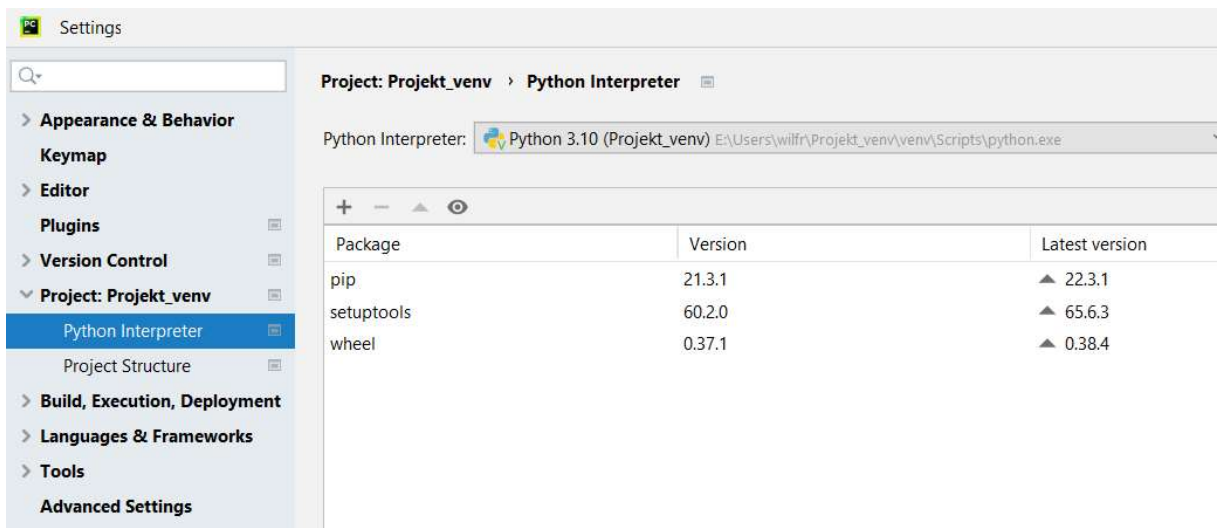
6. Virtuelle Entwicklungsumgebung mit dem venv-Modul

- Möglichkeiten einer virtuellen Entwicklungsumgebung
 - Sie isoliert verschiedene Python-Versionen und deren Paket-Installationen voneinander
 - einmal für ein Python-Projekt eingerichtet, bedeutet dies, dass die verwendeten Pakete nur diesem Projekt zur Verfügung stehen
 - die Aktualisierung eines dieser Pakete beeinflusst nicht andere - auf dem System installierte – Pakete
 - unterschiedliche Projekte können unterschiedliche Python-Versionen verwenden
 - Pakete nicht global zu installieren hat den Vorteil, Paket-Konflikte zu vermeiden, die bei einer systemweiten Installation auftreten könnten
 - Eine virtuelle Umgebung kann auch mit dem Module venv erzeugt werden
- PyCharm hat venv bereits integriert
 - beim Anlegen eines neuen Projektes kann im Konfigurationsfenster die virtuelle Entwicklungsumgebung aktiviert werden

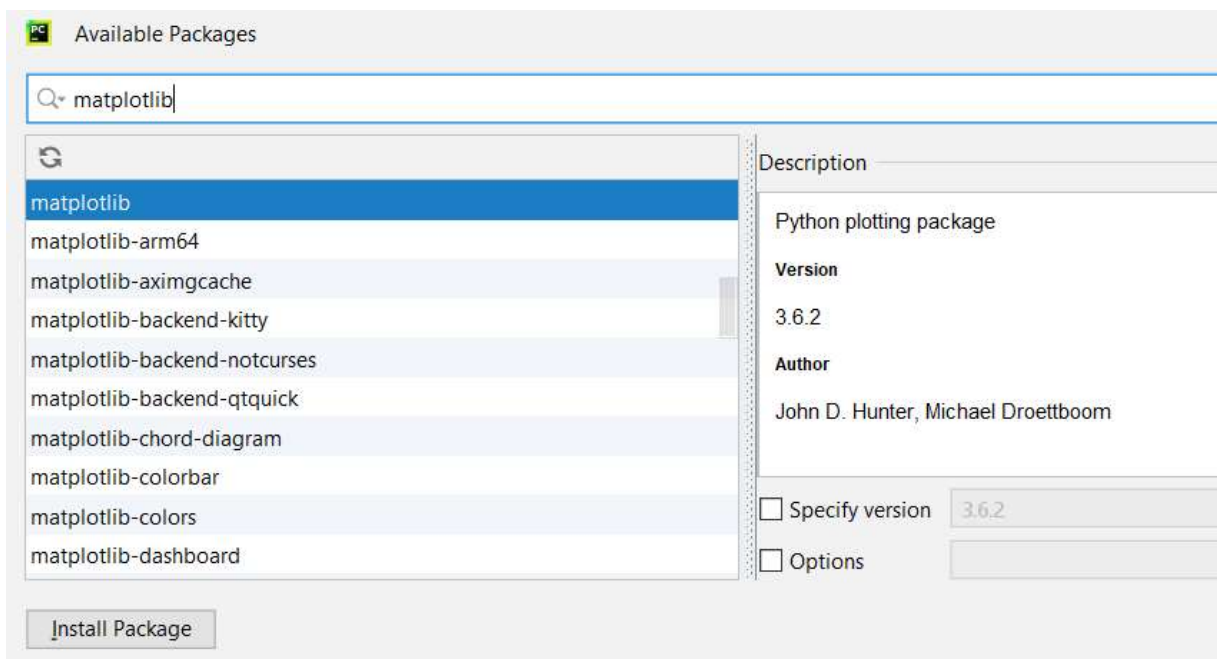


6.1. Paket in der virtuellen Entwicklungsumgebung installieren

- in der virtuellen Entwicklungsumgebung wird das Paket matplotlib installiert werden
- matplotlib ist eine Bibliothek zum Erzeugen von Diagrammen und Darstellungen in verschiedenen Formaten
- man kann mit nur wenigen Codezeilen Plots, Histogramme, Leistungsspektren, Balkendiagramme, Fehlerdiagramme, Streudiagramme (Punktwolken), und so weiter erzeugen
- im Konfigurationsfenster des Projektes (Settings) den Python-Interpreter „venv“ aktivieren



- das Paket (Package) matplotlib installieren
 - +Zeichen betätigen
 - Name des gewünschten Pakets eingeben – **matplotlib**
 - mit **Install Package** dann das Paket installieren



- Nach der Installation steht das Paket matplotlib nur in der virtuellen Umgebung des neuen Projektes zur Verfügung – jedoch wurde die Standard-Umgebung nicht verändert

- in dem Projekt mit der virtuellen Entwicklungsumgebung ist eine Python-Datei zu erstellen – mit den folgenden Anweisungen wird ein Diagramm angezeigt:

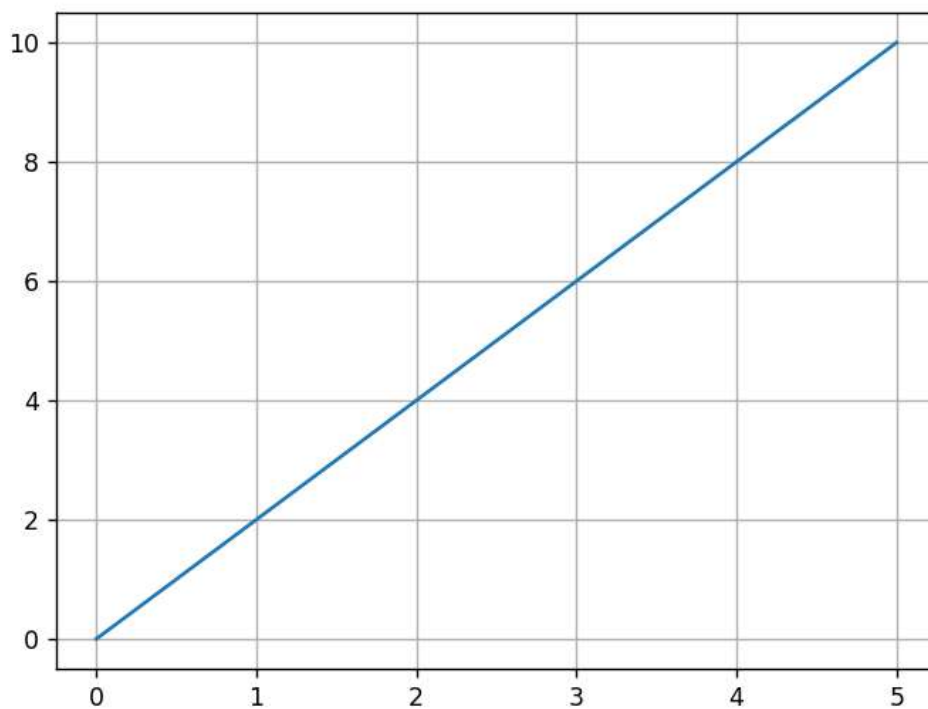
```
# Einbinden der Bibliothek matplotlib zum Plotten bzw. Erzeugen von
Diagrammen
# der Name ist sehr lang, daher empfiehlt es sich, eine Abkürzung zu
definieren, z.B. plt
import matplotlib.pyplot as plt

# plot-Funktion erhält eine Liste mit y-Werten
# die Indizes dieser Liste werden automatisch als x-Werte genommen
plt.plot([0, 2, 4, 6, 8, 10])

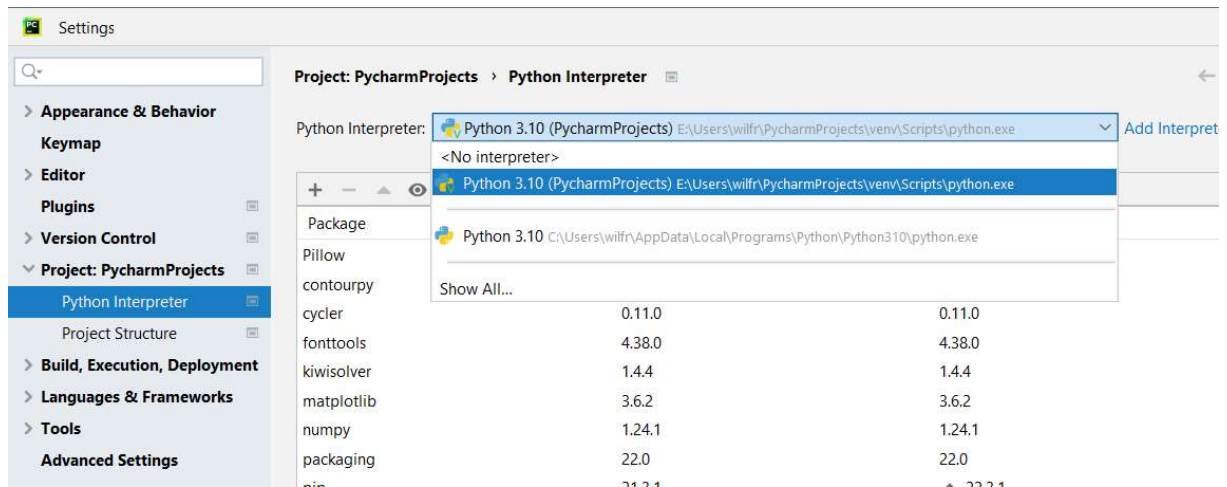
# ein gepunktetes Diagramm-Gitter einblenden
plt.grid(True)

# Diagramm anzeigen
plt.show()
```

- mit der Programmausführung wird das Diagramm angezeigt:



- venv-Umgebung deaktivieren:
 - Standard-Python-Interpreter auswählen und aktivieren:



C:\Users\wilfr\AppData\Local\Programs\Python\Python310\python.exe

E:/Users/wilfr/PycharmProjects/TestProjekt/test.py

Traceback (most recent call last):

File "E:\Users\wilfr\PycharmProjects\TestProjekt\test.py", line 3, in <module>

```
import matplotlib.pyplot as plt
```

ModuleNotFoundError: No module named 'matplotlib'

- Wird das Programm in der Standard-Umgebung oder bei deaktivierter venv-Umgebung ausgeführt, erhält man diesen Fehlerhinweis:

```
ModuleNotFoundError: No module named 'matplotlib'
```

6.2. Virtuelle Umgebungen bieten eine Reihe von Vorteilen:

- ohne Administrationsrechte Python-Module auf dem Rechner installieren
- die benötigten Module für die einzelnen Projekte sind voneinander getrennt
- die jeweiligen Projekte sind unabhängig von einem bestimmten Rechner und lassen sich problemlos auch auf anderen Rechnern wiederherstellen.
- das gleiche Modul kann in unterschiedlichen Versionen in verschiedenen Projekten verwendet werden

Aufgaben

1. Virtuelle Entwicklungsumgebung

es ist ein Projekt mit einer virtuellen Entwicklungsumgebung anzulegen.

In diesem Projekt soll das Paket matplotlib installiert werden. Mit einem Testprogramm (analog dieses Skript Seite 3) ist ein Diagramm zu erzeugen.

Anschließend soll die virtuelle Entwicklungsumgebung wieder deaktiviert werden – die Programmausführung zeigt dann nur noch

```
ModuleNotFoundError: No module named 'matplotlib'
```

an