PIP

Der Python Package Installer

Was ist PIP?

PIP ist der Python Package Installer und wird verwendet, um Python-Pakete zu installieren, zu verwalten und zu entfernen. Es ist das Standard-Werkzeug zur Verwaltung von Python-Bibliotheken und ein Muss für Python-Entwickler. Er wird von der Python-Foundation verwaltet.

Prüfen, ob PIP installiert ist

pip --version

Das verät uns neben der Versionsnummer auch den Ort, wo pip liegt:

pip 24.1.2 from C:\Users\admin\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)

Installieren von Paketen

Folgende Befehle können genutzt werden, um Pakete in die jeweils aktivierte Umgebung zu installieren. Falls keine virtuelle Umgebung genutzt wird, wird in die Python-Installation installiert, die zuerst im Pfad gefunden wird.

pip install package-name

Falls eine spezifische Version gewünscht ist, kann diese mit angegeben werden

pip install package-name==1.23.4

mit dem verbose-Flag bekommen wir mehr Ausgaben

pip install --verbose numpy

Extras mitinstallieren

Einige Pakete bieten optionale Features (Extras) an, die mit installiert werden können:

pip install paketname[extra1,extra2]

z.B.

pip install requests[security]

Versionsgrenzen setzen

Wenn du sicherstellen möchtest, dass keine inkompatiblen Versionen installiert werden:

pip install "paketname<2.0.0"

Pakete aus einer Requirements-Datei installieren

Pakete können aus einer Requirements.txt Datei installiert werden

pip install -r requirements.txt

Um Pakete in eine requirements.txt zu speichern, können wir den freeze-Befehlt und den Umleitungsoperator der Shell nutzen.

pip freeze > requirements.txt

Pakete updaten

Um ein Paket in der Umgebung upzudaten, kann der upgrade Befehl genutzt werden

pip install --upgrade numpy

Dabei wird die die neuste Version des Pakets installiert wenn möglich und die alte Version gelöscht. Eine alternative Schreibweise dafür ist:

pip install -U numpy

Pakete aus einer Requirements.txt updaten

Um alle Packages in einer requirements.txt upzudaten, können wir upgrade auch hier ausführen

pip install --upgrade -r requirements.txt

Paketinformationen

Um sich einen Überblick über alle installierten Pakete zu verschaffen, bietet sich der list-Befehl an

pip list

Dabei werden die Pakete inklusiver ihrer Versionsnummer angezeigt

rustworkx *0.15.1*
scikit-learn *1.4.1.post1*
scipy *1.9.3*

Informationen über ein Paket mit show

Genauere Informationen über ein spezifisches, installiertes Paket liefert der show Befehl:

pip show pandas

Dabei werden unter anderem auch die Abhängigkeiten des Pakets gezeigt, sowie andere Pakete, die Pandas in dieser Umgebung benötigen.

Version: 1.5.1

Summary: Powerful data structures for data analysis, time series, and statistics

Home-page: https://pandas.pydata.org

Author: The Pandas Development Team

Author-email: pandas-dev@python.org

License: BSD-3-Clause

Location: C:\Users\spielprinzip\AppData\Local\Programs\Python\Python311\Lib\site

-packages

Requires: numpy, python-dateutil, pyt

Required-by:

Deinstallation von Paketen mit uninstall

Um ein Paket aus der aktuellen Umgebung zu entfernen:

pip uninstall package

Falls eine Requirements-Datei genutzt wird, können alle Pakete auf einen Schlag deinstalliert werden.

pip uninstall -r requirements.txt -y

Das -y-Flag gibt an, dass die Pakete ohne Nachfrage gelöscht werden.

Abhängigkeitsgraph

Mit dem Paket pipdeptree lassen sich die Abhängkeiten visualisieren

python -m pip install pipdeptree

danach ausführen

```
pipdeptree
```