

```
setx PYENV %USERPROFILE%\pyenv
setx PYENV_ROOT %USERPROFILE%\pyenv
setx PYENV_HOME %USERPROFILE%\pyenv
```

pyenv – Installation

- unter Windows

```
git clone https://github.com/pyenv-win/pyenv-win.git $HOME\pyenv

setx PATH "%USERPROFILE%\pyenv\pyenv-win\bin;%USERPROFILE%\pyenv\pyenv-win\shims;%PATH%"
```

auch manuell
möglich, wer das
bevorzugt

- PC neustarten!
- verifizieren

```
pyenv --version
```

pyenv – wichtige Befehle

- Python-Version installieren/deinstallieren

```
pyenv install 3.9.6  
  
pyenv uninstall 3.9.6
```

- verfügbare Versionen anzeigen lassen

```
pyenv install --list
```

- globale Python-Version setzen


```
pyenv global 3.9.6
```


pyenv – wichtige Befehle

- lokale Python-Version für ein Verzeichnis

```
mkdir mein-projekt  
cd mein-project  
pyenv local 3.9.6
```

versuch es selbst:
pip install numpy



- temporär eine bestimmte Version in der Shell nutzen

```
pyenv shell 3.9.6
```

- interne Python-Tests laufen lassen (unabhängig von pyenv)

```
python -m test
```

virtuelle Umgebungen

- zum Erstellen von *virtual environments* in Python gibt es mehrere Wege ... wer ist noch verwirrt?
 - pyenv → Manager für mehrere Python-Versionen
 - virtualenv/venv → Manager für virtuelle Umgebungen für eine bestimmte Python-Version
- ```
python -m venv <your_virtual_env_name>
```
- pyenv-virtualenv → Manager für virtuelle Umgebungen über mehrere Python-Versionen hinweg



```
setx PATH "%USERPROFILE%\.pyenv\pyenv-win\bin;%USERPROFILE%\.pyenv\pyenv-win\shims;%USERPROFILE%\.pyenv\pyenv-win\plugins\pyenv-virtualenv\bin;%PATH%"
```

# pyenv-virtualenv

- Installation (Windows)

```
git clone https://github.com/pyenv/pyenv-virtualenv.git
~/.pyenv/plugins/pyenv-virtualenv
```

meistens muss man  
den PC nochmal  
neustarten hiernach

- virtuelle Umgebung erstellen

```
pyenv virtualenv 3.9.6 my-venv
```

- virtuelle Umgebung aktivieren

```
pyenv activate my-venv
```