# Programmieren I (Python)

Christian Osendorfer

2023-12-07

# Packages

# `pip` the package manager

- The package manager `pip` (*Pip Installs Packages*?) is part of the standard library.
- `pip install package_name`
- Packages available for installation are listed on Python Package Index.
  - `pip` can also use local pypi repository: `pip install pck --extra-index-url=http://my.org.pypi/`.
- Configure `pip` through `~/.pip/pip.conf`
  - E.g. have an entry `--extra-index-url` under `[global]`.
- Projects usually collect their required packages (and their version) into a text file `requirements.txt`.
  - Use `pip install -r requirements.txt` to establish the correct settings.

# requirements.txt

- The file `requirements.txt` looks like so:

```
numpy==1.20.2
scipy==1.4.1
```

- For a given project, create a `requirements.txt` file using
  - `pip freeze > requirements.txt`
  - This file usually sits at the top level of your project folder.

# Virtual Environments

- Different projects depend on different packages with different versions.
  - Some packages are incompatible with each other.
  - Some versions of the same package are incompatible with each other.
- Need ways to organize dependencies and isolate conflicting packages from each other.
- Solution for python: **virtual environments**.

# Anaconda

# (Ana)conda virtual environments

- Anaconda (or miniforge) comes with one main binary: `conda`.
- `conda create --name my_env_name` or `conda create -n my_env_name`.
  - This provides a minimal python installation.
- If you have a special version requirement regarding python:
  - `conda create -n my_env_name python=3.11`
- Get some information about available environments: `conda info --envs`.
- Using a created environment needs *activation*:
  - `conda activate my_env_name`
  - `conda deactivate`
- Remove an existing environment:
  - `conda env remove --name my_env_name`

https://haw-landshut.de/ki