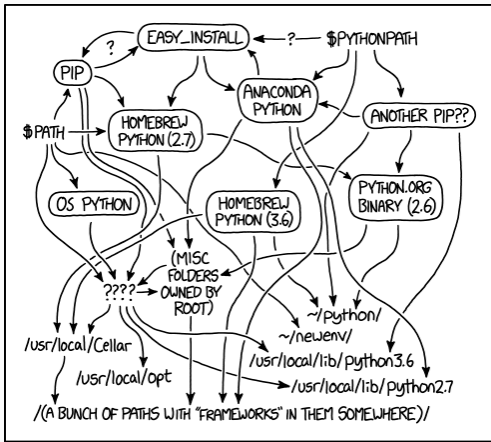


How to use poetry

November 28, 2023

Engineers for Exploration, UC San Diego

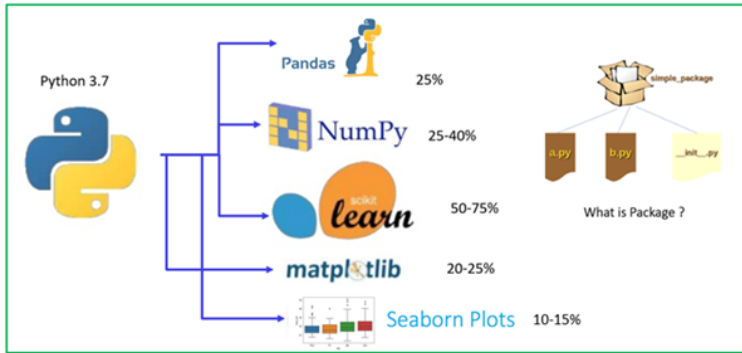
Obligatory XKCD



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Fundamentals: What are Python Packages?

- Collection of redistributable Python code
- Package metadata: PEP 518, PEP 621



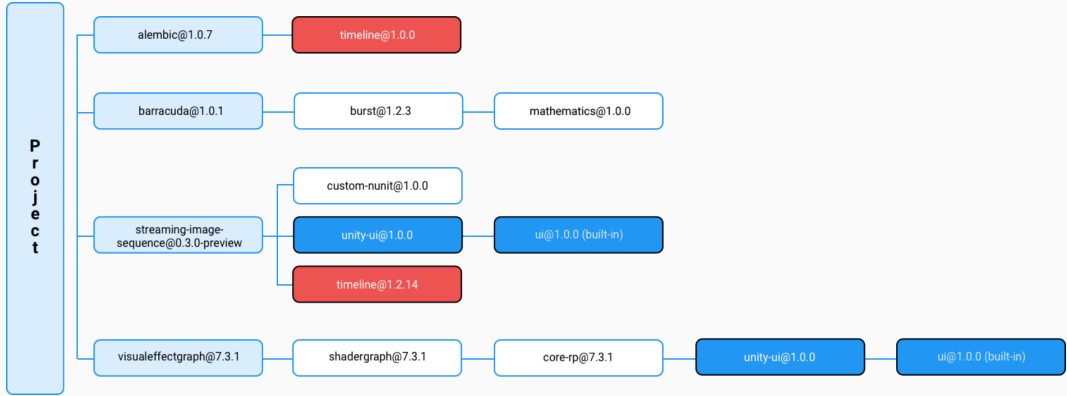
2

²<https://www.analyticsvidhya.com/blog/2021/02/key-python-packages-for-data-science/>

Fundamentals: What are Package Managers?

- Package Installation
- Package Dependency Resolution
- Examples:
 - conda
 - pip
 - pipenv
 - poetry

Package Dependency Management

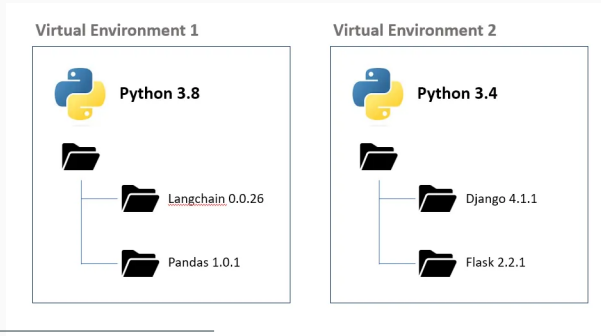


3

³<https://docs.unity3d.com/Manual/upm-conflicts.html>

Fundamentals: What are Virtual Environments?

- Isolated execution and configuration contexts
- Isolated packages



4

⁴[https:](https://medium.com/@TejasEkawade/installing-python-virtual-environments-cb865c3147a7)

[//medium.com/@TejasEkawade/installing-python-virtual-environments-cb865c3147a7](https://medium.com/@TejasEkawade/installing-python-virtual-environments-cb865c3147a7)

What else do we want in packages?

- Authors
- Contact info
- License info
- Version info
- Entry points
- Dependencies
- Development tools

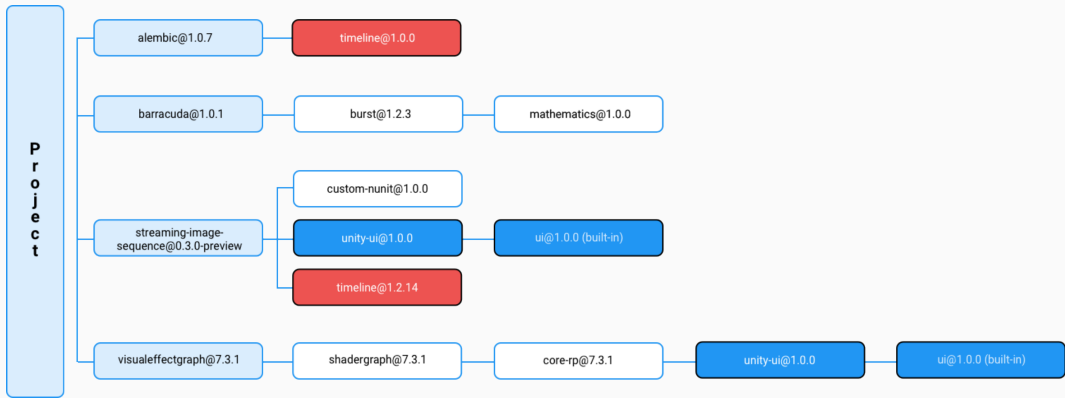
- `requirements.txt`, `setup.py`, and `setup.cfg` died a long time ago
- Use modern build tools
- `pyproject.toml`

- Package manager
- Environment manager
- Lockable packages
- Mostly accepted by Python community
- <https://python-poetry.org/>

Practical Example

- `https://github.com/UCSD-E4E/PyHa.git`
- Use `origin/poetry_demo`

Which dependencies to add?



5

⁵<https://docs.unity3d.com/Manual/upm-conflicts.html>

Defining dependencies

- <https://python-poetry.org/docs/dependency-specification/>
- Semantic Versioning is ideal
- But it depends!
- General philosophy - err on requiring manual intervention later

Using Poetry with conda

- conda doesn't provide lockfiles!
- Introducing conda-lock (<https://pypi.org/project/conda-lock/>)
- Only if you need conda only packages, otherwise, use vanilla poetry
- `environment.yml` only contains conda only packages and poetry
- `conda-lock -f environment.yml -p osx-64 -p linux-64 -p win-64` to create OS-specific lockfiles
- `conda create -f conda-[os].lock` to create conda environment
- `conda activate` to activate environment
- `poetry install` to install poetry lockfile

Additional tools to use with Poetry

- `semantic_release`