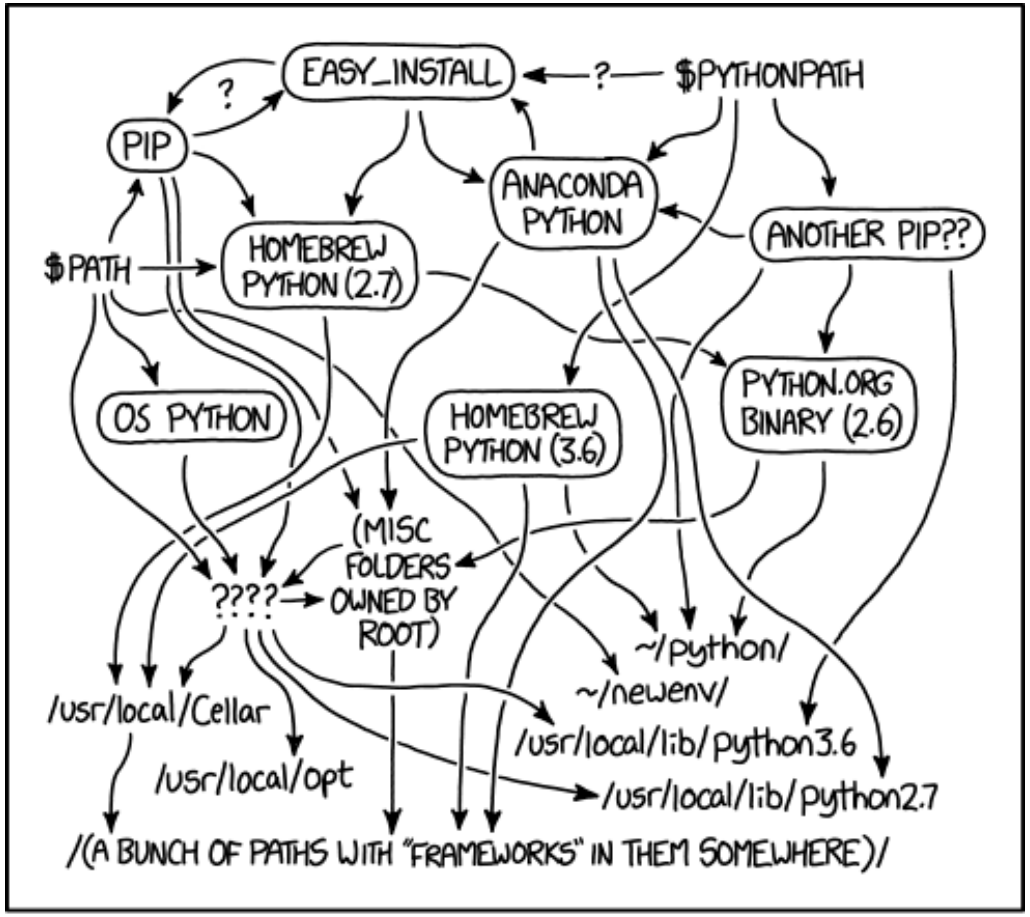
Supercharge your Python environments using Pixi

Barber Vos

Python – a mess

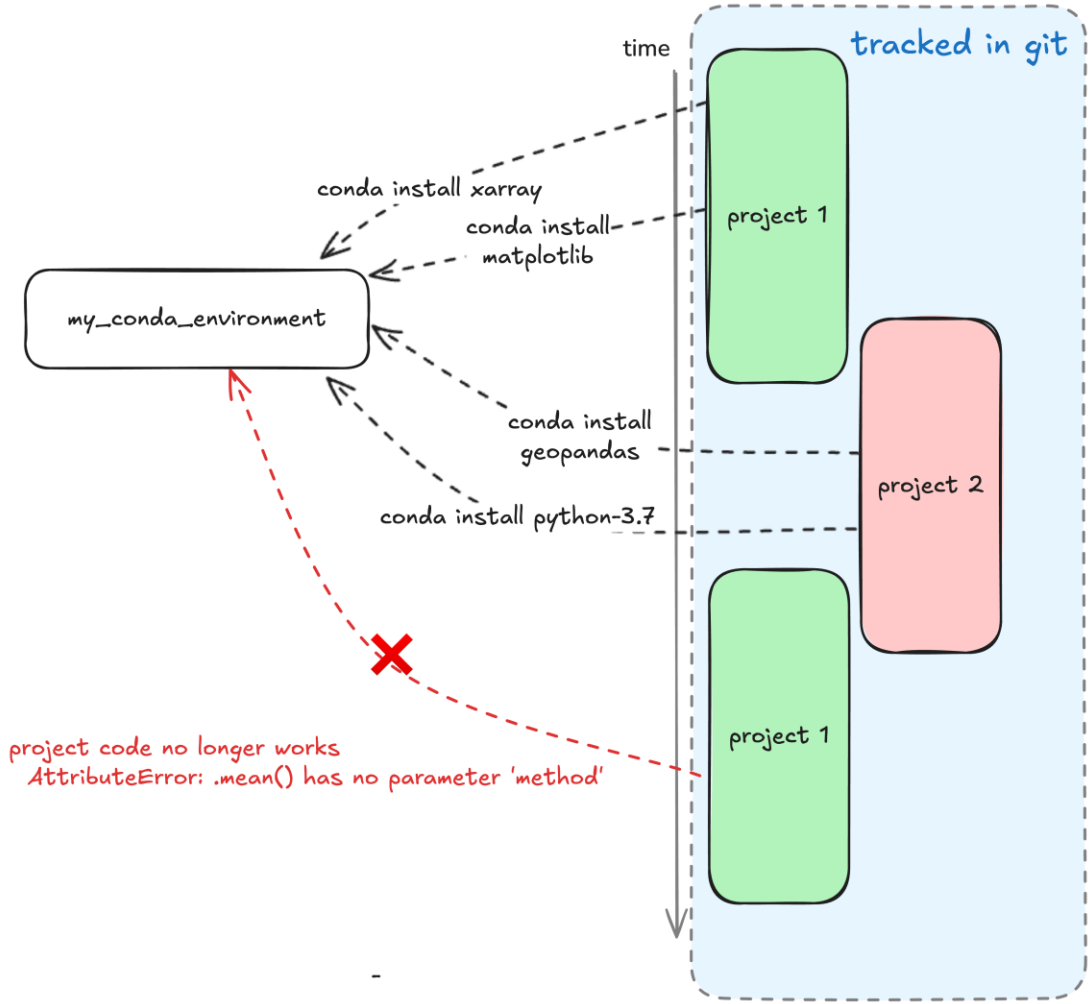


MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Conda helps a lot!

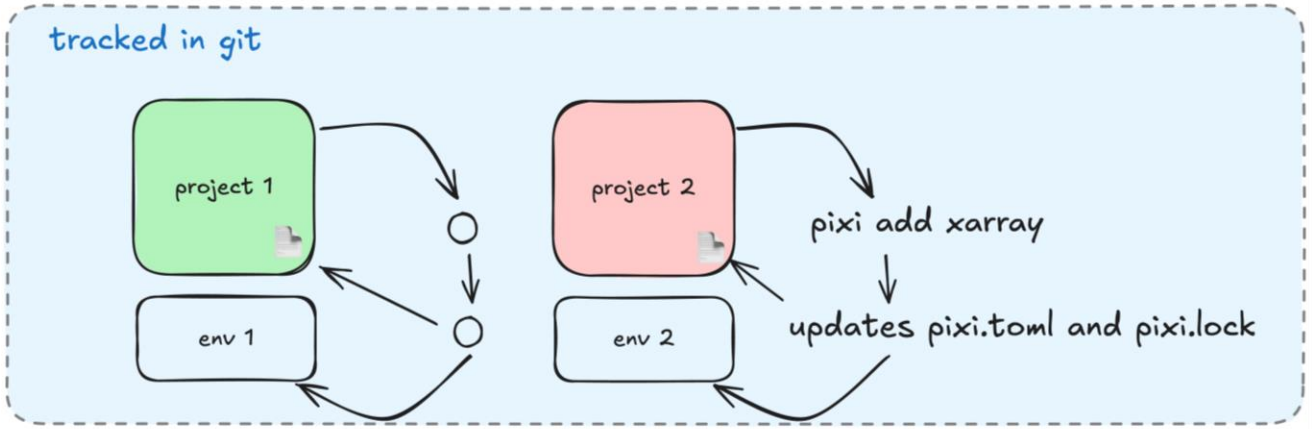
- Python is a package
- Virtual environments

But there's still pain points

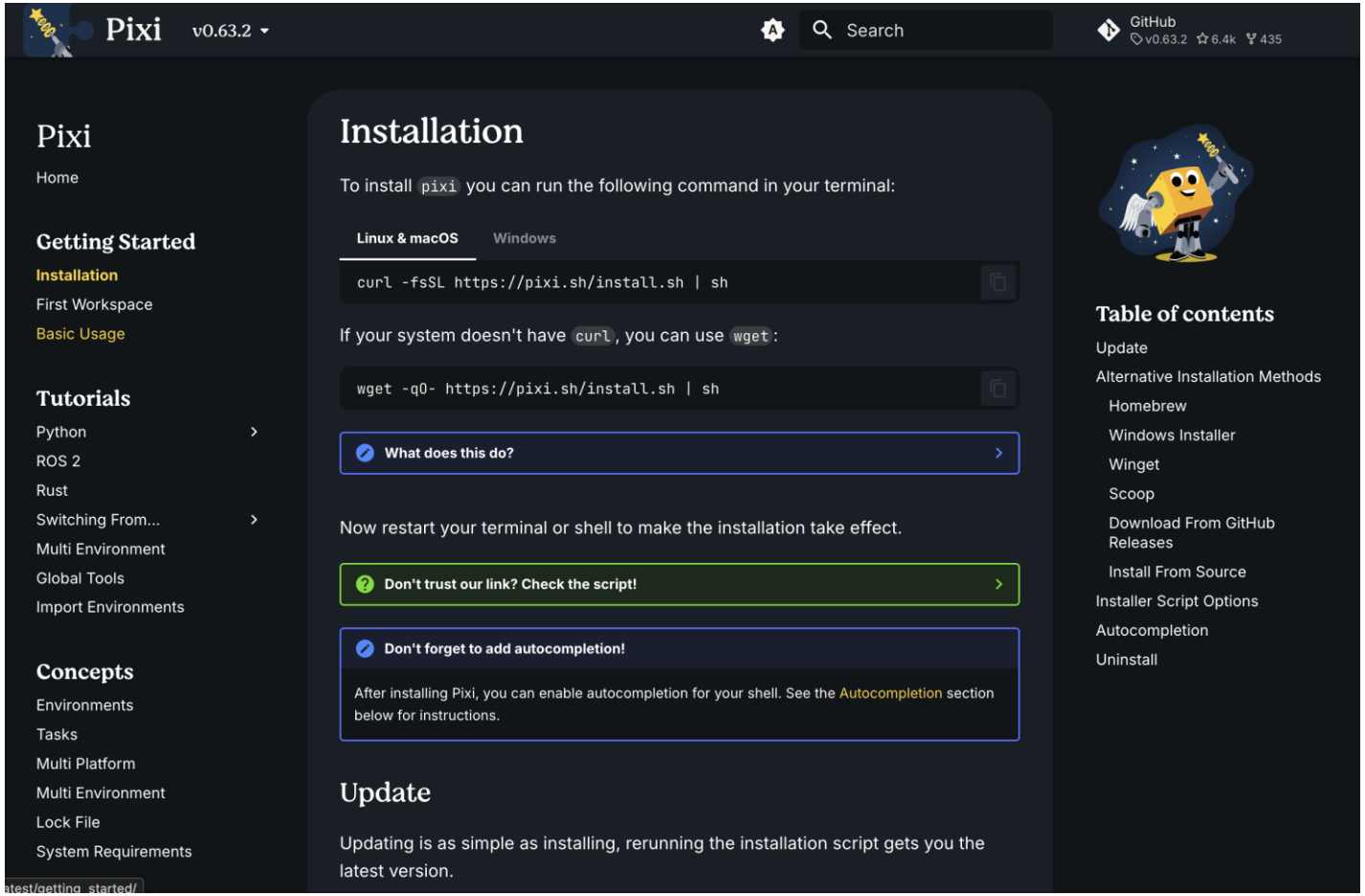


Advantages of Pixi

- Python is a package
- Virtual environment for each project
- .toml file: easily see which dependencies you are using
- Collaboration: share an environment
- Reproduction: exclude newer dependencies



Install Pixi



Virtual environment

Create a folder

```
mkdir my-project
cd my-project
```

Create a folder and go into the folder

Initialize pixi

```
pixi init
```

This creates a new Pixi project in the current folder.

Add python and packages

Add Python and Packages

Add python:

```
pixi add python
```

Add a specific Python version:

```
pixi add python=3.11
```

Add additional packages:

```
pixi add numpy pandas
```

.toml file

Project Files

Pixi manages two important files:

- pixi.toml – project configuration and dependency declarations
- pixi.lock – exact resolved versions for reproducibility

Create and Activate the Environment

Create and activate the environment defined in pixi.toml:

```
pixi shell
```

pixi.toml

```toml
[workspace]
channels = ["conda-forge"]
name = "conv_parametrization"
platforms = ["osx-arm64"]
version = "0.1.0"

[tasks]

[dependencies]
python = "3.13.*"
numpy = ">=2.4.0,<3"
xarray = ">=2025.12.0,<2026"
ipdb = ">=0.13.13,<0.14"
matplotlib = ">=3.10.8,<4"
scipy = ">=1.16.3,<2"
gsw = ">=3.6.20,<4"
ffmpeg = ">=8.0.1,<9"

```

Reproducibility

Reproducible Dependencies (Time-Based)

To ensure packages are no newer than a certain date, use a dated channel snapshot in pixi.toml.

```
exclude-newer = "2023-02-01"
```

Update the Environment

After changing pixi.toml:

```
pixi install
```

And way more!

- Ask Nick :)