

# Supercharge your Python environments using Pixi

Barber Vos

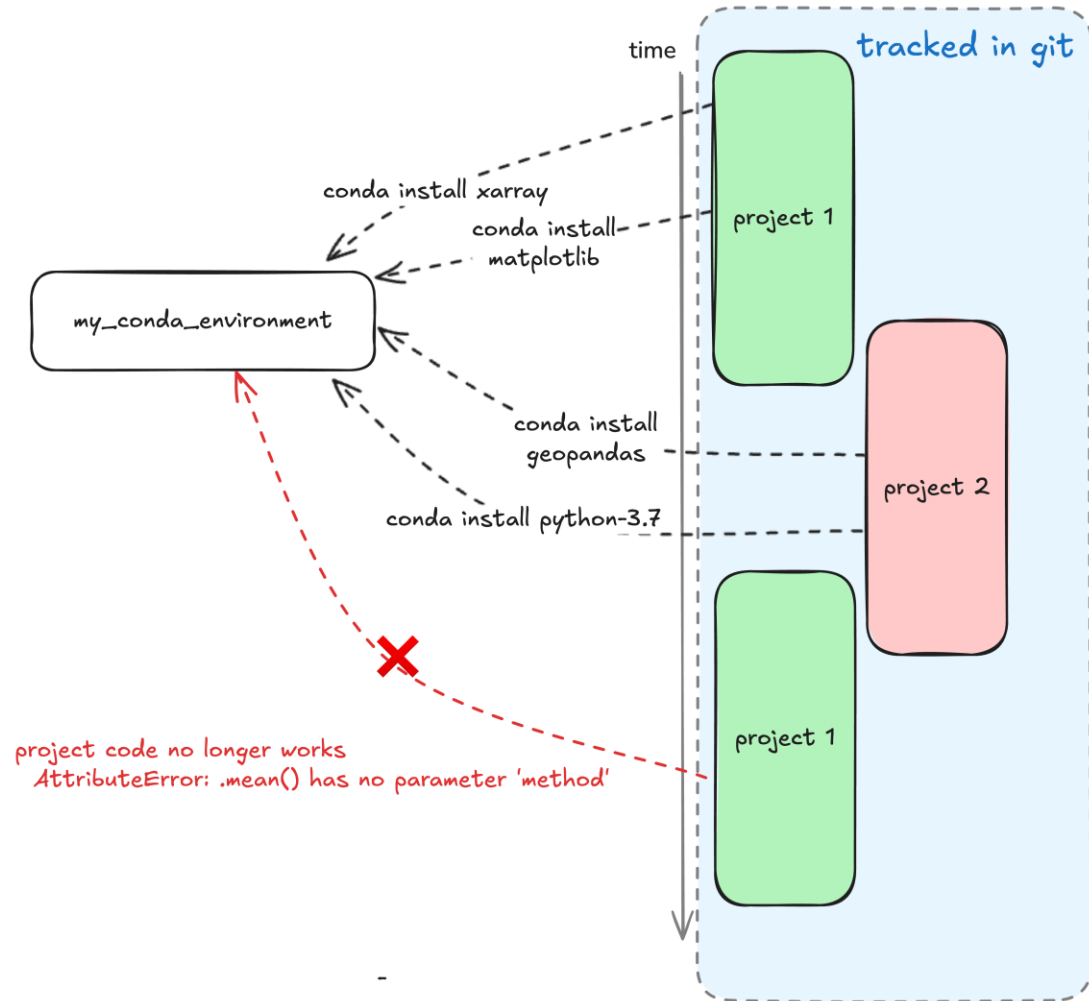




# Conda helps a lot!

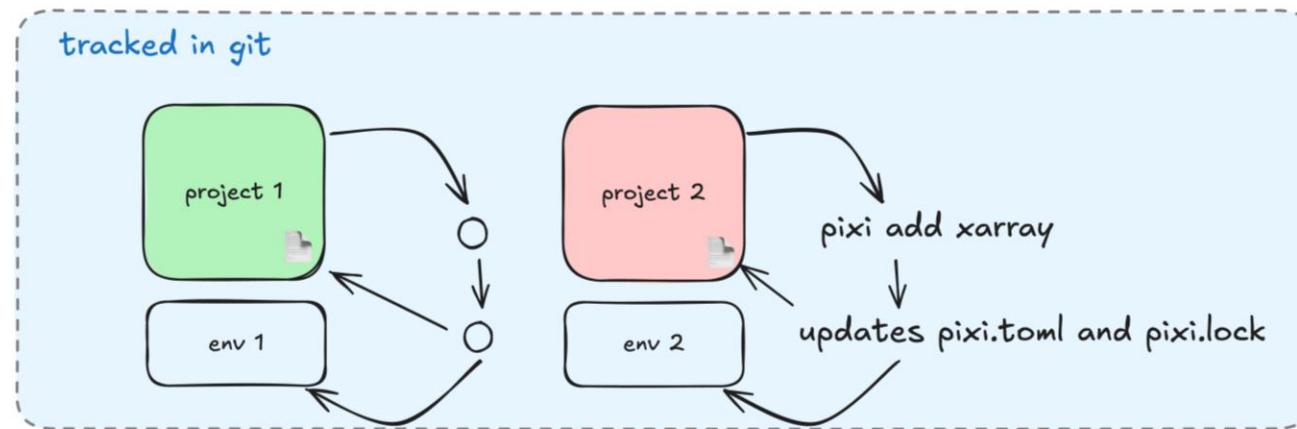
- Python is a package
- Virtual environments

# But there's still pain points

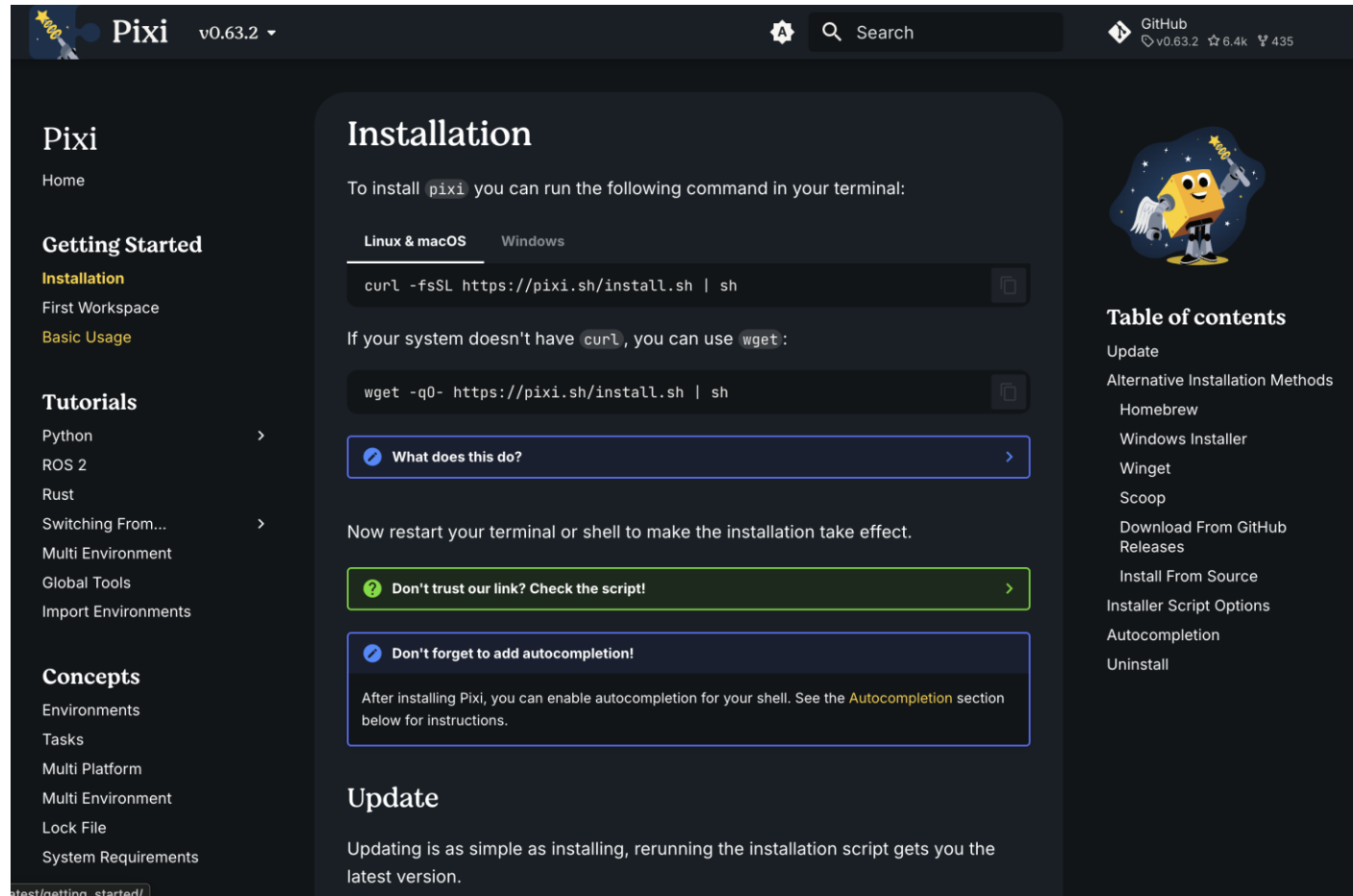


# Advantages of Pixi

- 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



The screenshot shows the GitHub page for Pixi v0.63.2. The page is dark-themed and features a sidebar on the left with navigation links for Home, Getting Started (Installation, First Workspace, Basic Usage), Tutorials (Python, ROS 2, Rust, Switching From..., Multi Environment, Global Tools, Import Environments), and Concepts (Environments, Tasks, Multi Platform, Multi Environment, Lock File, System Requirements). The main content area is titled "Installation" and provides instructions for installing Pixi on Linux & macOS and Windows. It includes terminal commands for using curl and wget, and three callout boxes: "What does this do?", "Don't trust our link? Check the script!", and "Don't forget to add autocompletion!". The "Update" section explains how to update Pixi by rerunning the installation script. On the right, there is a "Table of contents" with links to Update, Alternative Installation Methods (Homebrew, Windows Installer, Winget, Scoop, Download From GitHub Releases, Install From Source, Installer Script Options, Autocompletion, Uninstall), and a cartoon mascot character.

**Pixi** v0.63.2

GitHub v0.63.2 ☆ 6.4k 🗨 435

## Pixi

Home

### Getting Started

- Installation**
- First Workspace
- Basic Usage

### Tutorials

- Python >
- ROS 2
- Rust
- Switching From... >
- Multi Environment
- Global Tools
- Import Environments

### Concepts

- Environments
- Tasks
- Multi Platform
- Multi Environment
- Lock File
- System Requirements

## Installation

To install `pixi` you can run the following command in your terminal:

Linux & macOS Windows

```
curl -fsSL https://pixi.sh/install.sh | sh
```

If your system doesn't have `curl`, you can use `wget`:

```
wget -qO- https://pixi.sh/install.sh | sh
```

**What does this do?**

Now restart your terminal or shell to make the installation take effect.

**Don't trust our link? Check the script!**

**Don't forget to add autocompletion!**

After installing Pixi, you can enable autocompletion for your shell. See the [Autocompletion](#) section below for instructions.

## Update

Updating is as simple as installing, rerunning the installation script gets you the latest version.

### Table of contents

- Update
- Alternative Installation Methods
  - Homebrew
  - Windows Installer
  - Winget
  - Scoop
  - Download From GitHub Releases
  - Install From Source
  - Installer Script Options
  - Autocompletion
  - Uninstall

# 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
```

```
gear pixi.toml
1  [workspace]
2  channels = ["conda-forge"]
3  name = "conv_parametrization"
4  platforms = ["osx-arm64"]
5  version = "0.1.0"
6
7  [tasks]
8
9  [dependencies]
10 python = "3.13.*"
11 numpy = ">=2.4.0,<3"
12 xarray = ">=2025.12.0,<2026"
13 ipdb = ">=0.13.13,<0.14"
14 matplotlib = ">=3.10.8,<4"
15 scipy = ">=1.16.3,<2"
16 gsw = ">=3.6.20,<4"
17 ffmpeg = ">=8.0.1,<9"
18
```

# 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 :)