

How to Build Python & Rust Packages

With Pixi

About Me

-  Julian Hofer
-  Background in Physics
-  Thinks that languages are cool
-  Loves to talk about dependency management



Prefix.dev

What is Conda?

-  Package ecosystem:
 - Cross-platform
 - Cross-language
-  Commonly used for scientific Python
-  Decentralized channels like:
 - conda-forge
 - bioconda
 - fastai



Installing NumPy via Pip

From the NumPy contributor docs:

- Install NumPy as a user:

```
pip install numpy
```

- Install NumPy as a developer:

```
# Debian  
sudo apt build-dep numpy  
# Fedora  
sudo dnf builddep numpy  
# Arch  
sudo pacman -S gcc-fortran openblas pkgconf  
# macOS  
brew install openblas pkg-config gfortran
```



Finally

```
pip install . --no-build-isolation
```



Prefix.dev

Installing NumPy via Conda

From the **NumPy** contributor docs:

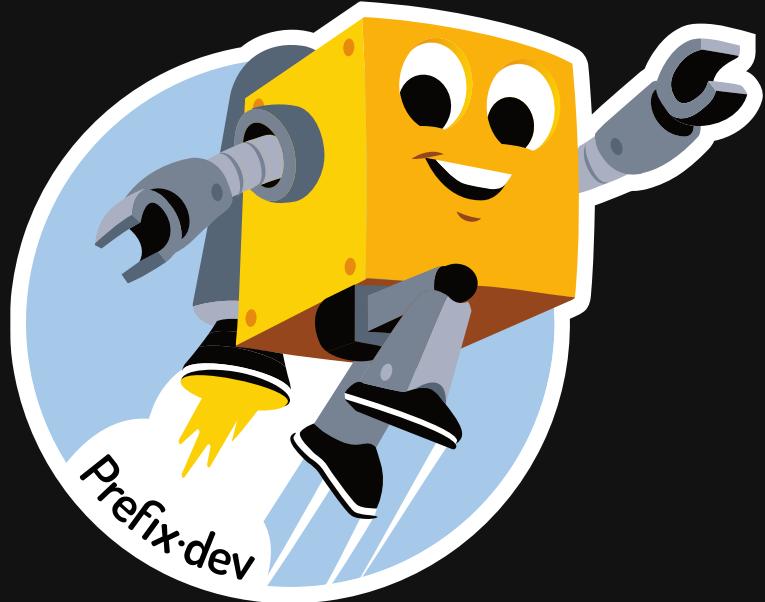
If you are using conda, you can skip the steps in this section - with the exception of installing compilers for Windows or the Apple Developer Tools for macOS. All other dependencies will be installed automatically [...]

```
conda env create -f environment.yml  
  
# or  
pixi init --import environment.yml
```



Introducing Pixi

-  Fast
-  Open-Source
-  Workflow management
-  Multi-environments
-  Reproducible thanks to lock-files
-  Supports conda and PyPI ecosystem



Ecosystem Comparison

Feature	conda	PyPI
Official Python Index	⚠️	✓
Cross-Platform	✓	✓
Cross-Language	✓	⚠️
Decentralized	✓	⚠️
Traditional Package Manager	conda	pip (conda)
Modern Package Manager	pixi	uv (pixi)



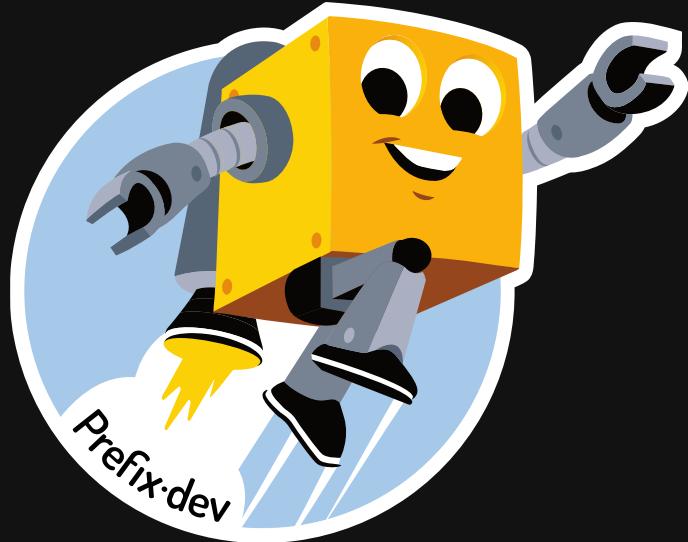
Demo Time

- Interactive Python
- Put code into a file
- Manage dependencies & tasks with Pixi manifest
- Organize code as package
- Add Rust to the mix



Conclusion

- Modernize your workflow
 - Reproducible
 - Fast
 - Cross language
- One tool for all your development needs
- Free & Open-Source



Thank you for your attention!



Pixi Website



LinkedIn



Discord