

Simplificando el setup de Python con uv

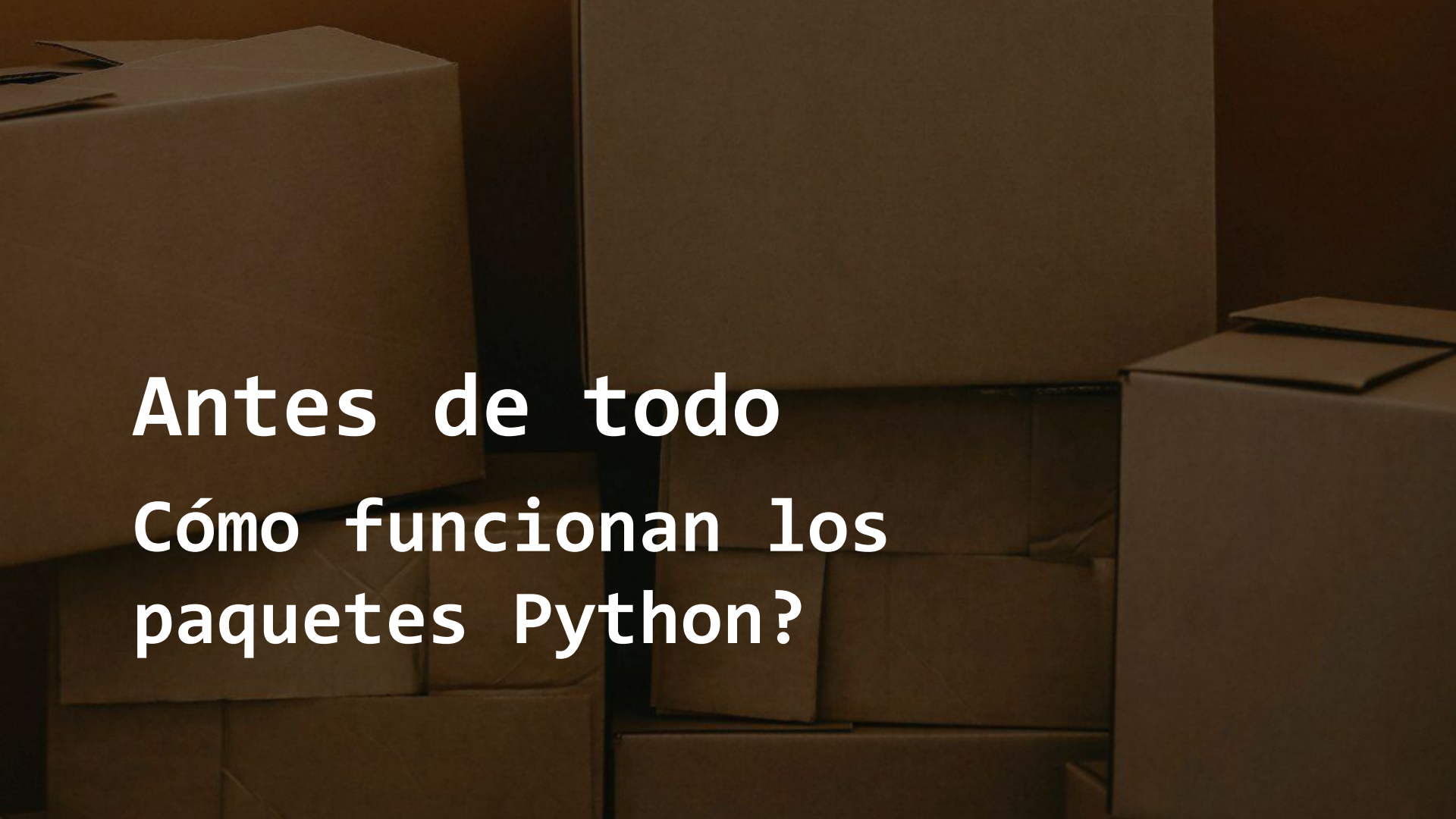
Ana Rute Mendes - Feb 2025 Python Coruña

Hola :)



Software Developer en **Igalia**.
Soy generalista, actualmente hago
un poco de todo y tengo un cariño
especial por Python.
En Coruña desde 2022. 🇧🇷

bsky.app/profile/anarute.com
anarute.com



Antes de todo
Cómo funcionan los
paquetes Python?

Paquetes y módulos en Python

"Un módulo es un fichero conteniendo definiciones y declaraciones de Python." [[docs](#)]

Un paquete puede ser un módulo o conjunto de submódulos python organizado con la finalidad de ser reutilizable.

Ejemplo básico de un paquete regular:

```
proyecto/  
└── src/  
    ├── example_package/  
    │   ├── __init__.py  
    │   └── modulo.py
```

Cómo utilizar paquetes

Puedes instalar un paquete con

```
pip install paquete
```

Y utilizar en otros proyectos

```
import example_package  
# or  
from example_package import modulo
```

O si es ejecutable, puedes usarlo directamente desde la línea de comandos.

[Ayuda](#)[Patrocinadores](#)[Acceder](#)[Registrarse](#)

Encuentre, instale y publique paquetes de Python con el Índice de paquetes de Python



O bien, [explore los proyectos](#)

607.329 proyectos

6.572.136 versiones

13.292.819 archivos

900.837 usuarios

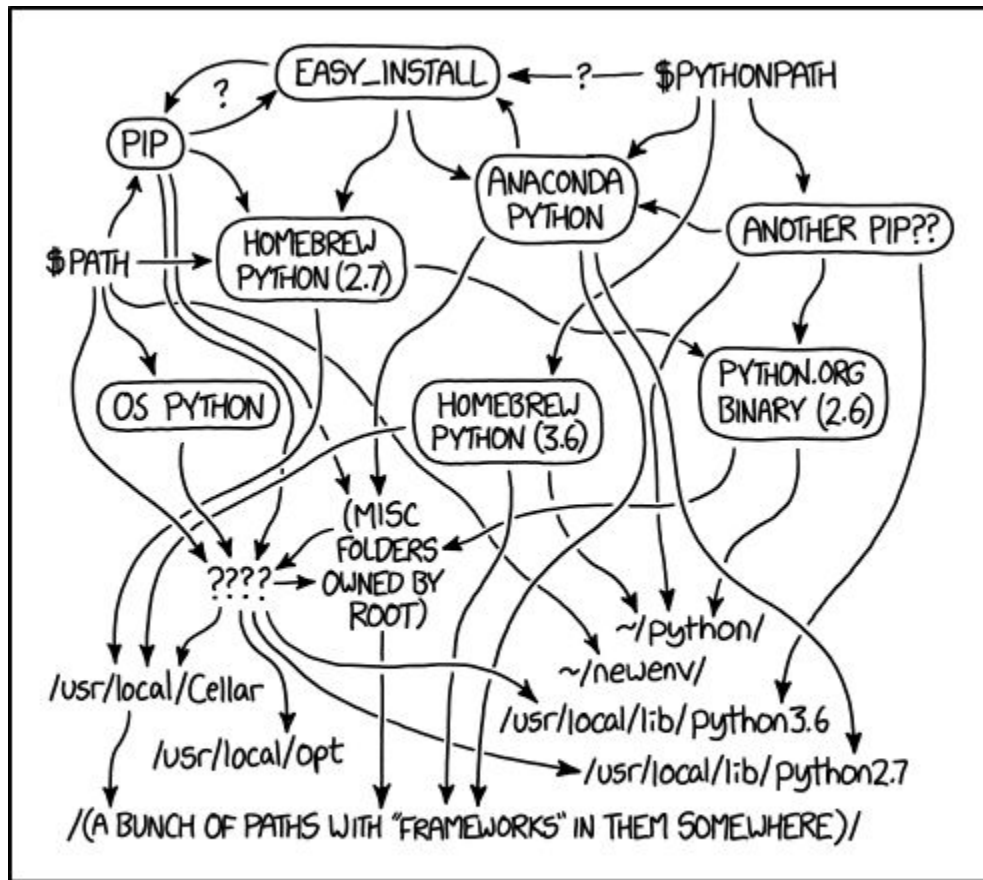


El Índice de paquetes de Python (PyPI) es un repositorio de *software* para el lenguaje de programación Python.

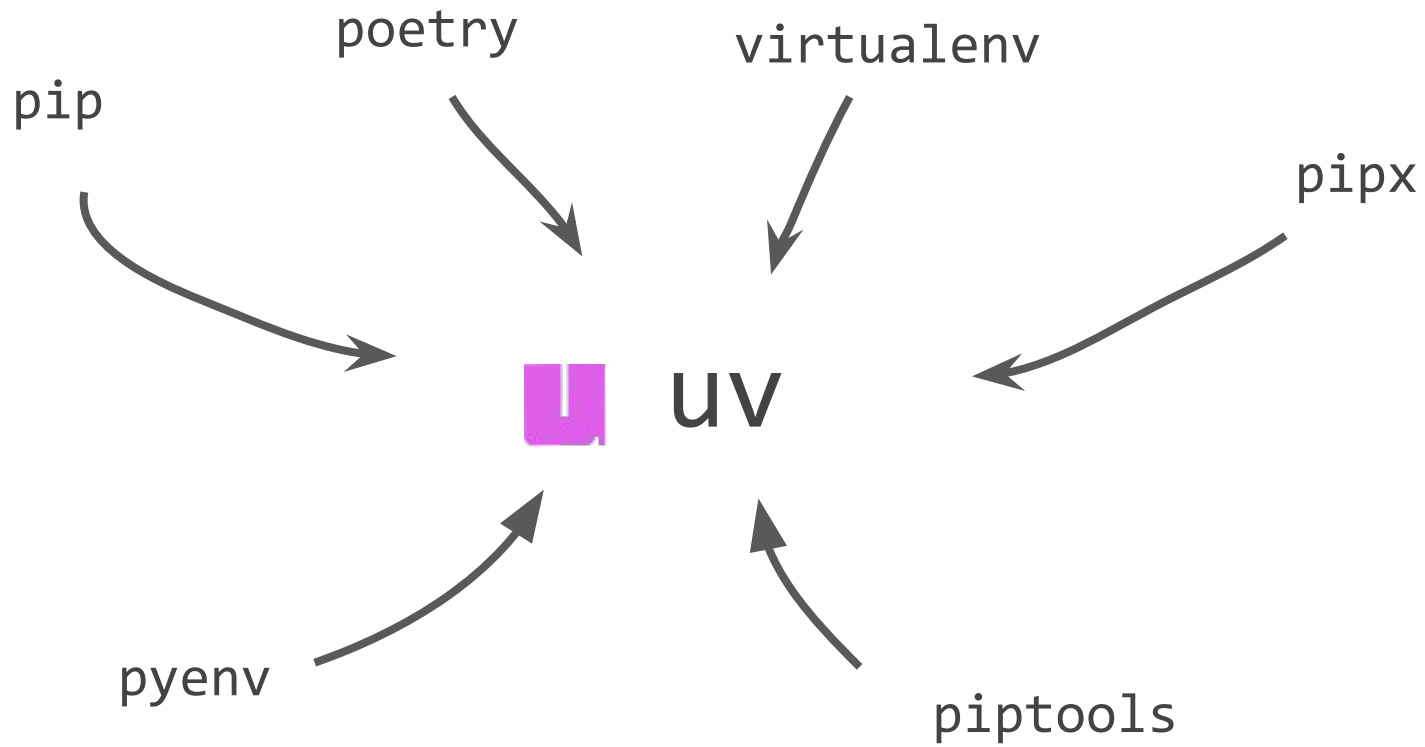
PyPI le ayuda a encontrar e instalar programas desarrollados y compartidos por la comunidad de Python. [Aprenda a instalar paquetes](#).

Los autores de paquetes utilizan PyPI para distribuir sus programas. [Aprenda a empaquetar su código en Python para PyPI](#).

Si ya tenemos pip,
pyenv, poetry, etc por
qué usar uv?

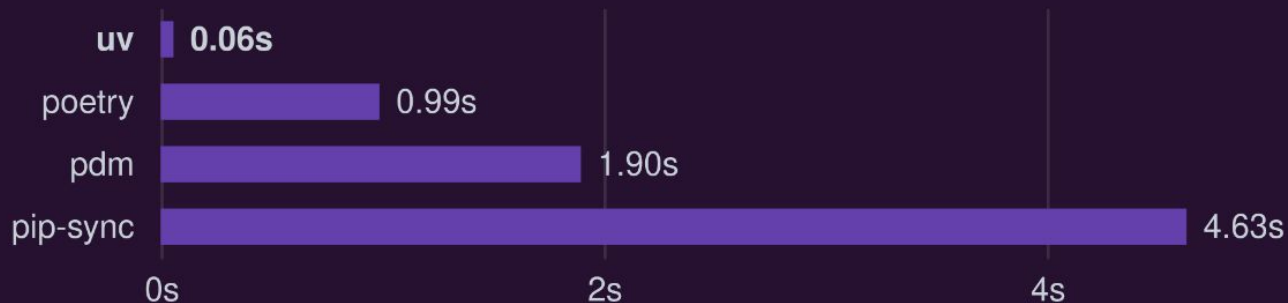


MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



uv

An extremely fast Python package and project manager, written in Rust.



*Installing **Trio**'s dependencies with a warm cache.*

main



Go to file

Code

About

An extremely fast Python package and project manager, written in Rust.

docs.astral.sh/uv

python resolver packaging uv

- Readme
- Apache-2.0, MIT licenses found
- Security policy
- Activity
- Custom properties
- 38.7k stars
- 77 watching

zanieb Fix credential caching for index roots ... 6dfac45 · 4 hours ago

.cargo Don't build uv-dev by default (#... 6 months ago

.config Abort tests running for more t... 9 months ago

lock... 17 hours ago

th... 11 months ago

index... 4 hours ago

ne ... 4 hours ago

tra... last week

Make build backend type anno... last month

Languages



Rust 97.9% Python 1.9%
Shell 0.1% Mustache 0.1%
Batchfile 0.0% Nushell 0.0%

python/uv

Inicializando proyectos con uv

```
anarute@tuca ~/code/uv-talk
└─> uv init test-project
Initialized project `test-project` at `/home/anarute/code/uv-talk/test-project`
anarute@tuca ~/code/uv-talk
└─> cd test-project
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> ls -la
total 16
drwxr-xr-x. 1 anarute anarute 120 Feb  9 15:45 .
drwxr-xr-x. 1 anarute anarute  24 Feb  9 15:45 ..
drwxr-xr-x. 1 anarute anarute  82 Feb  9 15:45 .git
-rw-r--r--. 1 anarute anarute 109 Feb  9 15:45 .gitignore
-rw-r--r--. 1 anarute anarute  90 Feb  9 15:45 hello.py
-rw-r--r--. 1 anarute anarute 158 Feb  9 15:45 pyproject.toml
-rw-r--r--. 1 anarute anarute   5 Feb  9 15:45 .python-version
-rw-r--r--. 1 anarute anarute   0 Feb  9 15:45 README.md
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> cat .python-version
3.13
```

Inicializando proyectos con uv

```
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> cat pyproject.toml
[project]
name = "test-project"
version = "0.1.0"
description = "Add your description here"
readme = "README.md"
requires-python = ">=3.13"
dependencies = []

anarute@tuca ~/code/uv-talk/test-project <main*>
└─> cat hello.py
def main():
    print("Hello from test-project!")

if __name__ == "__main__":
    main()

anarute@tuca ~/code/uv-talk/test-project <main*>
└─> █
```

Inicializando proyectos con uv

```
anarute@tuca ~/code/uv-talk/test-project <main*>  
└─▶ uv run hello.py  
Using CPython 3.13.1  
Creating virtual environment at: .venv  
Hello from test-project!
```

Automáticamente crea el .venv y el lockfile

```
-rw-r--r--. 1 anarute anarute 119 Feb  9 16:32 uv.lock  
drwxr-xr-x. 1 anarute anarute  86 Feb  9 16:32 .venv
```

Cómo añadir paquetes al proyecto

```
anarute@tuca ~/code/uv-talk/test-project <main*>  
└─> uv add pandas  
Using CPython 3.13.1  
Creating virtual environment at: .venv  
Resolved 7 packages in 8ms  
Installed 6 packages in 52ms  
+ numpy==2.2.2  
+ pandas==2.2.3  
+ python-dateutil==2.9.0.post0  
+ pytz==2025.1  
+ six==1.17.0  
+ tzdata==2025.1
```

Cómo añadir paquetes al proyecto

```
anarute@tuca ~/code/uv-talk/test-project <main*>  
└─ cat pyproject.toml  
[project]  
name = "test-project"  
version = "0.1.0"  
description = "Add your description here"  
readme = "README.md"  
requires-python = ">=3.13"  
dependencies = [  
    "pandas>=2.2.3",  
]
```


Cómo gestionar las dependencias del proyecto

```
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv sync
Resolved 7 packages in 1ms
Audited 6 packages in 0.01ms
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv lock
Resolved 7 packages in 0.88ms
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> █
```

Inicializando scripts con uv

```
anarute@tuca ~/code/uv-talk
└─> uv init --script test_script.py
Initialized script at `test_script.py`
anarute@tuca ~/code/uv-talk
└─> cat test_script.py
# /// script
# requires-python = ">=3.13"
# dependencies = []
# ///

def main() -> None:
    print("Hello from test_script.py!")

if __name__ == "__main__":
    main()
```

Inicializando scripts con uv

```
anarute@tuca ~/code/uv-talk  
└─▶ uv run test_script.py  
Reading inline script metadata from `test_script.py`  
Hello from test_script.py!
```

Creando paquetes con uv

```
anarute@tuca ~/code/uv-talk
➤ uv init --app --package que_horas_son
Initialized project `que-horas-son` at `/home/anarute/code/uv-talk/que_horas_son`
```

```
anarute@tuca ~/code/uv-talk/que_horas_son <main*>
➤ uv tool install . -e
Resolved 1 package in 0.91ms
  Built que-horas-son @ file:///home/anarute/code/uv-talk/que_horas_son
Prepared 1 package in 477ms
Installed 1 package in 0.87ms
+ que-horas-son==0.1.0 (from file:///home/anarute/code/uv-talk/que_horas_son)
Installed 1 executable: que-horas-son
```

Creando paquetes con uv

```
anarute@tuca ~/code/uv-talk/que_horas_son <main*>
└─ cat pyproject.toml
[project]
name = "que-horas-son"
version = "0.1.0"
description = "Add your description here"
readme = "README.md"
authors = [
    { name = "Ana Rute Mendes", email = "ana@igalia.com" }
]
requires-python = ">=3.13"
dependencies = []

[project.scripts]
que-horas-son = "que_horas_son:main"

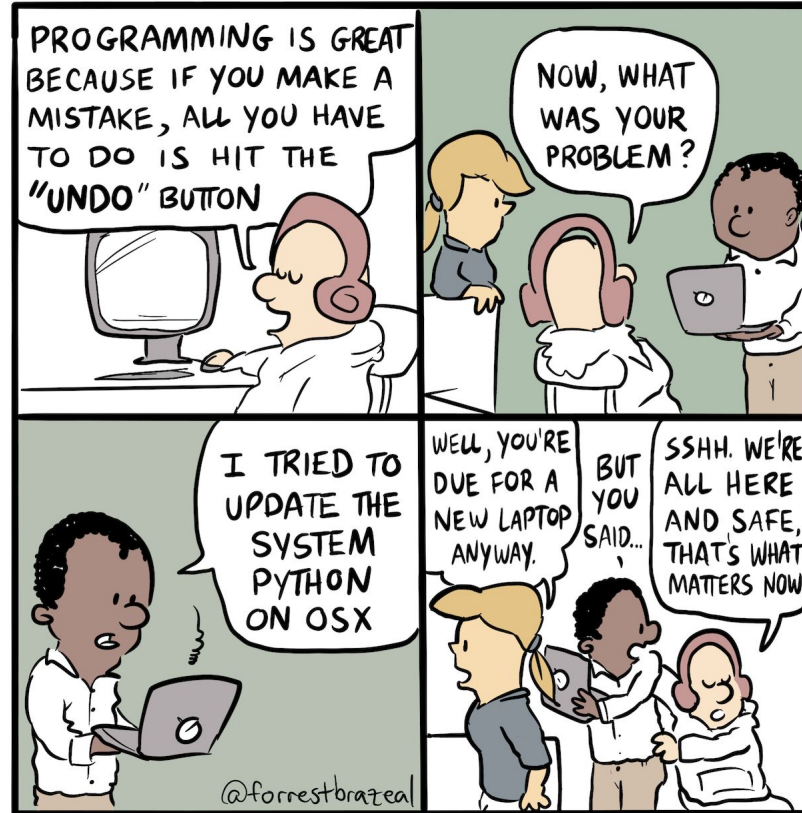
[build-system]
requires = ["hatchling"]
build-backend = "hatchling.build"
```

Creando paquetes con uv

```
anarute@tuca ~/code/uv-talk/que_horas_son <main*>
└─> uv tool install . -e
Resolved 1 package in 1ms
Installed 1 package in 2ms
+ que-horas-son==0.1.0 (from file:///home/anarute/code/uv-talk/que_horas_son)
Installed 1 executable: que-horas-son
anarute@tuca ~/code/uv-talk/que_horas_son <main*>
└─> que-horas-son
¿Qué horas son, mi corazón?

2025-02-15 11:32:24.969889
```

Virtualenvs y Python versions



Virtualenvs y Python versions

Con `uv` es posible instalar y gestionar varias versiones de Python y elegir cual usar al ejecutar un script o programa.

```
anarute@tuca ~/code/uv-talk
└─> uv python --help
Manage Python versions and installations

Usage: uv python [OPTIONS] <COMMAND>

Commands:
  list      List the available Python installations
  install   Download and install Python versions
  find      Search for a Python installation
  pin       Pin to a specific Python version
  dir       Show the uv Python installation directory
  uninstall Uninstall Python versions
```


Virtualenvs y Python versions

```
anarute@tuca ~/code/uv-talk
```

```
➤ cat test-py3-8.py
```

```
test = 'hello'
```

```
print(f"{test=}")
```

```
anarute@tuca ~/code/uv-talk
```

```
➤ uv run --python 3.7 test-py3-8.py
```

```
File "<fstring>", line 1
```

```
(test=)
```

```
^
```

```
SyntaxError: invalid syntax
```

```
anarute@tuca ~/code/uv-talk
```

```
➤ uv run --python 3.8 test-py3-8.py
```

```
test='hello'
```

Virtualenvs y Python versions

```
anarute@tuca ~/code/uv-talk
```

```
➤ cat test-py3-8.py
```

```
test = 'hello'
```

```
print(f"{test=}")
```

```
anarute@tuca ~/code/uv-talk
```

```
➤ uv run --python 3.7 test-py3-8.py
```

```
File "<fstring>", line 1
```

```
(test=)
```

```
^
```

```
SyntaxError: invalid syntax
```

```
anarute@tuca ~/code/uv-talk
```

```
➤ uv run --python 3.8 test-py3-8.py
```

```
test='hello'
```

Virtualenvs y Python versions

```
anarute@tuca ~/code/uv-talk
└─> uv venv --python 3.12.0
Using CPython 3.12.0
Creating virtual environment at: .venv
Activate with: source .venv/bin/activate
anarute@tuca ~/code/uv-talk
└─> uv python pin 3.11
warning: No interpreter found for Python 3.11 in managed installations or search path
Pinned `.python-version` to `3.11`
anarute@tuca ~/code/uv-talk
└─> uv run test_script.py
Reading inline script metadata from `test_script.py`
warning: The Python request from `.python-version` resolved to Python 3.11.11, which is incompatible with the script's Python requirement: `>=3.13`
Hello from test_script.py!
```

Virtualenvs y Python versions

```
anarute@tuca ~/code/uv-talk  
└─► uv run --no-project --python 3.8 python  
Python 3.8.20 (default, Oct 2 2024, 16:34:12)  
[Clang 18.1.8 ] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Virtualenvs y Python versions

```
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv venv
Using CPython 3.13.1
Creating virtual environment at: .venv
Activate with: source .venv/bin/activate
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv export --format requirements-txt > requirements.txt

Resolved 7 packages in 0.75ms
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> source .venv/bin/activate
(test-project) anarute@tuca ~/code/uv-talk/test-project <main*>
└─> pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy==2.2.2 in /home/anarute/.local/lib/python3.13/site-packages (2.2.2)
Requirement already satisfied: pandas==2.2.3 in /home/anarute/.local/lib/python3.13/site-packages (2.2.3)
Requirement already satisfied: python-dateutil==2.9.0.post0 in /home/anarute/.local/lib/python3.13/site-packages (2.9.0.post0)
Requirement already satisfied: pytz==2025.1 in /usr/lib/python3.13/site-packages (2025.1)
Requirement already satisfied: six==1.17.0 in /home/anarute/.local/lib/python3.13/site-packages (1.17.0)
Requirement already satisfied: tzdata==2025.1 in /home/anarute/.local/lib/python3.13/site-packages (2025.1)
(test-project) anarute@tuca ~/code/uv-talk/test-project <main*>
└─>
```

Virtualenvs y Python versions

```
(test-project) └─anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv pip sync requirements.txt
Resolved 6 packages in 68ms
Installed 6 packages in 38ms
+ numpy==2.2.2
+ pandas==2.2.3
+ python-dateutil==2.9.0.post0
+ pytz==2025.1
+ six==1.17.0
+ tzdata==2025.1
(test-project) └─anarute@tuca ~/code/uv-talk/test-project <main*>
└─> █
```

Más allá de dependencias...

Muchos paquetes vienen con un programa ejecutable y uv nos permite instalarlos localmente y usarlos con **uv tool**.

```
anarute@tuca ~/code/uv-talk/test-project <main*>
└─> uv tool --help
Run and install commands provided by Python packages

Usage: uv tool [OPTIONS] <COMMAND>

Commands:
  run          Run a command provided by a Python package
  install      Install commands provided by a Python package
  upgrade      Upgrade installed tools
  list         List installed tools
  uninstall    Uninstall a tool
  update-shell Ensure that the tool executable directory is on the `PATH`
  dir          Show the path to the uv tools directory
```

uvx funciona como **pipx** y es un alias para **uv tool**

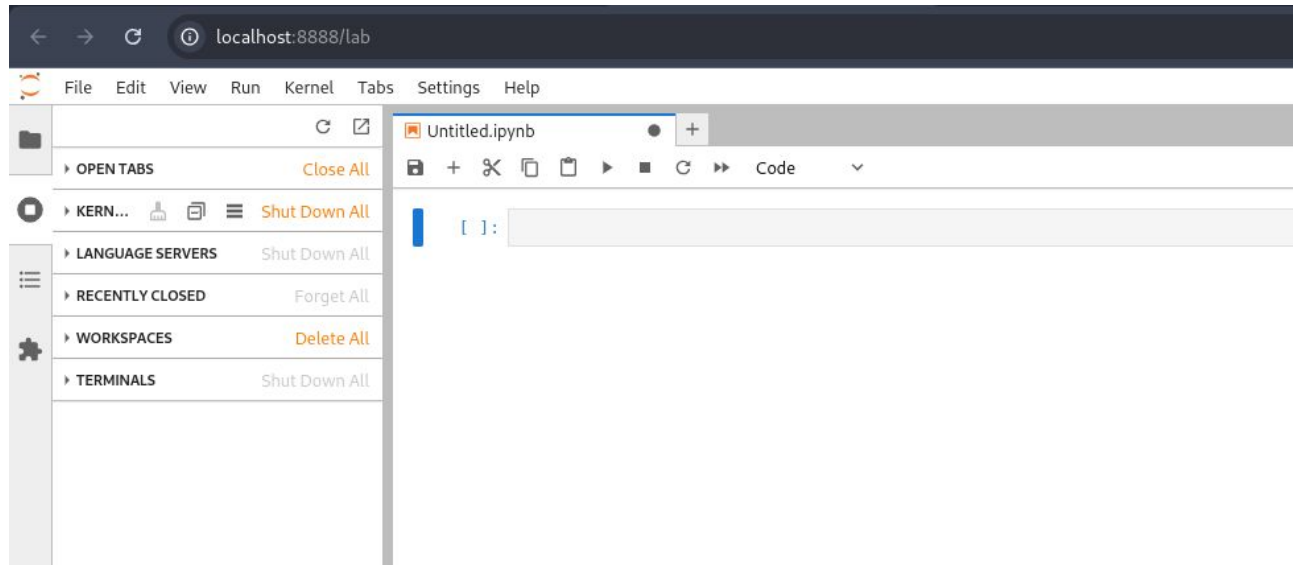
Más allá de dependencias...

```
anarute@tuca ~/code/uv-talk/test-project <main*>  
└─▶
```


Bonus

Para ejecutar pytorch en un Jupyter Notebook:

```
uv run --python 3.12 --with "torch~=2.4" --with jupyter  
--with "numpy<2" jupyter lab
```



Clean up

Para limpiar el cache, puedes coger `uv cache clean`

```
(test-project) └─anarute@tuca ~/code/uv-talk  
└─► uv cache clean  
Clearing cache at: /home/anarute/.cache/uv  
Removed 114450 files (18.2GiB)
```

Moitas grazas! :)

`anarutecvm@gmail.com`

`bsky.app/profile/anarute.com`