

# Python Virtual Environment

## Tutorials

- [How to Install and Manage Python Versions in Linux - Make Tech Easier](#)

## UV

- ??: ?? + ??????????
- ??: [uv](#)
- ?????? root ??
- ?? uv ??????? pip, pip-tools, pipx, poetry, pyenv, twine, virtualenv ???

```
# Install with curl or wget
curl -Lsf https://astral.sh/uv/install.sh | sh
wget -qO- https://astral.sh/uv/install.sh | sh

# Instal with pip
pip install uv
```

```
# To install the latest Python version:
uv python install

# To install a specific Python version:
uv python install 3.12

# To reinstall uv-managed Python versions
uv python install --reinstall

# To view available and installed Python versions:
uv python list
uv python list --managed-python
uv python list --no-managed-python
```

```
# Crerate a new project
uv init hello-world
```

```
cd hello-world

# Alternatively, you can initialize a project in the working directory:
mkdir hello-world
cd hello-world
uv init

# Create virtual environment
uv venv
# Or
uv venv --python=3.12 .venv

# Activate
source .venv/bin/activate

# Exit the VENV
deactivate
```

```
# Add a package
uv add requests
uv add 'requests==2.31.0'

# Add all dependencies from `requirements.txt`.
uv add -r requirements.txt

# To remove a package
uv remove requests
```

???????????????

```
git clone https://path/to/your/project/yourepo.git

cd yourepo/
uv venv # []
source .venv/bin/activate # []
uv sync # [] pyproject.toml + uv.lock []
```

## Pixi

- [Home](#)
- [Pypi: A Smarter Way to Manage Python Environments - KDnuggets](#)

## Miniconda

- [Miniconda - Anaconda](#)

```
# Install
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O
~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.sh

# Post-Install
source ~/miniconda3/bin/activate
conda init --all

# Verify
conda list
conda --version

# Optional: Uninstall
conda deactivate
~/miniconda3/uninstall.sh
```

## Conda

```
# Create a virtual env
conda create -n myproj python=3.11

# Activate the virtual env
conda activate myproj

# Deactivate the virtual env
conda deactivate
```

## Python 3.4+ built-in venv

```
# Install venv
sudo apt install python3-venv

# Enable venv
mkdir myproject
cd myproject
python -m venv .venv

# Activate the venv
source .venv/bin/activate

# Delete the venv
deactivate
rm -rf .venv

# Change the App directory after activating venv
cd /path/to
mv old new
cd new/.venv/bin
old_path="/path/to/old/.venv"
new_path="/path/to/new/.venv"
find ./ -type f -exec sed -i "s|$old_path|$new_path|g" {} \;
cd /path/to/new
source .venv/bin/activate
```

## virtualenv and virtualenvwrapper

```
# Installing virtualenv and virtualenvwrapper
sudo pip install virtualenv virtualenvwrapper

# Update the profile ~/.bashrc
# Add the following lines

# Python virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh

# Reload the profile
source ~/.bashrc
```

```
# Creating python virtual environment
# The py3cv3 is a self-defined name
mkvirtualenv py3cv3 -p python3

# Enter the specified virtual environment
workon py3cv3

# Exit the the specified virtual environment
deactivate

# List all of the environments.
lsvirtualenv

# Remove an environment
rmvirtualenv py3cv3
```

---

Revision #15

Created 2025-12-10 09:50:25 CST by A-Lang (Admin)

Updated 2026-04-30 11:14:27 CST by A-Lang (Admin)