Python: Entornos virtuales

python

- venv
- virtualenv
- pyenv

Una combinación de **pyenv** (para gestión de versiones de python) y **virtualenv** (para encapsular librerias por proyecto) parace una buena solución

virtualenv

```
python3 -m pip install --upgrade pip
pip3 install virtualenv

which virtualenv
virtualenv -p /home/username/opt/python-3.10.1/bin/python3 venv # crea venv

pip3 install --upgrade setuptools # si error setuptools pip failed with error code 1` error

source venv/bin/activate # activar
pip3 install <module> # instalación de módulos
deactivate # desactivar
rm -rf venv # eliminar
```

/via: https://help.dreamhost.com/hc/es/articles/115000695551-Instalar-y-usar-virtualenv-con-Python-3

```
pip install virtualenv

# navegar a directorio y crear con:
python -m venv <environment_name>
# alternativamente
virtualenv -p /home/username/opt/python-3.10.1/bin/python3 <environment_name>

# activar
source <environment_name>/bin/activate

# desactivar
deactivate
```

venv

- https://docs.python.org/es/3/tutorial/venv.html

creación

```
python -m venv tutorial-env
```

- crea una carpeta con la versión de python disponible

activación

- windows:

```
tutorial-env\Scripts\activate
```

    - gestión versiones en windows con comando py ?
- linux:

```
source tutorial-env/bin/activate
```

desactivación

```
deactivate
```

pyenv

instalación

- install:

```
curl https://pyenv.run | bash
```

- configuración:

  .zshrc

```
# pyenv
export PATH="$HOME/.pyenv/bin:$PATH"
eval "$(pyenv init --path)"
eval "$(pyenv virtualenv-init -)"
```

```
export PYENV_ROOT="$HOME/.pyenv"
command -v pyenv >/dev/null || export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init -)"
```

- verificar instalación:

```
pyenv --version
```

uso

- listado versiones disponibles:

```
pyenv install --list | grep " 3\.[67]"
```

    - versiones 3.6.* y 3.7.* en este caso
- instalar una versión en concreto:

```
pyenv install -v 3.6.0
```

    - paquetes para asegurarse una instalación y compilación correctas:

```
sudo apt-get install -y make build-essential libssl-dev zlib1g-dev \
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev
\
libncursesw5-dev xz-utils tk-dev libffi-dev liblzma-dev python-openssl
```

- versiones instaladas:

```
pyenv versions
```

    - `ls ~/.pyenv/versions/`

- establecer versión del sistema:

```
pyenv global 3.9.9
```

- establecer versión en un directorio concreto:

```
pyenv local 3.9.9
```

- desinstalar:

```
pyenv uninstall 2.7.9
```

- ejecutar shell de alguna versión instalada:

```
pyenv shell 2.7.9
```

entornos virtuales

- inicializar entorno virtual:

```
pyenv virtualenv 3.9.9 ecomerce
```

- activar entorno virtual:

```
pyenv activate ecomerce
```

- listar entornos viruales:

```
pyenv virtualenvs
```

- desactivar:

```
pyenv deactivate
```

- borrar:

```
pyenv virtualenv-delete proyecto2
```

- configurar entorno virtual en directorio concreto:

```
pyenv local ecomerce
```

/via:

- https://roylans.dev/entorno-virtual-en-python-pyenv
- https://www.slayerx.org/2020/05/27/pyenv-para-tener-distintas-versiones-de-python/

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/development:python:entornos-virtuales

Last update: **09/10/2025 14:14**