

# ENTORNOS VIRTUALES

¿Qué son? ¿Para qué sirven? ¿Cómo crearlos?



# VENV



# ¿ QUÉ ES UN ENTORNO VIRTUAL?

Es un espacio aislado que nos permite tener uno o más *espacios aislados* donde instalar distintas versiones de programas, librerías y paquetes.

## Intérprete base: Python versión x.x.



# UN CLARO EJEMPLO



**Virtualenv**

Así evitamos conflictos de librerías y tenemos  
espacios más organizados

# Entornos virtuales con Virtualenv y Virtualenv wrapper

Asumiremos que se cuenta con:

- Python instalado comprobar con:

```
python --version
```

- Se cuenta con pip instalado

- Instalar *virtualenvwrapper*

```
pip install virtualenvwrapper-win
```

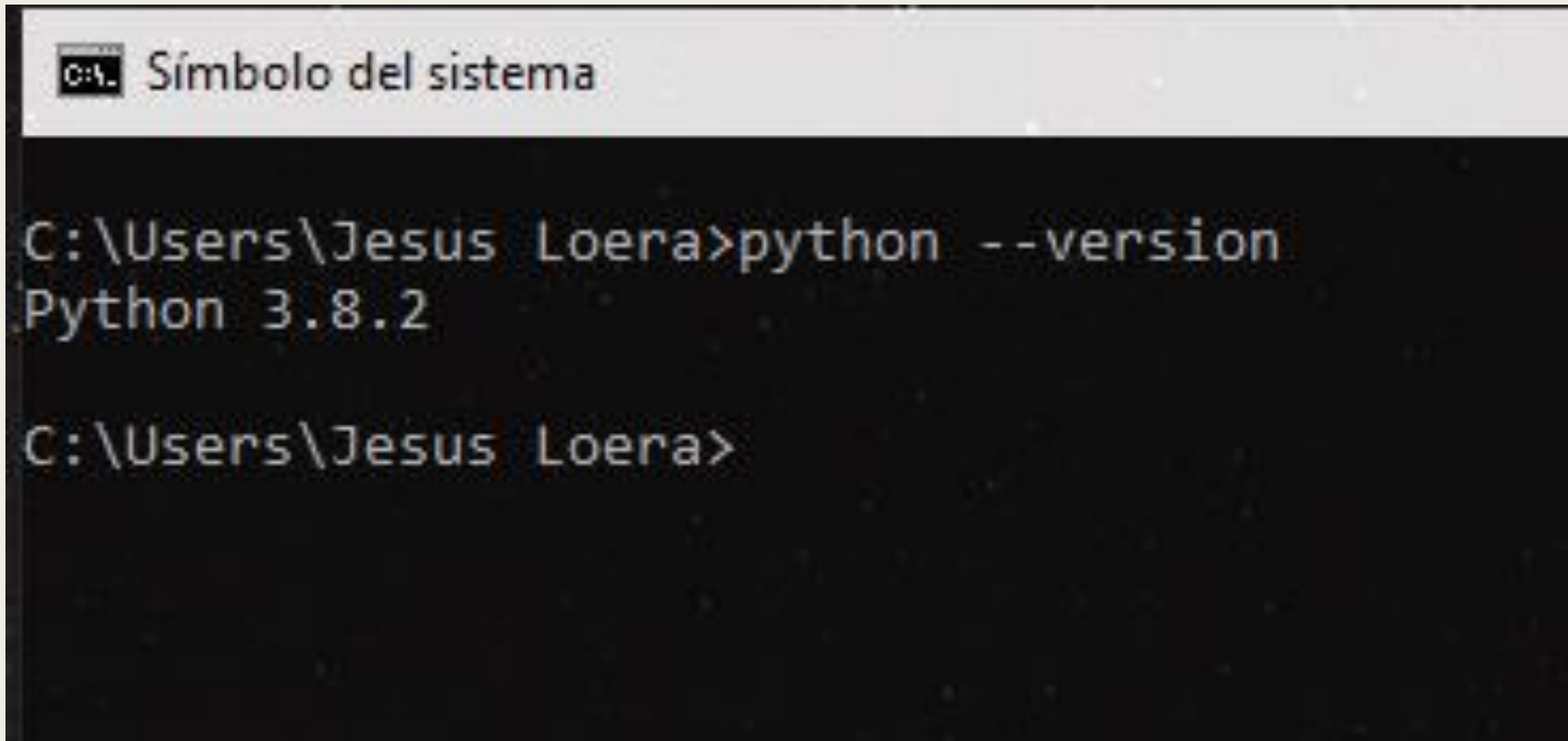


Virtualenv



YA PUEDES CREAR Y MOVERTE DE UN ENTORNO VIRTUAL A OTRO

# Python instalado



```
C:\Users\Jesus Loera>python --version
Python 3.8.2

C:\Users\Jesus Loera>
```

The image shows a Windows Command Prompt window with a light gray title bar that reads "C:\\_ Símbolo del sistema". The main area has a black background with white text. The command prompt shows the user's current directory as "C:\Users\Jesus Loera". The user has entered the command "python --version", and the system has responded with "Python 3.8.2". The prompt is now waiting for the next command.



# Instalar *virtualenvwrapper*

pip install virtualenvwrapper-win

```
C:\Users\Jesus Loera>pip install virtualenvwrapper-win
Requirement already satisfied: virtualenvwrapper-win in c:\python38\lib\site-pac
Requirement already satisfied: virtualenv in c:\python38\lib\site-packages (from
Requirement already satisfied: filelock<4,>=3.0.0 in c:\python38\lib\site-packag
in) (3.0.12)
Requirement already satisfied: distlib<1,>=0.3.1 in c:\python38\lib\site-package
n) (0.3.1)
Requirement already satisfied: appdirs<2,>=1.4.3 in c:\python38\lib\site-package
n) (1.4.4)
Requirement already satisfied: six<2,>=1.9.0 in c:\python38\lib\site-packages (f
1.14.0)

C:\Users\Jesus Loera>_
```

# Crear un entorno virtual

`mkvirtualenv -p pythonX.X.X escriba_aquí_el_nombre_del_entorno`

`pythonX.X.X` : la versión de Python que deseamos

```
C:\Users\Jesus Loera>mkvirtualenv -p python2.7 prueba1
created virtual environment CPython2.7.16.final.0-64 in 11440ms
  creator CPython2Windows(dest=C:\Users\Jesus Loera\Envs\prueba1, clear=False, no_vcs_
ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=
copy, app_data_dir=C:\Users\Jesus Loera\AppData\Local\pypa\virtualenv)
    added seed packages: pip==20.3.3, setuptools==44.1.1, wheel==0.36.2
  activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActi
vator
```

Creamos un entorno virtual con Python 2.7 y se llama prueba1

# ¿Cómo sé si está activo?

Debería aparecer algo así en tu terminal

```
(prueba1) C:\Users\Jesus Loera>
```

# ¿Cómo desactivarlo?

Escribe en la terminal: ***deactivate***

```
(prueba1) C:\Users\Jesus Loera>deactivate
```

Debería aparecer algo así en tu terminal

```
C:\Users\Jesus Loera>
```

# ¿Cómo ver mis entornos virtuales?

Escribe en la terminal: *workon*

```
C:\Users\Jesus Loera>workon
```

Debería aparecer algo así en tu terminal

```
C:\Users\Jesus Loera>workon  
  
Pass a name to activate one of the following virtualenvs:  
=====2  
esto_es_un_entorno_virtual  
prueba1  
  
C:\Users\Jesus Loera>
```

# ¿Cómo volver a activar mis entornos virtuales?

Escribe en la terminal: **workon** aquí\_el\_nombre\_del\_entorno

```
C:\Users\Jesus Loera>workon 2
```

Debería aparecer algo así en tu terminal

```
C:\Users\Jesus Loera>workon 2  
(2) C:\Users\Jesus Loera>
```