

Para instalar Jupyter creando un entorno virtual en Python, sigue estos pasos detalladamente:

1. Instalar `virtualenv` (si no lo tienes):

`virtualenv` es una herramienta para crear entornos virtuales aislados en Python. Si ya lo tienes instalado, puedes saltar este paso. Abre tu terminal o símbolo del sistema y ejecuta:

```
Bash
pip install virtualenv
```

2. Crear el directorio del proyecto (opcional pero recomendado):

Es una buena práctica crear un directorio específico para tu proyecto de Jupyter. Navega hasta donde quieras crear este directorio y ejecuta:

```
Bash
mkdir mi_proyecto_jupyter
cd mi_proyecto_jupyter
```

Reemplaza `mi_proyecto_jupyter` con el nombre que desees para tu proyecto.

3. Crear el entorno virtual:

Dentro del directorio de tu proyecto (o donde prefieras), ejecuta el siguiente comando para crear el entorno virtual:

```
Bash
python -m venv mi_entorno
```

Reemplaza `mi_entorno` con el nombre que quieras darle a tu entorno virtual. `venv` es un módulo estándar en Python 3 para crear entornos virtuales.

4. Activar el entorno virtual:

Debes activar el entorno virtual antes de instalar Jupyter dentro de él. La forma de activarlo depende de tu sistema operativo:

- **Windows:**

```
Bash

mi_entorno\Scripts\activate
```

- **macOS y Linux:**

Bash

```
source mi_entorno/bin/activate
```

Una vez activado, verás el nombre de tu entorno virtual entre paréntesis al principio de la línea de comandos (ej: (mi_entorno) C:\Users\TuUsuario\mi_proyecto_jupyter>).

5. Instalar Jupyter Notebook o JupyterLab:

Con el entorno virtual activado, puedes instalar Jupyter. Tienes dos opciones principales:

- **Jupyter Notebook:** La interfaz clásica de Jupyter.

Bash

```
pip install notebook
```

- **JupyterLab:** La interfaz de próxima generación de Jupyter, que ofrece más funcionalidades y una disposición más flexible.

Bash

```
pip install jupyterlab
```

Puedes instalar ambos si lo deseas.

6. Verificar la instalación (opcional):

Puedes verificar que Jupyter se ha instalado correctamente ejecutando alguno de los siguientes comandos (sin necesidad de crear un nuevo notebook aún):

- Para Jupyter Notebook:

Bash

```
jupyter notebook --version
```

- Para JupyterLab:

Bash

```
jupyter lab --version
```

7. Ejecutar Jupyter:

Para iniciar Jupyter, asegúrate de que tu entorno virtual esté activado y ejecuta uno de los siguientes comandos:

- Para Jupyter Notebook:

Bash

```
jupyter notebook
```

Esto abrirá una nueva pestaña en tu navegador con la interfaz de Jupyter Notebook.

- Para JupyterLab:

Bash

```
jupyter lab
```

Esto abrirá una nueva pestaña en tu navegador con la interfaz de JupyterLab.

8. Desactivar el entorno virtual (cuando termines):

Cuando hayas terminado de trabajar con tu proyecto de Jupyter, puedes desactivar el entorno virtual ejecutando el siguiente comando en la terminal:

Bash

```
deactivate
```

Esto te devolverá a tu entorno de Python global o base.

¿Por qué usar un entorno virtual?

- **Aislamiento:** Cada proyecto tiene sus propias dependencias, y un entorno virtual asegura que las bibliotecas instaladas para un proyecto no interfieran con las de otros proyectos.
- **Reproducibilidad:** Facilita compartir tu proyecto con otros, ya que puedes especificar las versiones exactas de las bibliotecas utilizadas en el archivo `requirements.txt`.
- **Limpieza:** Mantiene tu instalación de Python global más limpia y organizada.

Siguiendo estos pasos, tendrás Jupyter instalado en un entorno virtual aislado, listo para que comiences a trabajar en tus proyectos de Python. ¡Disfruta!

Si deseas instalar Jupyter sin crear un entorno virtual en Python, puedes hacerlo directamente en tu instalación global de Python. Sin embargo, **esta no es la práctica recomendada**, ya que puede llevar a conflictos entre las dependencias de diferentes proyectos en el futuro.

Aun así, si entiendes los riesgos y prefieres hacerlo de esta manera, sigue estos pasos:

1. Asegúrate de tener Python y pip instalados:

Jupyter se instala utilizando `pip`, el gestor de paquetes de Python. Si tienes Python instalado, es muy probable que `pip` también esté instalado. Puedes verificarlo abriendo tu terminal o símbolo del sistema y ejecutando:

```
Bash
python --version
pip --version
```

Si alguno de estos comandos no funciona, necesitas instalar Python y/o `pip` primero. Puedes descargar Python desde el sitio web oficial: <https://www.python.org/downloads/>. Durante la instalación de Python, asegúrate de marcar la opción para agregar Python al PATH y para instalar `pip`.

2. Instalar Jupyter Notebook o JupyterLab:

Abre tu terminal o símbolo del sistema y ejecuta uno de los siguientes comandos, dependiendo de la interfaz de Jupyter que prefieras:

- **Para Jupyter Notebook (la interfaz clásica):**

```
Bash

pip install notebook
```

- **Para JupyterLab (la interfaz de próxima generación):**

```
Bash

pip install jupyterlab
```

Puedes instalar ambos si lo deseas. `pip` descargará e instalará Jupyter y sus dependencias en tu instalación global de Python.

3. Esperar la instalación:

`pip` mostrará el progreso de la descarga e instalación de los paquetes. Una vez que termine, Jupyter estará instalado.

4. Verificar la instalación (opcional):

Puedes verificar la instalación ejecutando los siguientes comandos:

- Para Jupyter Notebook:

Bash

```
jupyter notebook --version
```

- Para JupyterLab:

Bash

```
jupyter lab --version
```

5. Ejecutar Jupyter:

Para iniciar Jupyter, abre tu terminal o símbolo del sistema y ejecuta uno de los siguientes comandos:

- Para Jupyter Notebook:

Bash

```
jupyter notebook
```

Esto abrirá una nueva pestaña en tu navegador con la interfaz de Jupyter Notebook.

- Para JupyterLab:

Bash

```
jupyter lab
```

Esto abrirá una nueva pestaña en tu navegador con la interfaz de JupyterLab.

Consideraciones importantes al instalar globalmente:

- **Conflictos de dependencias:** Si trabajas en múltiples proyectos que requieren diferentes versiones de las mismas bibliotecas, instalarlas globalmente puede llevar a conflictos y hacer que algunos proyectos dejen de funcionar correctamente.

- **Entorno desordenado:** Tu instalación global de Python se llenará con las dependencias de todos los proyectos en los que trabajes, lo que puede dificultar la gestión y el seguimiento de qué bibliotecas están instaladas.
- **Dificultad para la reproducción:** Compartir proyectos con otros puede ser más complicado, ya que no tendrás un archivo `requirements.txt` específico del proyecto que liste las dependencias exactas.

En resumen, aunque es posible instalar Jupyter sin un entorno virtual, se recomienda encarecidamente utilizar entornos virtuales para mantener tus proyectos aislados y evitar posibles problemas de dependencias. Si en el futuro te encuentras con problemas, considera la posibilidad de crear entornos virtuales para tus proyectos.