Guía para utilizar pip en Python desde Visual Studio Code

**¿Qué es pip?**

**pip** es el administrador de paquetes para Python. En el mundo del desarrollo de software, un paquete se refiere a un conjunto de módulos y scripts reutilizables que permiten a los desarrolladores agregar funcionalidad a sus aplicaciones sin tener que escribir el código desde cero. **pip** facilita la tarea de instalar, actualizar y gestionar estos paquetes desde repositorios en línea, como [PyPI (Python Package Index)](https://pypi.org/), así como desde otras fuentes. Gracias a **pip**, los desarrolladores pueden acceder a una vasta biblioteca de soluciones ya creadas, lo que acelera el proceso de desarrollo y fomenta el uso de prácticas estándar en la comunidad.

**Guía para utilizar pip en Python desde Visual Studio Code**

**1. Asegurarse de que la extensión de Python esté instalada:**

Antes de poder trabajar cómodamente con Python y **pip** en VS Code, debes tener instalada la extensión de Python:

* Ve al icono de extensiones en la barra lateral o presiona **Ctrl+Shift+X**.
* Busca "Python" y selecciona la extensión oficial ofrecida por Microsoft.
* Haz clic en "Instalar".

**2. Abre la terminal integrada en Visual Studio Code:**

VS Code tiene una terminal integrada que facilita el trabajo con **pip**:

* Ve al menú **Ver** y selecciona **Terminal**.

**3. Uso básico de pip desde Visual Studio Code:**

Una vez en la terminal, los comandos de **pip** son:

* **Instalar un paquete:** **pip install nombre\_del\_paquete**
* **Desinstalar un paquete:** **pip uninstall nombre\_del\_paquete**
* **Actualizar un paquete:** **pip install --upgrade nombre\_del\_paquete**
* **Listar paquetes instalados:** **pip list**

**4. Uso de archivos requirements.txt:**

* **Crear un requirements.txt:** En la terminal integrada: **pip freeze > requirements.txt**

Esto creará un archivo **requirements.txt** en el directorio de tu proyecto.

* **Instalar paquetes desde un requirements.txt:** **pip install -r requirements.txt**

**5. Entornos virtuales y Visual Studio Code:**

VS Code soporta entornos virtuales, lo cual es ideal para aislar las dependencias del proyecto:

1. En la terminal integrada, crea un entorno virtual:
   * Para Windows: **python -m venv .venv**
   * Para macOS/Linux: **python3 -m venv .venv**
2. Activa el entorno virtual:
   * Para Windows: **.\.venv\Scripts\Activate**
   * Para macOS/Linux: **source .venv/bin/activate**
3. Una vez activado el entorno, VS Code puede preguntar si deseas seleccionar el entorno virtual para el proyecto. Acepta para que se utilice ese entorno.

**6. Consejos adicionales:**

* **IntelliSense y paquetes**: Tras instalar paquetes con **pip**, VS Code debería reconocer automáticamente las nuevas bibliotecas y ofrecer autocompletado y sugerencias gracias a IntelliSense.
* **Seleccionar el intérprete de Python**: Si tienes múltiples versiones de Python o entornos virtuales, puedes seleccionar el intérprete deseado haciendo clic en la esquina inferior izquierda donde aparece la versión de Python o presionando **Ctrl+Shift+P** y buscando "Python: Select Interpreter".
* **Actualiza regularmente**: Mantén actualizado tanto Python, **pip** como Visual Studio Code.