

Primeros pasos con Python en VS Code

En este tutorial, usará Python 3 para crear la aplicación Python "Hello World" más simple en Visual Studio Code. Al usar la extensión de Python, convierte VS Code en un excelente IDE de Python liviano (que puede encontrar una alternativa productiva a PyCharm).

Este tutorial le presenta VS Code como un entorno de Python, principalmente cómo editar, ejecutar y depurar código a través de las siguientes tareas:

- Escribir, ejecutar y depurar una aplicación Python "Hello World"
- Aprenda a instalar paquetes creando entornos virtuales de Python
- Escriba un script de Python simple para trazar figuras dentro de VS Code

Este tutorial no pretende enseñarte Python en sí. Una vez que esté familiarizado con los conceptos básicos de VS Code, puede seguir cualquiera de los [tutoriales de programación en python.org](https://www.python.org/tutorials/) dentro del contexto de VS Code para obtener una introducción al lenguaje.

Si tiene algún problema, puede buscar respuestas o hacer una pregunta en la [extensión de Python Discusiones Preguntas y respuestas](#).

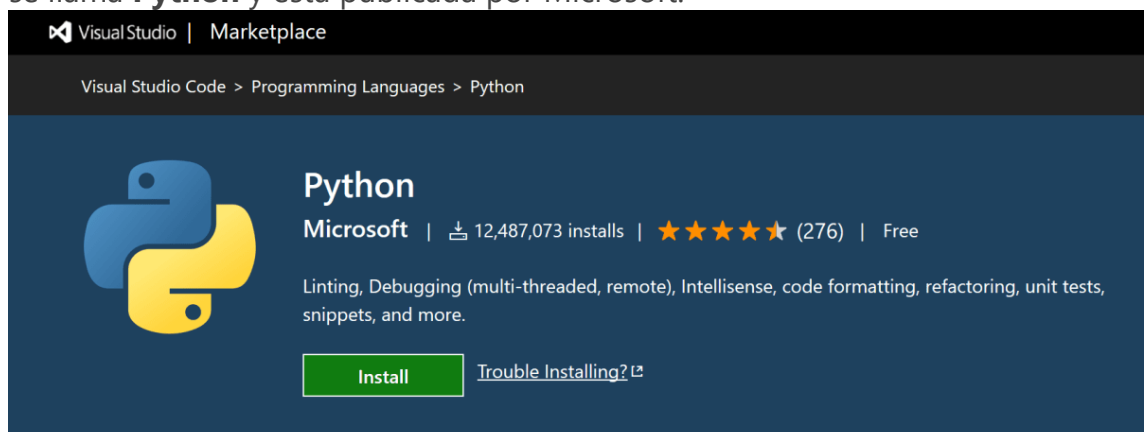
requisitos previos

Para completar con éxito este tutorial, primero debe configurar su entorno de desarrollo de Python. Específicamente, este tutorial requiere:

- Pitón 3
- Aplicación de código VS
- Extensión VS Code Python

Instale Visual Studio Code y la extensión de Python

1. Si aún no lo ha hecho, instale [VS Code](#).
2. A continuación, instale la [extensión de Python para VS Code](#) desde Visual Studio Marketplace. Para obtener detalles adicionales sobre la instalación de extensiones, consulte [Extension Marketplace](#). La extensión de Python se llama **Python** y está publicada por Microsoft.



Instalar un intérprete de Python

Junto con la extensión de Python, debe instalar un intérprete de Python. El intérprete que utilice depende de sus necesidades específicas, pero a continuación se proporciona alguna orientación.

ventanas

Instale [Python desde python.org](https://python.org) . Por lo general, puede usar el botón **Descargar Python** que aparece primero en la página para descargar la última versión.

Nota : si no tiene acceso de administrador, una opción adicional para instalar Python en Windows es usar Microsoft Store. Microsoft Store proporciona instalaciones de [versiones compatibles de Python](#) .

Para obtener información adicional sobre el uso de Python en Windows, consulte [Uso de Python en Windows en Python.org](#)

Mac OS

No se admite la instalación del sistema de Python en macOS. En su lugar, se recomienda un sistema de gestión de paquetes como [Homebrew](#) . Para instalar Python usando Homebrew en macOS, use `brew install python3` el indicador de Terminal.

Nota En macOS, asegúrese de que la ubicación de su instalación de VS Code esté incluida en su variable de entorno PATH. Consulte [estas instrucciones de configuración](#) para obtener más información.

linux

La instalación integrada de Python 3 en Linux funciona bien, pero para instalar otros paquetes de Python debe instalar `pip` con [get-pip.py](#) .

Otras opciones

- **Ciencia de datos** : si su propósito principal para usar Python es la ciencia de datos, entonces podría considerar una descarga de [Anaconda](#) . Anaconda proporciona no solo un intérprete de Python, sino muchas bibliotecas y herramientas útiles para la ciencia de datos.
- **Subsistema de Windows para Linux** : si está trabajando en Windows y quiere un entorno Linux para trabajar con Python, el [Subsistema de Windows para Linux](#) (WSL) es una opción para usted. Si elige esta opción, también querrá instalar la [extensión WSL](#) . Para obtener más información sobre el uso de WSL con VS Code, consulte [Desarrollo remoto de VS Code](#) o pruebe el [tutorial Trabajar en WSL](#) , que lo guiará a través de la configuración de WSL, la instalación de Python y la creación de una aplicación Hello World que se ejecute en WSL.

Verificar la instalación de Python

Para verificar que instaló Python con éxito en su máquina, ejecute uno de los siguientes comandos (según su sistema operativo):

- Linux/macOS: abra una ventana de terminal y escriba el siguiente comando:

```
python3 --version
```

- Windows: abra un símbolo del sistema y ejecute el siguiente comando:

```
py -3 --version
```

Si la instalación fue exitosa, la ventana de salida debería mostrar la versión de Python que instaló.

Nota Puede usar el `py -0` comando en el terminal integrado de VS Code para ver las versiones de python instaladas en su máquina. El intérprete predeterminado se identifica con un asterisco (*).

Inicie VS Code en una carpeta de espacio de trabajo

Al iniciar VS Code en una carpeta, esa carpeta se convierte en su "área de trabajo". VS Code almacena configuraciones que son específicas de ese espacio de trabajo en `.vscode/settings.json`, que son independientes de las configuraciones de usuario que se almacenan globalmente.

Usando un símbolo del sistema o terminal, cree una carpeta vacía llamada "hola", navegue hasta ella y abra VS Code (`code`) en esa carpeta (`.`) ingresando los siguientes comandos:

```
mkdir hello  
  
cd hello  
  
code .
```

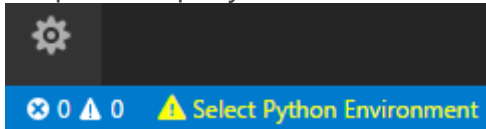
Nota : si está usando una distribución de Anaconda, asegúrese de usar un símbolo del sistema de Anaconda.

Como alternativa, puede ejecutar VS Code a través de la interfaz de usuario del sistema operativo y luego usar **Archivo > Abrir carpeta** para abrir la carpeta del proyecto.

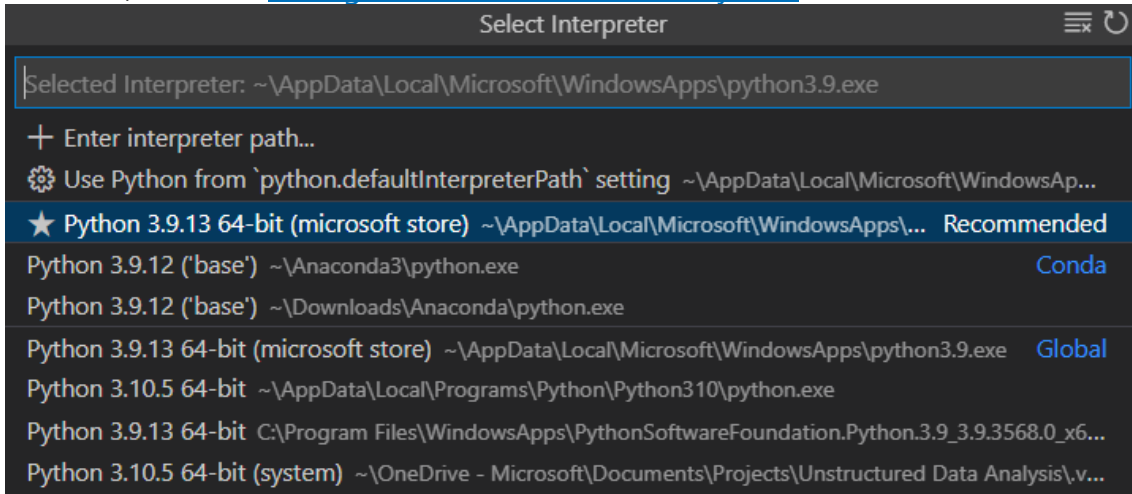
Seleccione un intérprete de Python

Python es un lenguaje interpretado. Por lo tanto, para ejecutar el código de Python y obtener Python IntelliSense, debe decirle a VS Code qué intérprete usar. Desde VS Code, seleccione un intérprete de Python 3 abriendo la **Paleta de comandos** (`Ctrl+Shift+P`), comience a escribir el comando **Python: Select Interpreter** para buscar, luego seleccione el comando. También puede usar la

opción **Seleccionar entorno de Python** en la barra de estado si está disponible (es posible que ya muestre un intérprete seleccionado también):



El comando presenta una lista de intérpretes disponibles que VS Code puede encontrar automáticamente, incluidos los entornos virtuales. Si no ve el intérprete deseado, consulte [Configuración de entornos de Python](#).



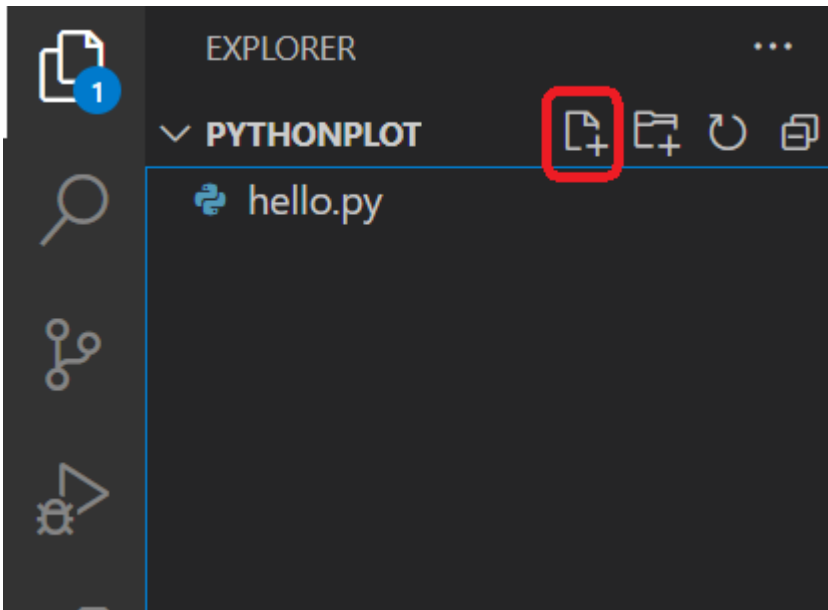
Nota : cuando se utiliza una distribución de Anaconda, el intérprete correcto debe tener el sufijo `('base':conda)`, por ejemplo `Python 3.7.3 64-bit ('base':conda)`.

Al seleccionar un intérprete, se establece qué intérprete utilizará la extensión de Python para ese espacio de trabajo.

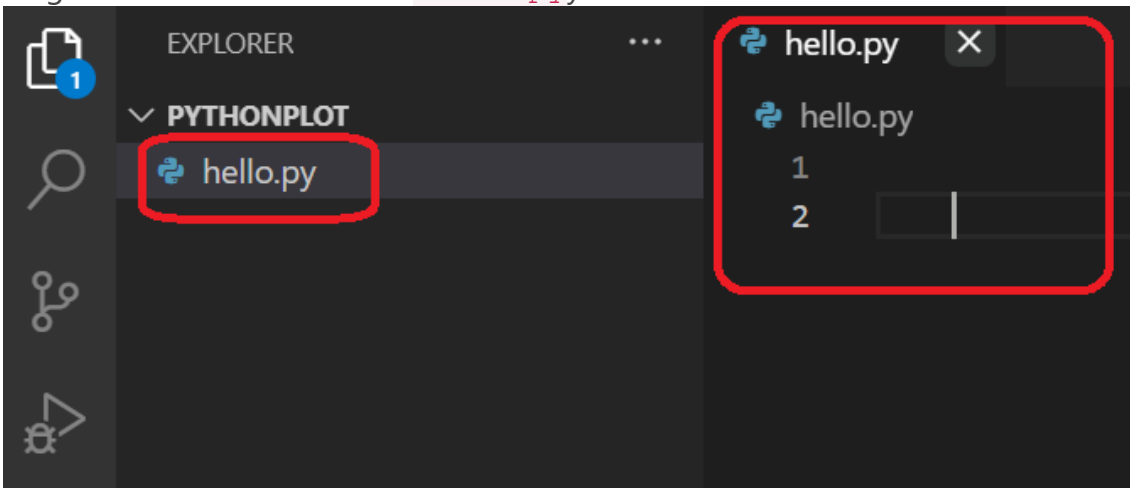
Nota : si selecciona un intérprete sin una carpeta de espacio de trabajo abierta, VS Code se establece `python.defaultInterpreterPath` en el alcance del usuario, lo que establece el intérprete predeterminado para VS Code en general. La configuración del usuario garantiza que siempre tenga un intérprete predeterminado para los proyectos de Python. La configuración del espacio de trabajo le permite anular la configuración del usuario.

Cree un archivo de código fuente de Python Hello World

En la barra de herramientas del Explorador de archivos, seleccione el botón **Nuevo archivo** en la `hello` carpeta:



Asigne un nombre al archivo `hello.py` y se abrirá automáticamente en el editor:



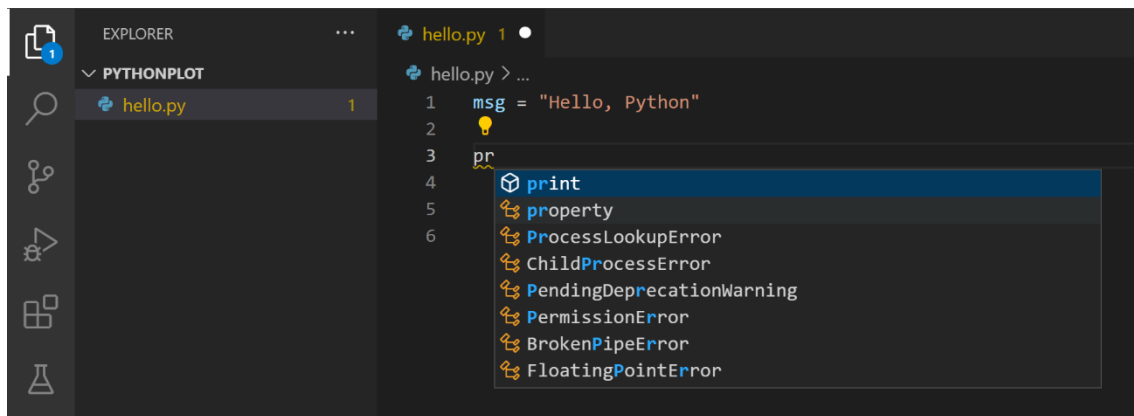
Al usar la `.py` extensión de archivo, le indica a VS Code que interprete este archivo como un programa de Python, de modo que evalúe el contenido con la extensión de Python y el intérprete seleccionado.

Nota : la barra de herramientas del Explorador de archivos también le permite crear carpetas dentro de su espacio de trabajo para organizar mejor su código. Puede usar el botón **Nueva carpeta** para crear rápidamente una carpeta. Ahora que tiene un archivo de código en su espacio de trabajo, ingrese el siguiente código fuente en `hello.py`:

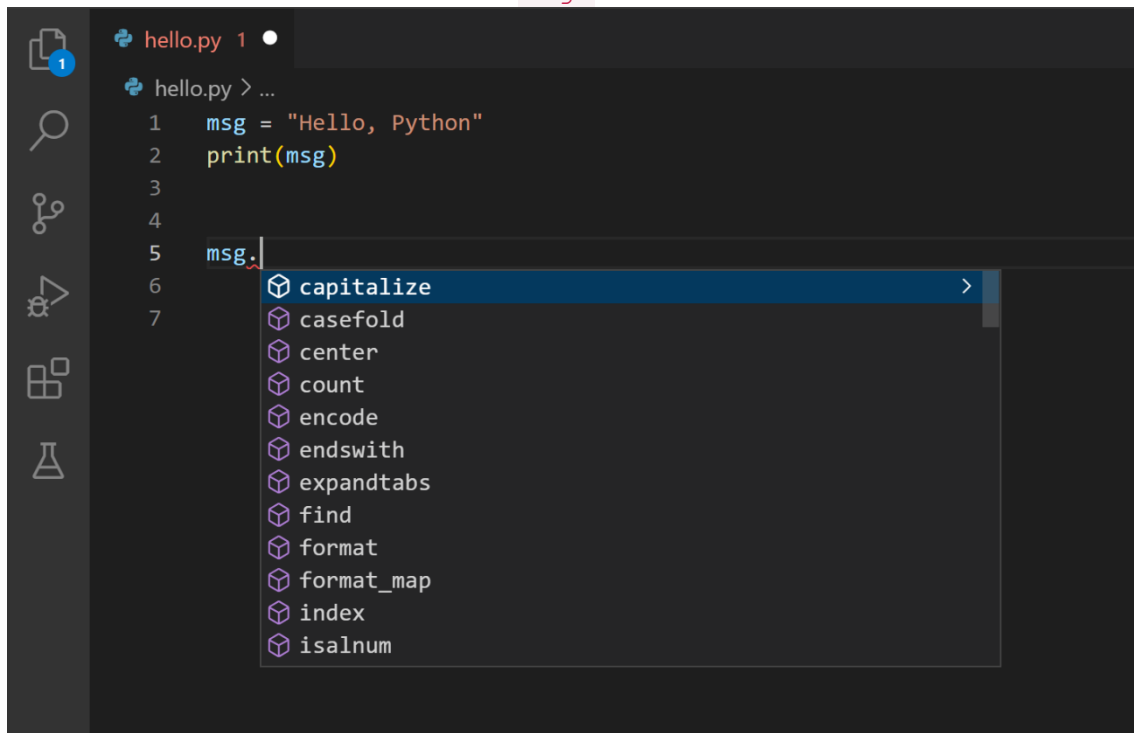
```
msg = "Hello World"

print(msg)
```

Cuando comience a escribir `print`, observe cómo [IntelliSense](#) presenta las opciones de finalización automática.



IntelliSense y las funciones de finalización automática funcionan para los módulos estándar de Python, así como para otros paquetes que haya instalado en el entorno del intérprete de Python seleccionado. También proporciona terminaciones para métodos disponibles en tipos de objetos. Por ejemplo, debido a que la `msg` variable contiene una cadena, IntelliSense proporciona métodos de cadena cuando escribe `msg.:`

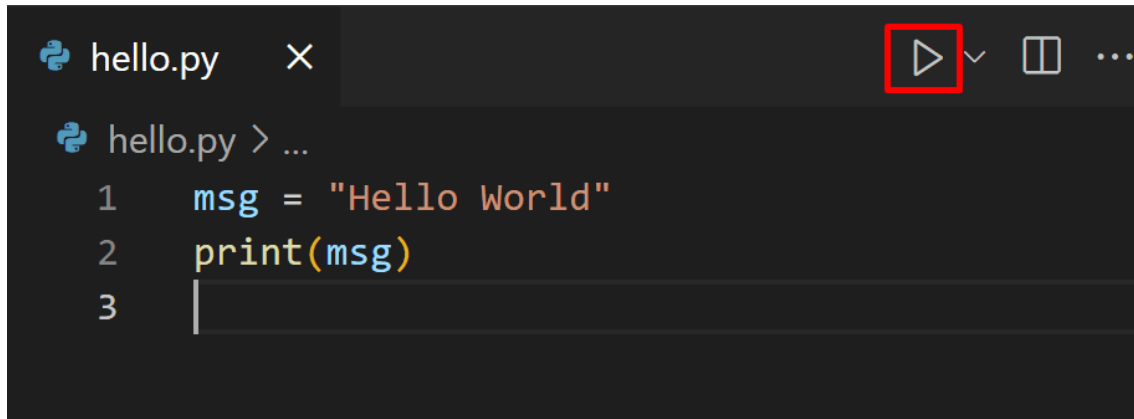


Siéntase libre de experimentar con IntelliSense un poco más, pero luego revierta sus cambios para que solo tenga la `msg` variable y la `print` llamada, y guarde el archivo ([Ctrl+S](#)).

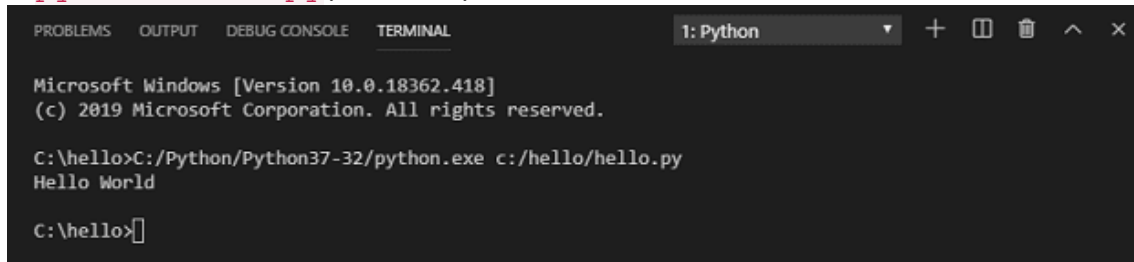
Para obtener detalles completos sobre la edición, el formato y la refactorización, consulte [Edición de código](#) . La extensión de Python también tiene soporte completo para [Linting](#) .

Corre hola mundo

Es fácil de ejecutar `hello.py` con Python. Simplemente haga clic en el botón de reproducción **Run Python File in Terminal** en la parte superior derecha del editor.

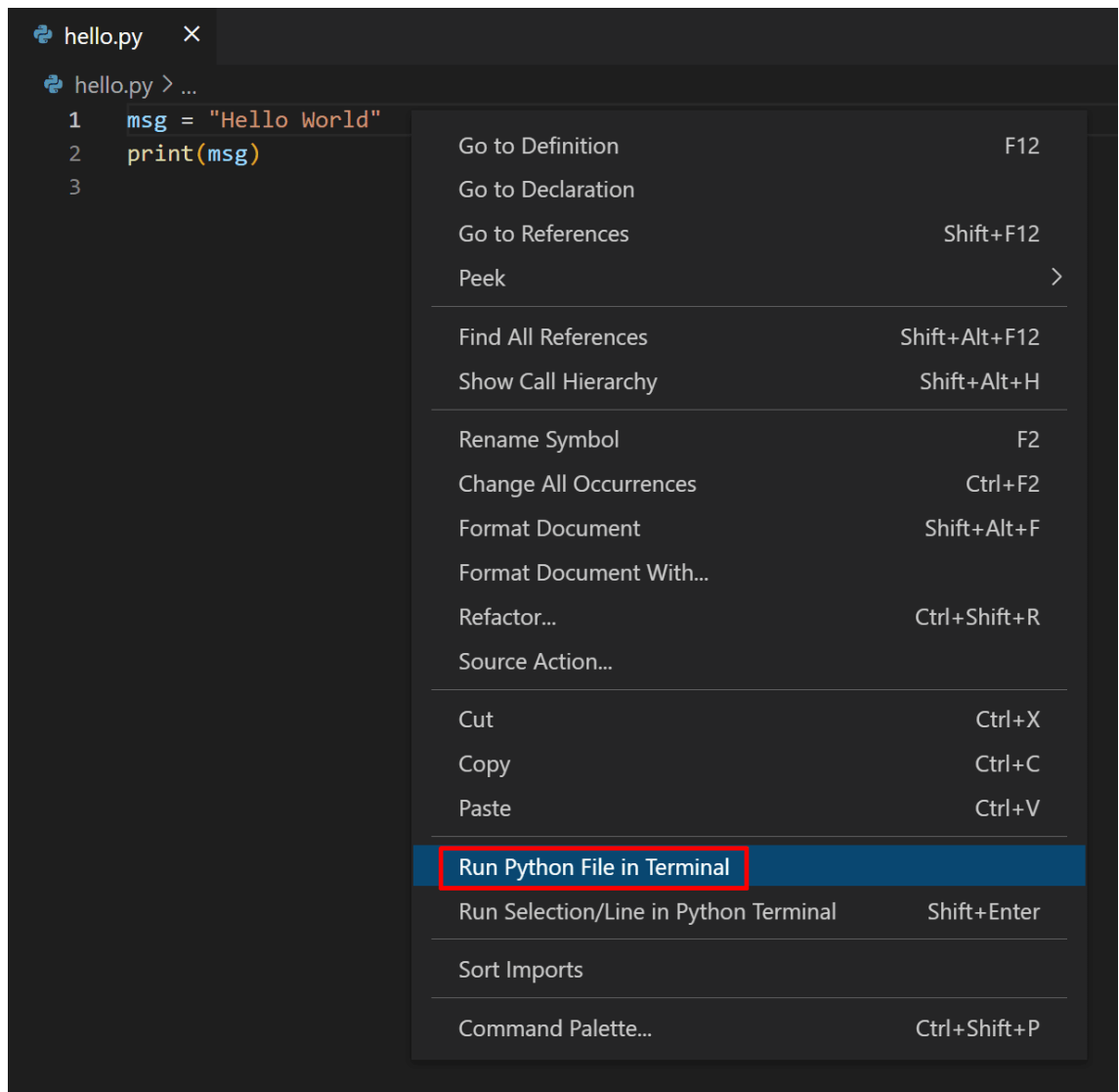


El botón abre un panel de terminal en el que su intérprete de Python se activa automáticamente y luego se ejecuta `python3 hello.py` (macOS/Linux) o `python hello.py` (Windows):



Hay otras tres formas de ejecutar código Python dentro de VS Code:

- Haga clic con el botón derecho en cualquier parte de la ventana del editor y seleccione **Ejecutar archivo de Python en la terminal** (que guarda el archivo automáticamente):



- Seleccione una o más líneas, luego presione **Shift+Enter** o haga clic con el botón derecho y seleccione **Run Selection/Line in Python Terminal**. Este comando es conveniente para probar solo una parte de un archivo.
- Desde la Paleta de comandos (**Ctrl+Shift+P**), seleccione el comando **Python: Iniciar REPL para abrir un terminal REPL para el intérprete de Python seleccionado actualmente**. En el REPL, puede ingresar y ejecutar líneas de código una a la vez.

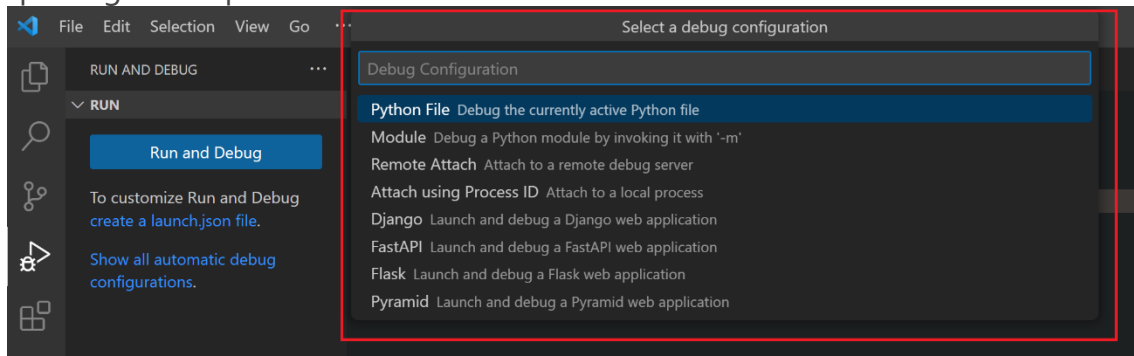
Configurar y ejecutar el depurador

Ahora intentemos depurar nuestro sencillo programa Hello World.

Primero, establezca un punto de interrupción en la línea 2 `hello.py` colocando el cursor en la `print` llamada y presionando **F9**. Alternativamente, simplemente haga clic en el margen izquierdo del editor, junto a los números de línea. Cuando establece un punto de interrupción, aparece un círculo rojo en el medianil.


```
hello.py x
hello.py > ...
1 msg = "Hello World"
2 print(msg)
```

A continuación, para inicializar el depurador, presione **F5** . Dado que es la primera vez que depura este archivo, se abrirá un menú de configuración desde la Paleta de comandos que le permitirá seleccionar el tipo de configuración de depuración que le gustaría para el archivo abierto.



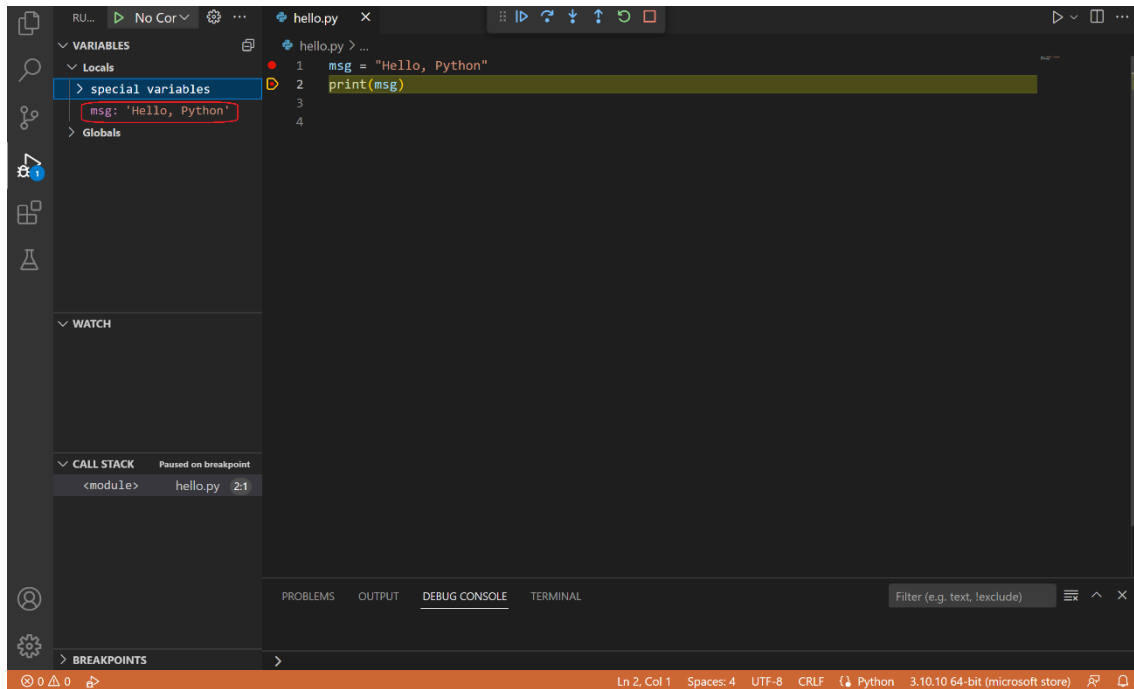
Nota : VS Code usa archivos JSON para todas sus diversas configuraciones; `launch.json` es el nombre estándar para un archivo que contiene configuraciones de depuración.

Estas diferentes configuraciones se explican completamente en [Configuraciones de depuración](#) ; por ahora, simplemente seleccione **Python File** , que es la configuración que ejecuta el archivo actual que se muestra en el editor utilizando el intérprete de Python seleccionado actualmente.

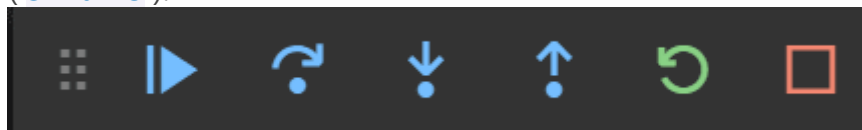
También puede iniciar el depurador haciendo clic en la flecha hacia abajo junto al botón Ejecutar en el editor y seleccionando **Depurar archivo de Python en Terminal** .



El depurador se detendrá en la primera línea del punto de interrupción del archivo. La línea actual se indica con una flecha amarilla en el margen izquierdo. Si examina la ventana Variables **locales** en este punto, verá que ahora `msg` la variable definida aparece en el panel **Local** .



Aparece una barra de herramientas de depuración en la parte superior con los siguientes comandos de izquierda a derecha: continuar (**F5**), pasar por encima (**F10**), entrar (**F11**), salir (**Mayús+F11**), reiniciar (**Ctrl+Mayús+F5**), y detener (**Shift+F5**).



La barra de estado también cambia de color (naranja en muchos temas) para indicar que está en modo de depuración. La **consola de depuración de Python** también aparece automáticamente en el panel inferior derecho para mostrar los comandos que se están ejecutando, junto con la salida del programa. Para continuar ejecutando el programa, seleccione el comando continuar en la barra de herramientas de depuración (**F5**). El depurador ejecuta el programa hasta el final.

Sugerencia La información de depuración también se puede ver al pasar el mouse sobre el código, como las variables. En el caso de `msg`, al pasar el cursor sobre la variable se mostrará la cadena `Hello world` en un cuadro encima de la variable.

También puede trabajar con variables en la **Consola de depuración** (si no la ve, seleccione **Consola de depuración** en el área inferior derecha de VS Code, o selecciónela en el menú ...). Luego intente ingresar las siguientes líneas, una por uno, en el indicador `>` en la parte inferior de la consola:

```
msg
msg.capitalize()
msg.split()
```



Seleccione el botón azul **Continuar** en la barra de herramientas nuevamente (o presione F5) para ejecutar el programa hasta su finalización. "Hello World" aparece en la **Consola de depuración de Python** si vuelve a cambiar a ella, y VS Code sale del modo de depuración una vez que se completa el programa.

Si reinicia el depurador, el depurador vuelve a detenerse en el primer punto de interrupción.

Para detener la ejecución de un programa antes de que finalice, use el botón de parada cuadrado rojo en la barra de herramientas de depuración ([Shift+F5](#)), o use el comando de menú **Ejecutar > Detener la depuración** .

Para obtener detalles completos, consulte [Configuraciones de depuración](#) , que incluye notas sobre cómo usar un intérprete de Python específico para la depuración.

Sugerencia: use puntos de registro en lugar de declaraciones impresas : los desarrolladores a menudo ensucian el código fuente con `print` declaraciones para inspeccionar rápidamente las variables sin tener que pasar necesariamente por cada línea de código en un depurador. En VS Code, puede usar **Logpoints** en su lugar . Un punto de registro es como un punto de interrupción, excepto que registra un mensaje en la consola y no detiene el programa. Para obtener más información, consulte [Puntos de registro](#) en el artículo principal de depuración de VS Code.

Instalar y usar paquetes

Veamos ahora un ejemplo que es un poco más interesante. En Python, los paquetes son la forma de obtener cualquier cantidad de bibliotecas de códigos útiles, generalmente de [PyPI](#) . Para este ejemplo, usa los paquetes `matplotlib` `numpy` para crear un diagrama gráfico como se hace comúnmente con la ciencia de datos. (Tenga en cuenta que `matplotlib` no puede mostrar gráficos cuando se ejecuta en el [subsistema de Windows para Linux](#) , ya que carece de la compatibilidad necesaria con la interfaz de usuario).

Regrese a la vista del **Explorador** (el ícono superior en el lado izquierdo, que muestra los archivos), cree un nuevo archivo llamado `standardplot.py` pegue el siguiente código fuente:

```
import matplotlib.pyplot as plt

import numpy as np
```

```
x = np.linspace(0, 20, 100) # Create a list of evenly-spaced numbers over the range

plt.plot(x, np.sin(x))      # Plot the sine of each x point

plt.show()                  # Display the plot
```

Sugerencia: si ingresa el código anterior a mano, es posible que los autocompletados cambien los nombres después de las palabras clave cuando presiona [Intro](#) al final de una línea. Para evitar esto, escriba un espacio, luego [Enter](#).

A continuación, intente ejecutar el archivo en el depurador usando la configuración "Python: archivo actual" como se describe en la última sección.

A menos que esté utilizando una distribución de Anaconda o haya instalado previamente el `matplotlib` paquete, debería ver el mensaje "**ModuleNotFoundError: No module named 'matplotlib'**". Dicho mensaje indica que el paquete requerido no está disponible en su sistema.

Para instalar el `matplotlib` paquete (que también se instala `numpy` como una dependencia), detenga el depurador y use la Paleta de comandos para ejecutar **Terminal: Crear nueva terminal** ([Ctrl+Shift+`](#)). Este comando abre un símbolo del sistema para su intérprete seleccionado.

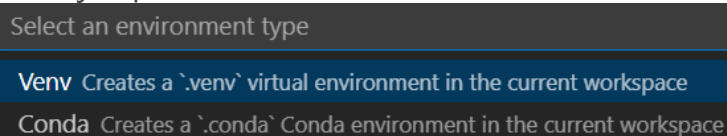
Una buena práctica entre los desarrolladores de Python es evitar instalar paquetes en un entorno de interpretación global. En su lugar, utiliza un proyecto específico `virtual environment` que contiene una copia de un intérprete global. Una vez que activa ese entorno, cualquier paquete que instale se aísla de otros entornos. Tal aislamiento reduce muchas complicaciones que pueden surgir de versiones de paquetes en conflicto. Para crear un *entorno virtual* e instalar los paquetes necesarios, ingrese los siguientes comandos según corresponda a su sistema operativo:

Nota: Para obtener información adicional sobre entornos virtuales, consulte [Entornos](#).

1. Cree un entorno virtual con el comando Crear entorno

Desde VS Code, puede crear entornos no globales, usando Venv o Anaconda, abriendo la Paleta de comandos ([Ctrl+Shift+P](#)), comience a escribir el comando **Python: Create Environment** para buscar y luego seleccione el comando. También puede activar el comando **Python: Crear entorno** a través de la página Primeros pasos con Python.

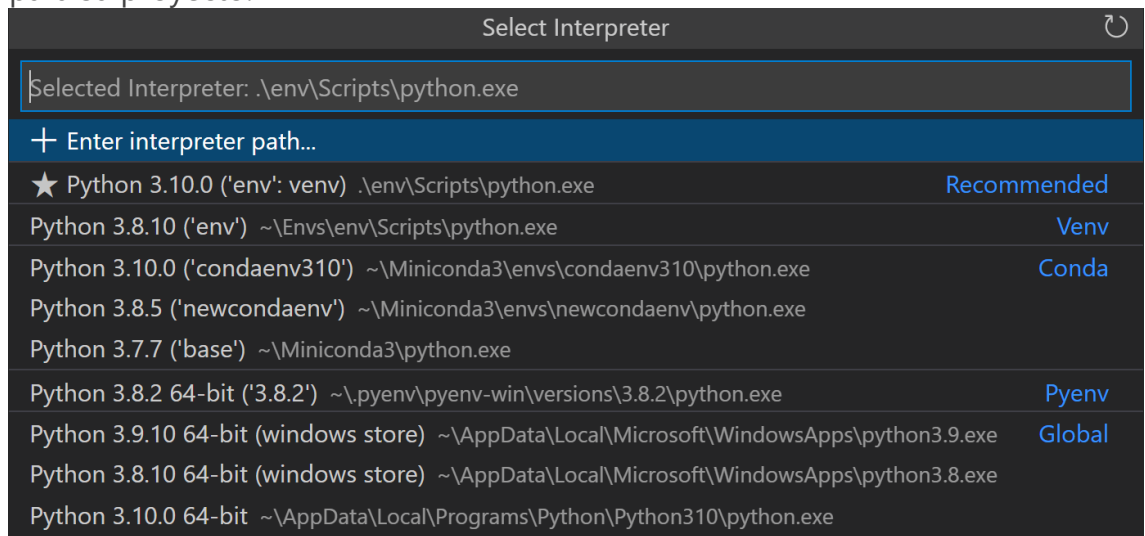
El comando presenta una lista de tipos de entorno, Venv o Conda. Para este ejemplo, seleccione **Venv**.



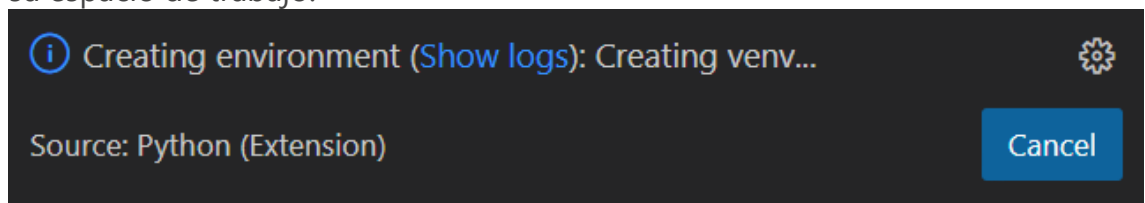
Select an environment type

- Venv Creates a `.venv` virtual environment in the current workspace
- Conda Creates a `.conda` Conda environment in the current workspace

Luego, el comando presenta una lista de intérpretes que se pueden usar para su proyecto.



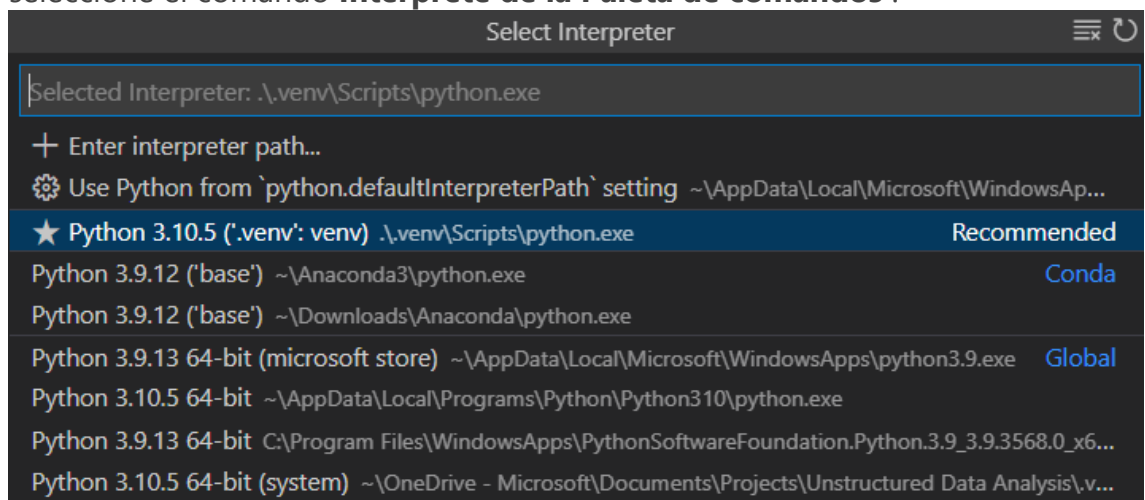
Después de seleccionar el intérprete deseado, una notificación mostrará el progreso de la creación del entorno y la carpeta del entorno aparecerá en su espacio de trabajo.



El comando también instalará los paquetes necesarios descritos en un archivo de requisitos/dependencias, como `requirements.txt`, `pyproject.toml` o `environment.yml`, ubicado en la carpeta del proyecto.

Nota : si desea crear un entorno manualmente o si se produce un error en el proceso de creación del entorno, visite la página [Entornos](#).

2. Asegúrese de que su nuevo entorno esté seleccionado mediante Python: seleccione el comando **Intérprete de la Paleta de comandos**.



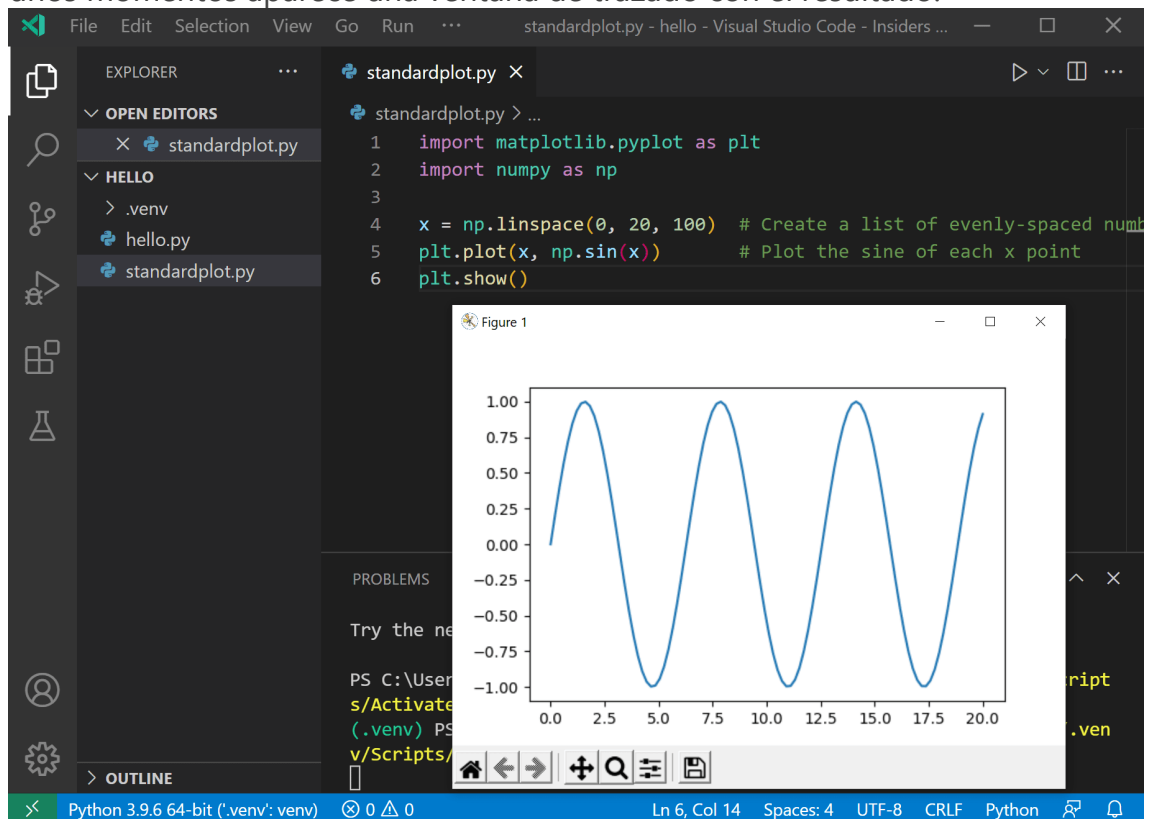
3. Instalar los paquetes

```

4. # Don't use with Anaconda distributions because they include matplotlib already.
5.
6. # macOS
7. python3 -m pip install matplotlib
8.
9. # Windows (may require elevation)
10. py -m pip install matplotlib
11.
12. # Linux (Debian)
13. apt-get install python3-tk
14. python3 -m pip install matplotlib

```

15. Vuelva a ejecutar el programa ahora (con o sin el depurador) y después de unos momentos aparece una ventana de trazado con el resultado:



16. Una vez que haya terminado, escriba `deactivate` en la ventana de la terminal para desactivar el entorno virtual.