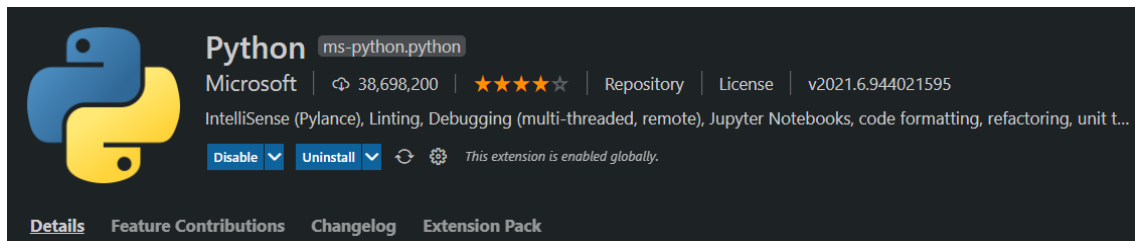


GETTING STARTED:

Using Python Inside VS Code

1. Install the [Python extension for VS Code](#) from the Visual Studio Marketplace. The Python extension is named **Python** and it's published by Microsoft.

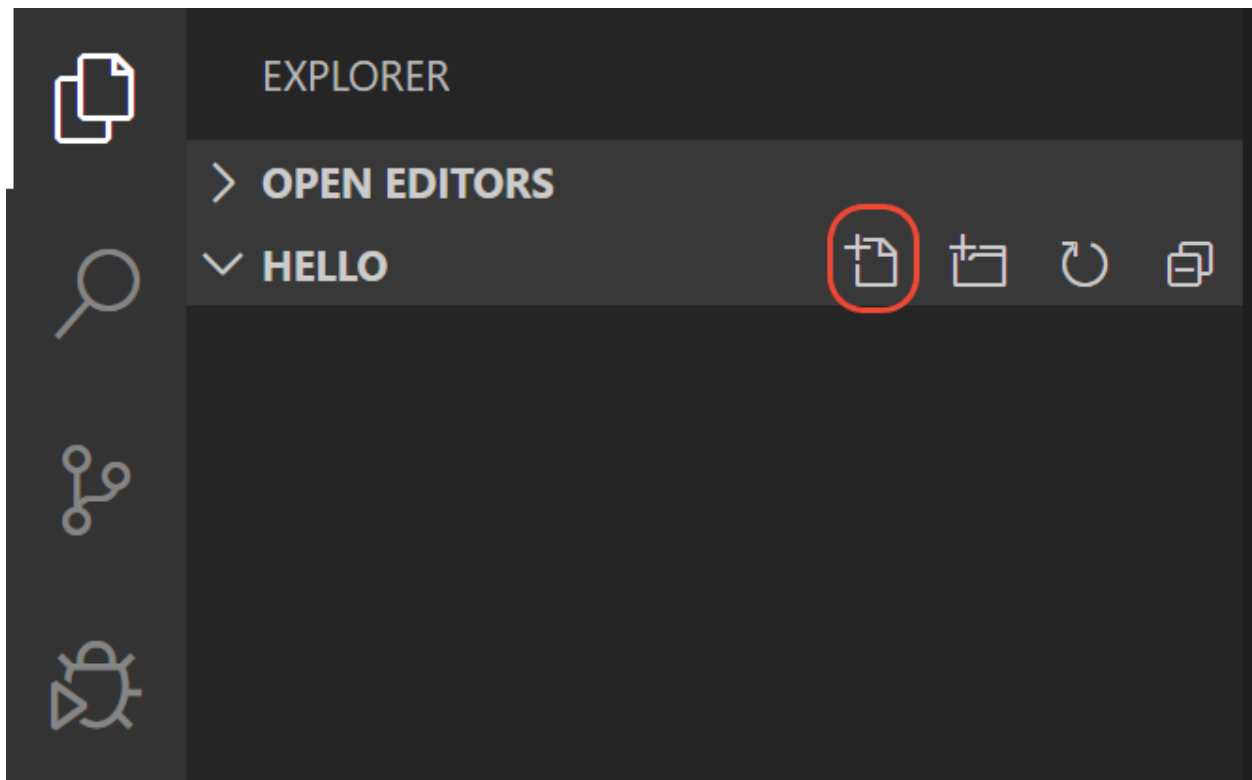


2. To verify that you've installed Python successfully on your machine, run one of the following commands (depending on your operating system):
 - a. Linux/macOS: open a Terminal Window and type the following command:
 - i. `python3 --version`
 - b. Windows: open a command prompt and run the following command:
 - i. `py -3 --version`
 - c. If the installation was successful, the output window should show the version of Python that you installed.

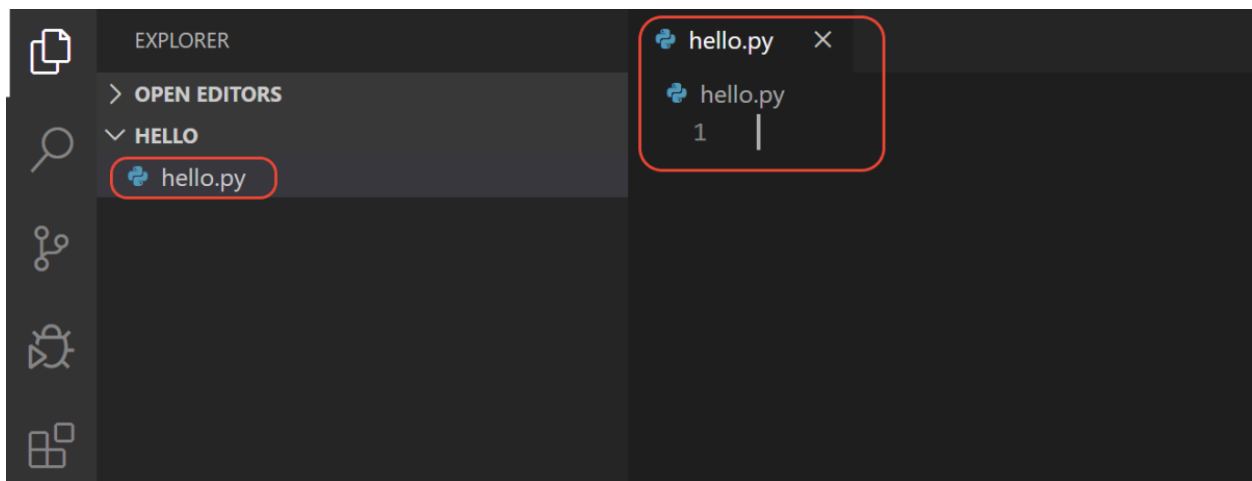
CREATING, RUNNING, and DEBUGGING

Let's create a new folder to hold our project and call it "HELLO."

From the File Explorer toolbar, select the **New File** button on the `hello` folder:



Name the file `hello.py`, and it automatically opens in the editor:



By using the `.py` file extension, you tell VS Code to interpret this file as a Python program, so that it evaluates the contents with the Python extension and the selected interpreter.

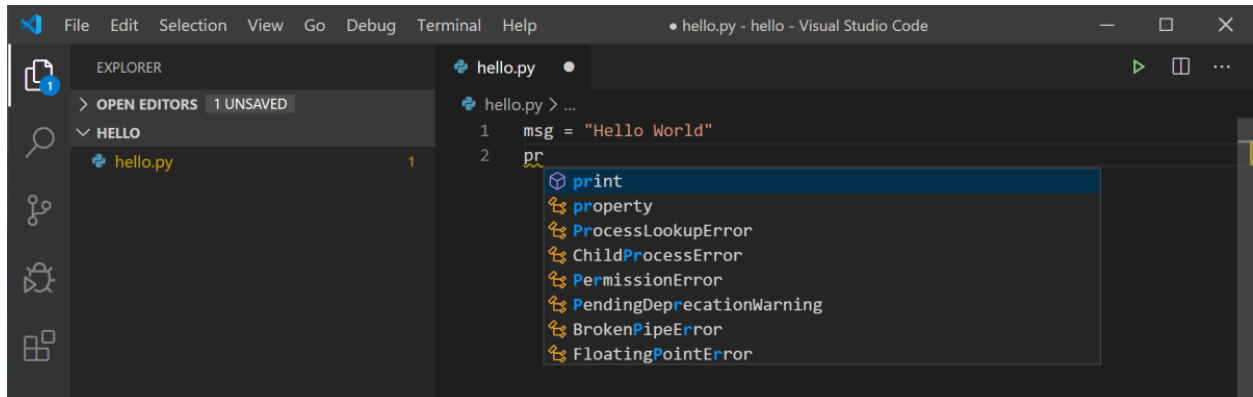
Note: The File Explorer toolbar also allows you to create folders within your workspace to better organize your code. You can use the **New folder** button to quickly create a folder.

Now that you have a code file in your Workspace, enter the following source code in `hello.py`:

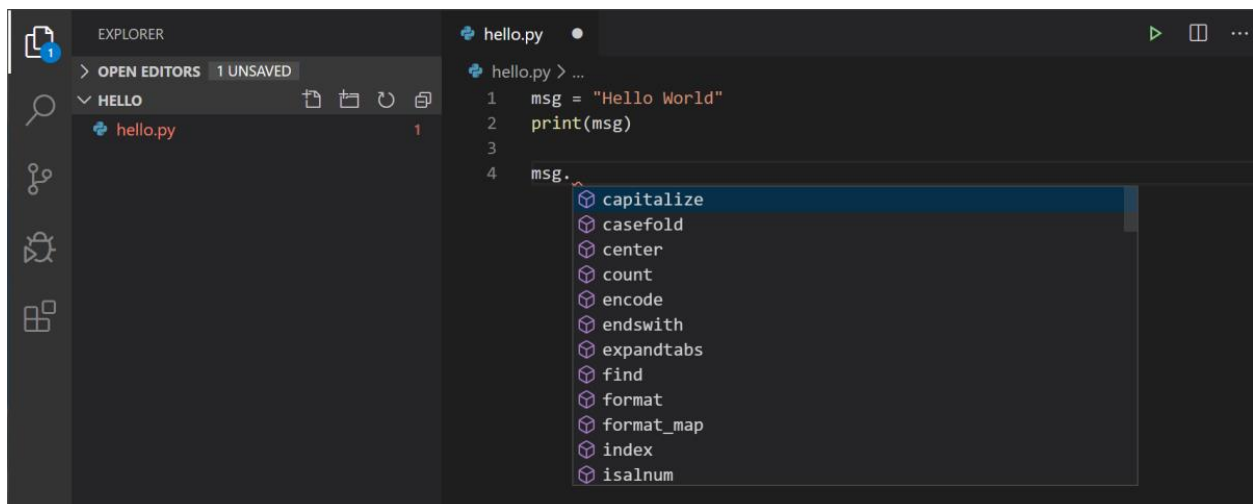
```
msg = "Hello World"

print(msg)
```

When you start typing `print`, notice how **IntelliSense** presents auto-completion options.



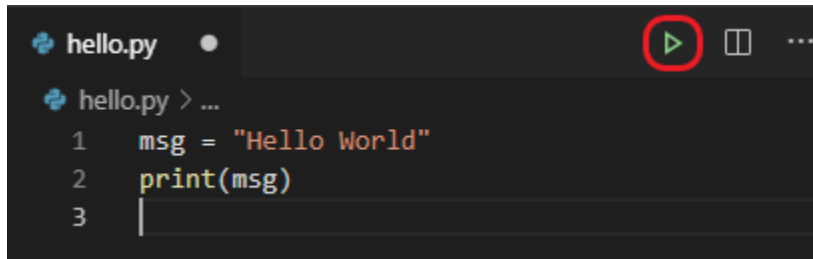
IntelliSense and auto-completions work for standard Python modules as well as other packages you've installed into the environment of the selected Python interpreter. It also provides completions for methods available on object types. For example, because the `msg` variable contains a string, IntelliSense provides string methods when you type `msg.`:



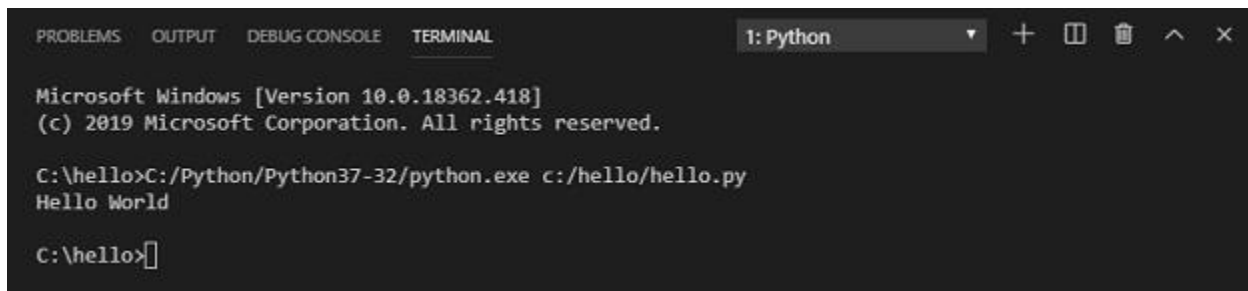
Feel free to experiment with IntelliSense some more, but then revert your changes so you have only the `msg` variable and the `print` call, and save the file.

Run Hello World#

It's simple to run `hello.py` with Python. Just click the **Run Python File in Terminal** play button in the top-right side of the editor.

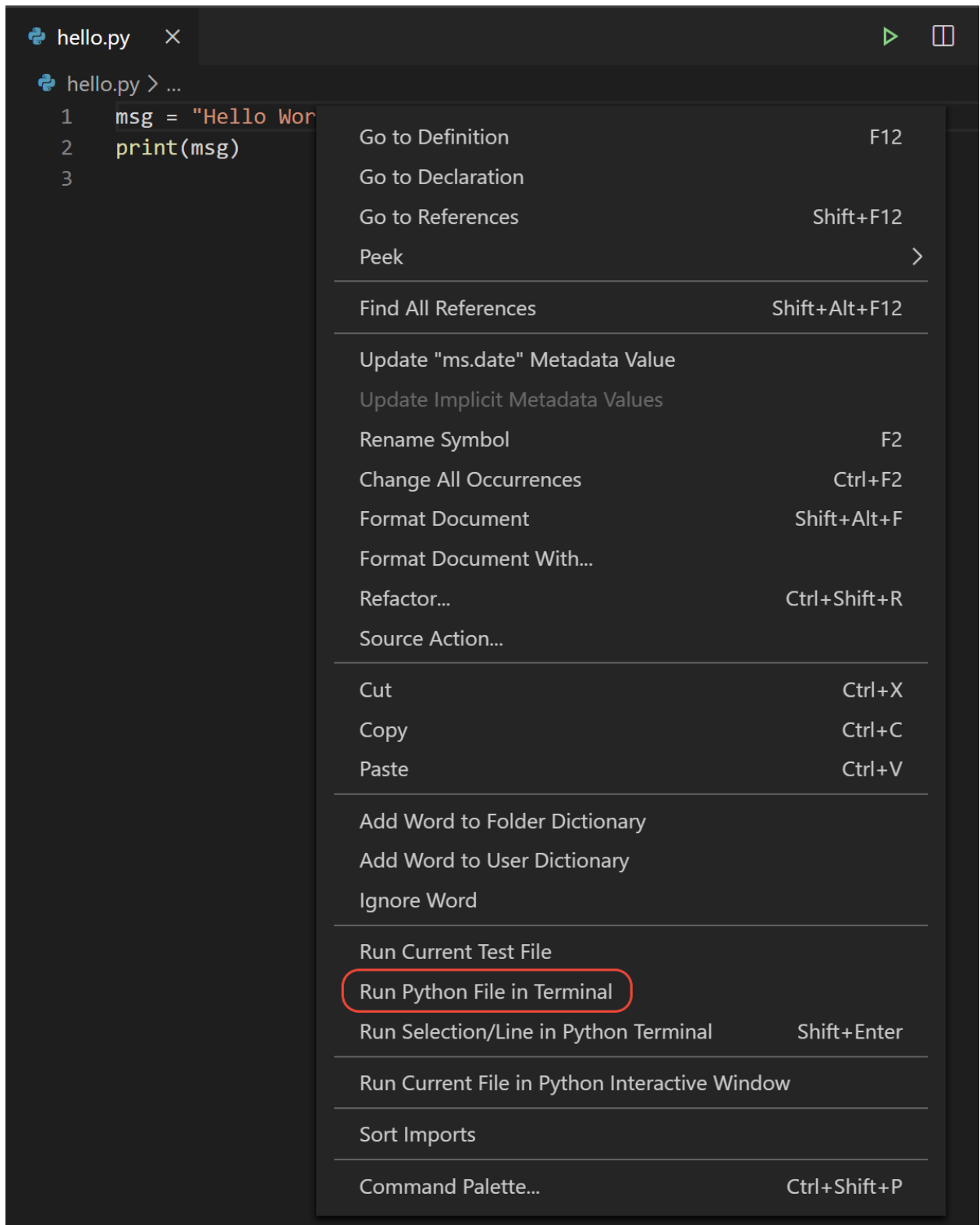


The button opens a terminal panel in which your Python interpreter is automatically activated, then runs `python3 hello.py` (macOS/Linux) or `python hello.py` (Windows):



There are three other ways you can run Python code within VS Code:

- Right-click anywhere in the editor window and select **Run Python File in Terminal** (which saves the file automatically):



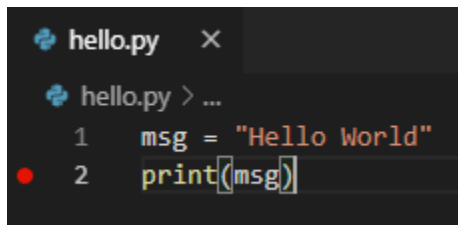
- Select one or more lines, then press **Shift+Enter** or right-click and select **Run Selection/Line in Python Terminal**. This command is convenient for testing just a part of a file.

- From the Command Palette (**Ctrl+Shift+P**), select the **Python: Start REPL** command to open a REPL terminal for the currently selected Python interpreter. In the REPL, you can then enter and run lines of code one at a time.

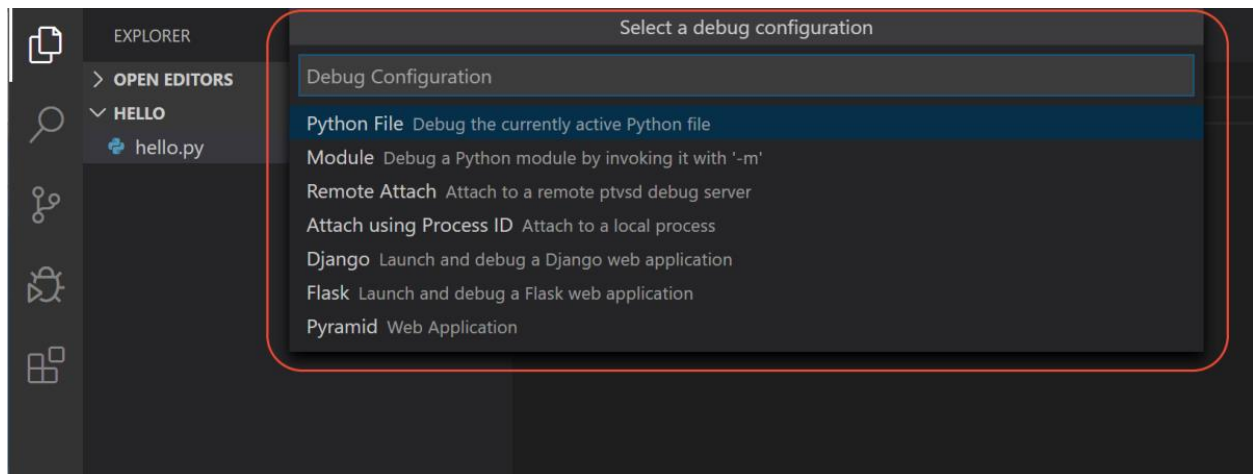
Configure and run the debugger#

Let's now try debugging our simple Hello World program.

First, set a breakpoint on line 2 of `hello.py` by placing the cursor on the `print` call and pressing **F9**. Alternately, just click in the editor's left gutter, next to the line numbers. When you set a breakpoint, a red circle appears in the gutter.



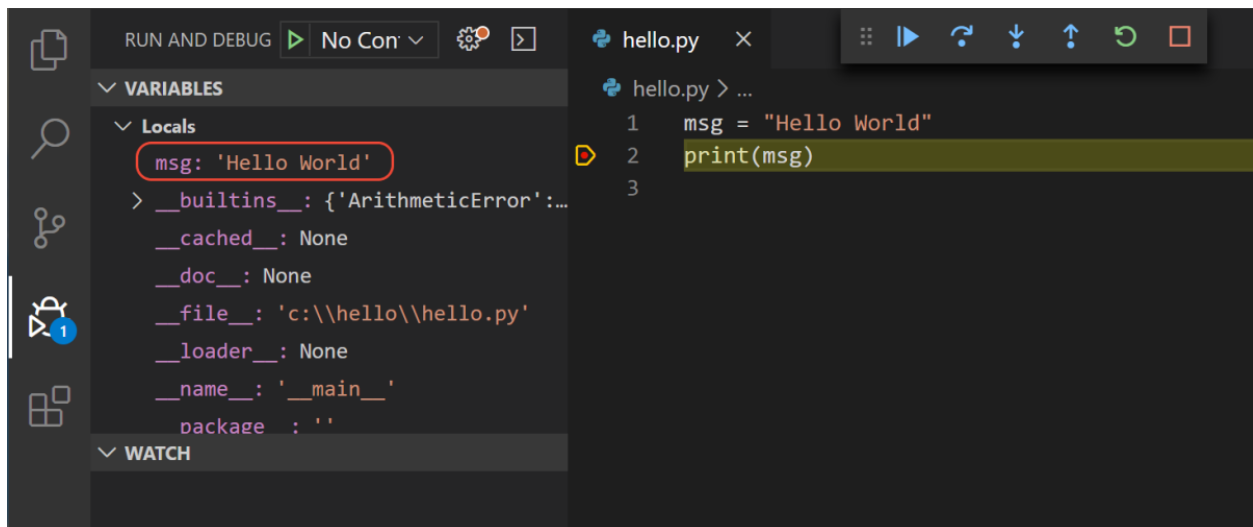
Next, to initialize the debugger, press **F5**. Since this is your first time debugging this file, a configuration menu will open from the Command Palette allowing you to select the type of debug configuration you would like for the opened file.



Note: VS Code uses JSON files for all of its various configurations; `launch.json` is the standard name for a file containing debugging configurations.

These different configurations are fully explained in [Debugging configurations](#); for now, just select **Python File**, which is the configuration that runs the current file shown in the editor using the currently selected Python interpreter.

The debugger will stop at the first line of the file breakpoint. The current line is indicated with a yellow arrow in the left margin. If you examine the **Local** variables window at this point, you will see now defined `msg` variable appears in the **Local** pane.



A debug toolbar appears along the top with the following commands from left to right: continue (F5), step over (F10), step into (F11), step out (Shift+F11), restart (Ctrl+Shift+F5), and stop (Shift+F5).



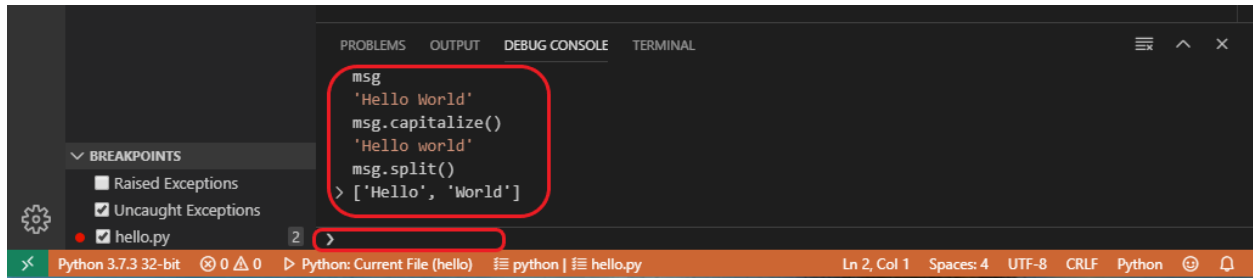
The Status Bar also changes color (orange in many themes) to indicate that you're in debug mode. The **Python Debug Console** also appears automatically in the lower right panel to show the commands being run, along with the program output.

To continue running the program, select the continue command on the debug toolbar (F5). The debugger runs the program to the end.

Tip Debugging information can also be seen by hovering over code, such as variables. In the case of `msg`, hovering over the variable will display the string `Hello world` in a box above the variable.

You can also work with variables in the **Debug Console** (If you don't see it, select **Debug Console** in the lower right area of VS Code, or select it from the ... menu.) Then try entering the following lines, one by one, at the `>` prompt at the bottom of the console:

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



Select the blue **Continue** button on the toolbar again (or press F5) to run the program to completion. "Hello World" appears in the **Python Debug Console** if you switch back to it, and VS Code exits debugging mode once the program is complete.

If you restart the debugger, the debugger again stops on the first breakpoint.

To stop running a program before it's complete, use the red square stop button on the debug toolbar ([Shift+F5](#)), or use the **Run > Stop debugging** menu command.

That's the basics to using Python in VS Code!