

Python Setup For Simulator Training

The following instructions outline how to get a machine set up with the configuration we will be using for training. If you are experienced with Python and won't need to follow along directly, you are welcome to use whatever configuration you are most comfortable with.

Installing Python

For this course, we will be working with Python 3.7.4. The Windows x86-64 executable installer can be obtained from <https://www.python.org/downloads/release/python-374/>

The installer will offer the option to add Python 3.7 to PATH. For the purposes of training, we would recommend this option. (If you intend to use multiple versions of Python, it's better to leave it unchecked.)



After installing Python, reboot before proceeding.

Installing Pip

Pip is a package management tool for Python. We will be using a number of modules not included in the core package, and will be installing these packages using Pip.

To install Pip, download the python script "get-pip.py" from <https://bootstrap.pypa.io/get-pip.py>

Save this script to your machine, then open a CMD window, navigate to the directory where the script is saved, and run the command:

```
> python get-pip.py
```

```
D:\Downloads>python get-pip.py
Collecting pip
  Using cached https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b0c6f96af8ce16cfbc1cb06bedf8e9164ce5551ec1/pip-19.3.1-py2.py3-none-any.whl
Collecting wheel
  Downloading https://files.pythonhosted.org/packages/00/83/b4a77d044e78ad1a45610eb88f745be2fd2c6d658f9798a15e384b7d57c9/wheel-0.33.6-py2.py3-none-any.whl
Installing collected packages: pip, wheel
  Found existing installation: pip 19.3.1
  Uninstalling pip-19.3.1:
    Successfully uninstalled pip-19.3.1
  Successfully installed pip-19.3.1 wheel-0.33.6
```

To ensure installation went correctly, run the command following command to display the installed version of Pip:

```
> pip -v
```

```
D:\Downloads>pip -U
pip 19.3.1 from c:\users\thomas\AppData\Local\Programs\Python\Python37\lib\site-packages\pip (python 3.7)
```

Installing Python Modules

Once Pip is set up, installing the requisite Python modules is straightforward. To install Pywin32, run the command:

```
> pip install pywin32
```

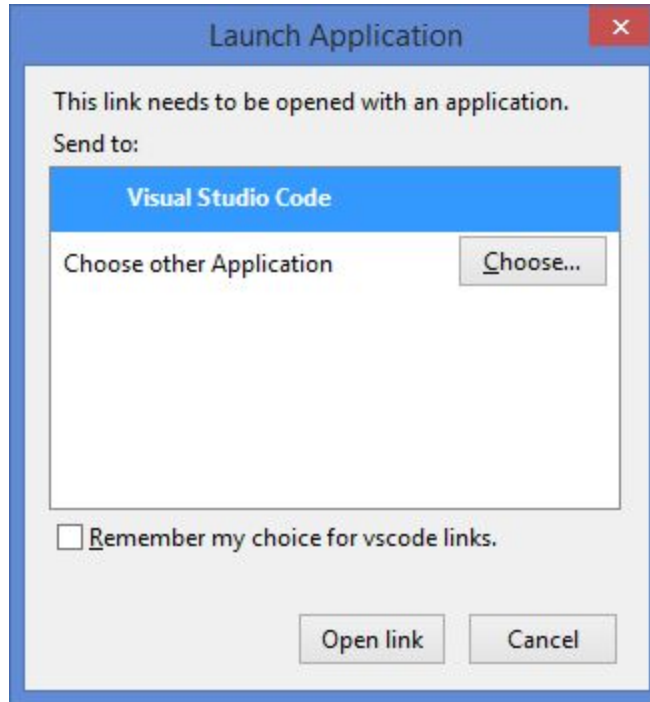
Repeat the process for the following modules:

- pywin32
- appjar
- pyQT5
- matplotlib
- pandas

Installing Visual Studio Code

Visual Studio Code (VSC) is a free IDE with support for Python. You can obtain the required installation file from <https://code.visualstudio.com/>.

Once VSC is installed, you'll want to install the Python extension available here: <https://marketplace.visualstudio.com/items?itemName=ms-python.python>. When you click the installation link, you will be asked to confirm that Visual Studio Code is installed, and then a "Launch Application" dialog will open asking you to confirm that you want to use Visual Studio Code to open the link.



After opening the link in VSC, click the Install button to run the installation. Upon completion, you may be presented with a “Reload Required” button as well.

Testing Your Setup

To test your configuration, first open a working directory in VSC using **File > Open Folder**.

Create a new file in VSC using **File > New File**

In your new file, write the line:

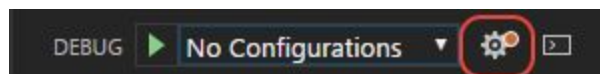
```
print('Hello World')
```

Save the file to your working directory as “HelloWorld.py” with **File > Save**.

Open Debug View by clicking the Debug icon on the left edge of the window:



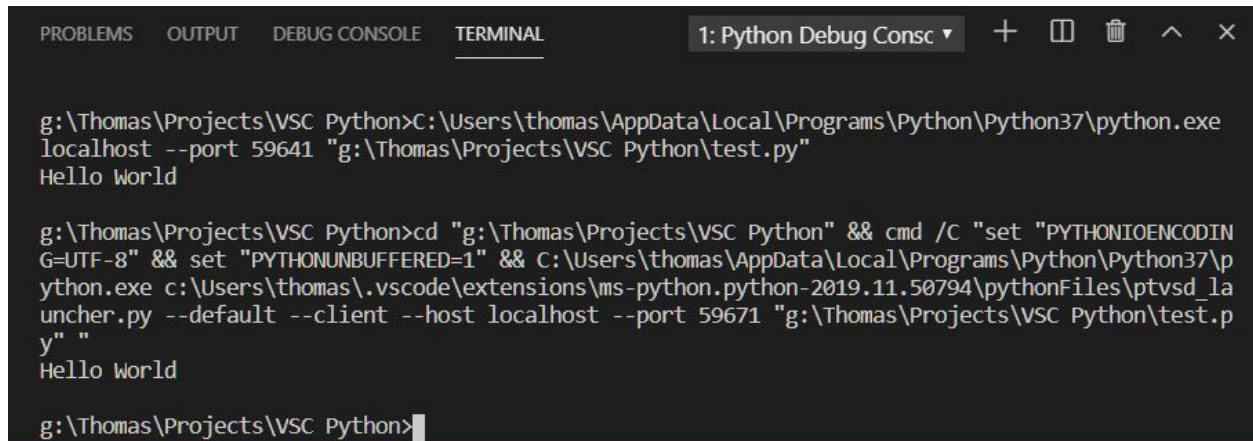
If the configurations dropdown displays “No Configurations” and an orange dot is displayed on the settings icon, you will need to create a debug configuration.



To generate a launch.json file with Python configurations, click the settings button, and select **Python File** from the dropdown that opens in the Command Palette. This will create a JSON file with a

pre-defined configuration for Python debugging. This default should be fine for our needs; go ahead and close the launch.json file tab.

Now you can run your script by navigating to the HelloWorld.py file tab and clicking the green Debug arrow, or by hitting the hotkey **F5**. Your code will run in the Terminal tab at the bottom of the window:

A screenshot of a terminal window with a dark background and light text. The window title bar shows '1: Python Debug Consc' and standard window control icons. The terminal content shows two separate Python execution commands and their outputs. The first command runs 'python.exe localhost --port 59641' on 'test.py', resulting in 'Hello World'. The second command runs 'cmd /C' with environment variables and the same Python command, also resulting in 'Hello World'. The prompt 'g:\Thomas\Projects\VSC Python>' is visible at the end of the terminal.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python Debug Consc
g:\Thomas\Projects\VSC Python>C:\Users\thomas\AppData\Local\Programs\Python\Python37\python.exe
localhost --port 59641 "g:\Thomas\Projects\VSC Python\test.py"
Hello World

g:\Thomas\Projects\VSC Python>cd "g:\Thomas\Projects\VSC Python" && cmd /C "set "PYTHONIOENCODIN
G=UTF-8" && set "PYTHONUNBUFFERED=1" && C:\Users\thomas\AppData\Local\Programs\Python\Python37\p
ython.exe c:\Users\thomas\.vscode\extensions\ms-python.python-2019.11.50794\pythonFiles\ptvsd_la
uncher.py --default --client --host localhost --port 59671 "g:\Thomas\Projects\VSC Python\test.p
y" "
Hello World

g:\Thomas\Projects\VSC Python>
```