

Setup up a Python Environment with IDE for the OPC UA Assignment

April 2023

This assignment requires you to use Python 3 ,IDE and Python OPC UA libraries. In order to complete it, the following steps are necessary (if you do not have Python set up on your computer):

1. Install Python 3
2. Install a Code-Editor, IDE (Integrated Development Environment)
3. Launch Code-Editor via Anaconda Navigator (Anaconda3) (Engineering lab PC only)
4. Setup a Code-Editor, IDE
5. Create a Python Virtual Environment
6. Download OPC UA Python libraries

Please note:

If you use your own desktop or laptop to complete the assignment, you may need to install Python and IDE, please follow steps 1->2->4->5->6 (you don't need to do step 3)

If you are using a PC in the Engineering Building computer lab, please skip steps 1 and 2 and follow steps 3->4->5->6 (make sure to start from step 3)

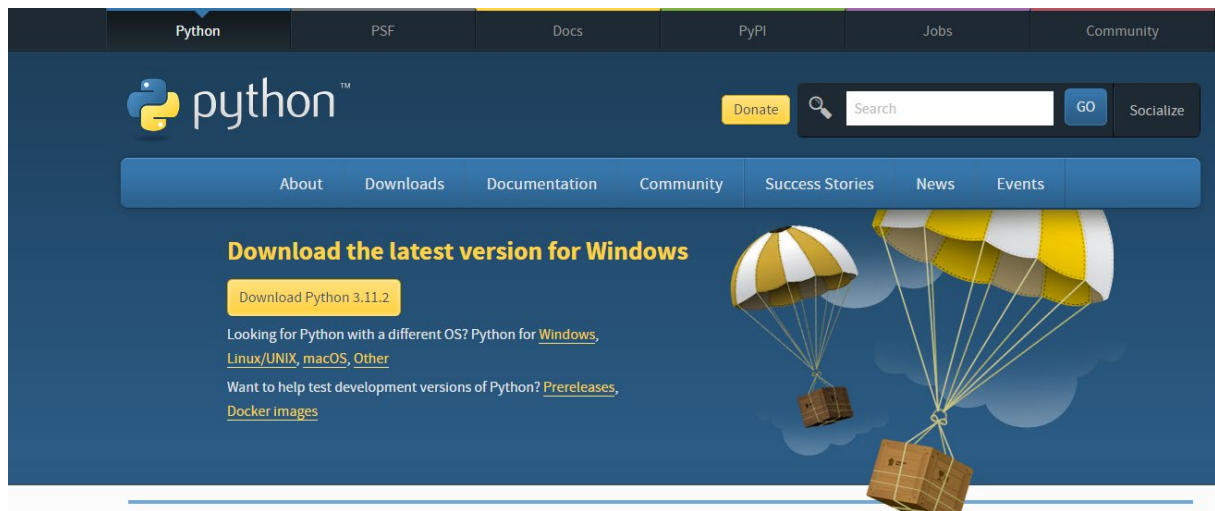
It is possible that you may have already completed some of these steps. Please read the document anyway to make sure you do not miss anything.

1. Install Python 3

Python has been around for decades; in recent years it has become one of the most popular programming languages in the world.

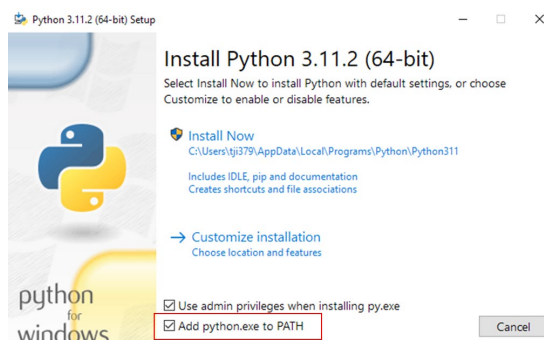
1.1. Installing Python

Head over to <https://www.python.org/downloads/> and download the distribution (e.g. build 3.11.2) appropriate for your platform. Hit the big **yellow button** (e.g. **Download Python 3.11.2**) as shown in the following screen).

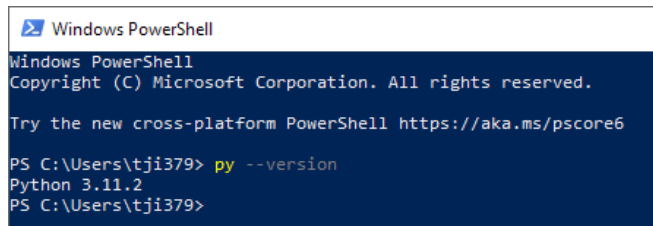


You may direct to a page with different Python builds (bottom of the page); choose the right version for your computer, 64 or 32-bit version.

You may need admin right to install it. Different platforms have different setup procedures. During installation on Windows, there might be additional options. Recommend checking 'Add Python _._ to PATH', but if you use VS Code (outlined in the next section) you should be fine either way. No need to go for "Customize installation".



Verify your Python installation: From your operating system open a new terminal, bash or shell window. Execute "python -version".



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\tji379> py --version
Python 3.11.2
PS C:\Users\tji379>
```

If the command does not execute or not return the version you just installed, you can try the following:

1. Restart your computer.
2. Try "python3" or "py" instead of "python"

If "python3" or "py" succeeds, use this alternative command for creating a virtual environment as described in section 3.

1.2. Learning Python

There are many resources out there about learning Python. At the conclusion of installing Python, you will be prompted for some [tutorials](#) and [documentation](#). Otherwise, you may find a very concise primer (~10min) for Python at <https://www.stavros.io/tutorials/python/>.

2. Install a Code-Editor

The Python distribution you installed in the previous step has a simple IDE included. It is recommended that you install a more advanced one. There are many free or commercial solutions available. You might already have your preferred IDE or code-editor. If not, we recommend Visual Studio Code (VS Code). The rest of the instruction refers to VS Code.

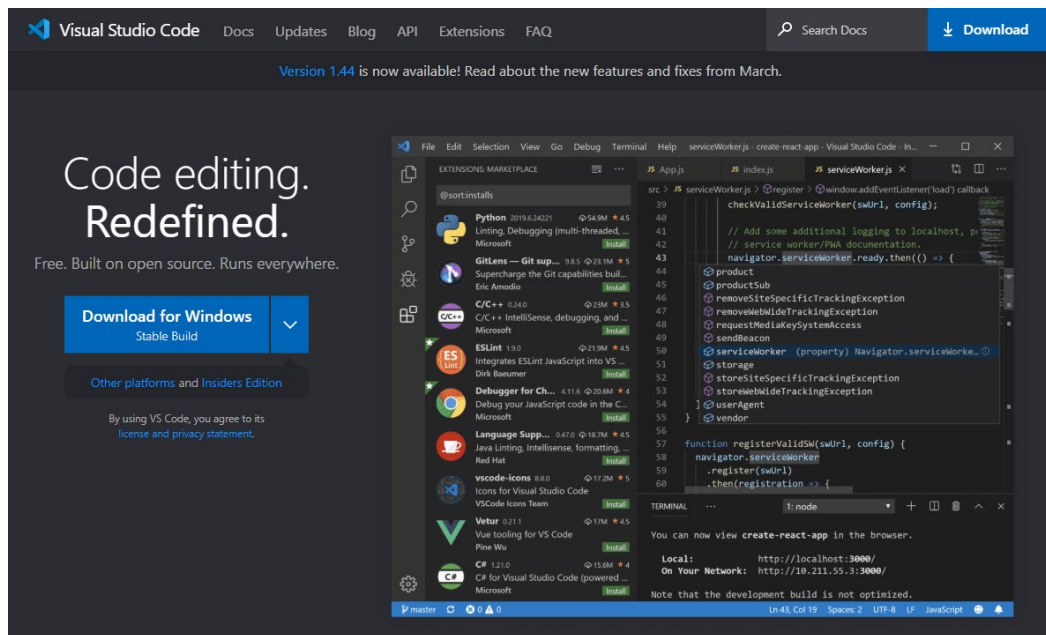
2.1. Visual Studio Code

Not to confuse Visual Studio with VS Code - although both are IDEs and both are developed by Microsoft. Visual Studio is very powerful commercial software. It provides in-depth debugging and development tools, especially for C#, C++ and Windows Platforms. However, it is limited to Windows systems and for this assignment, it might be a bit too bloated.

VS Code is a lightweight alternative. A multitude of plug-ins allows the user to add more functionalities and customise the software to their liking.

2.2. Download and install VS Code

Follow the link <https://code.visualstudio.com/> and download the appropriate binary for your platform (Win, mac, Linux). Choose a stable build.

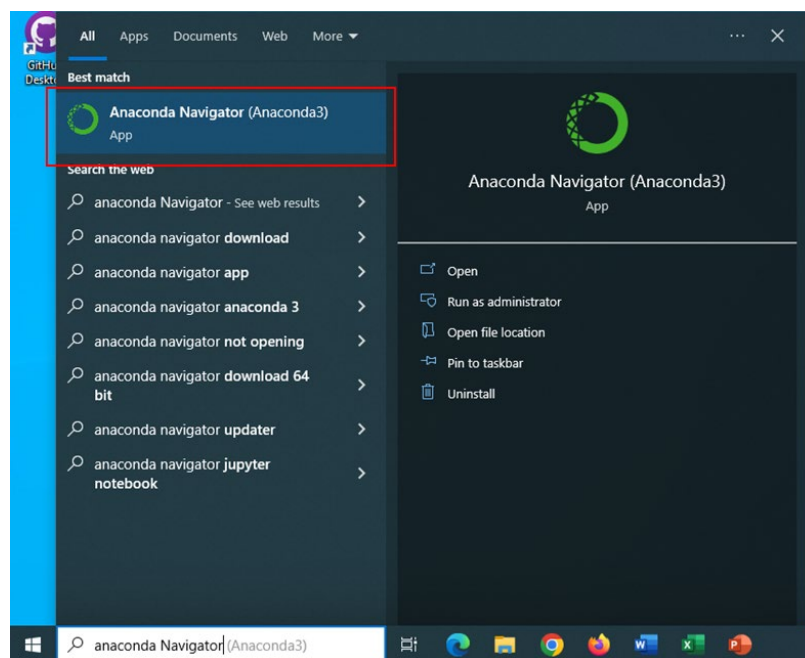


Note that different platforms will have different setup procedures. As of April 2023, we had Version 1.76 VS Code.

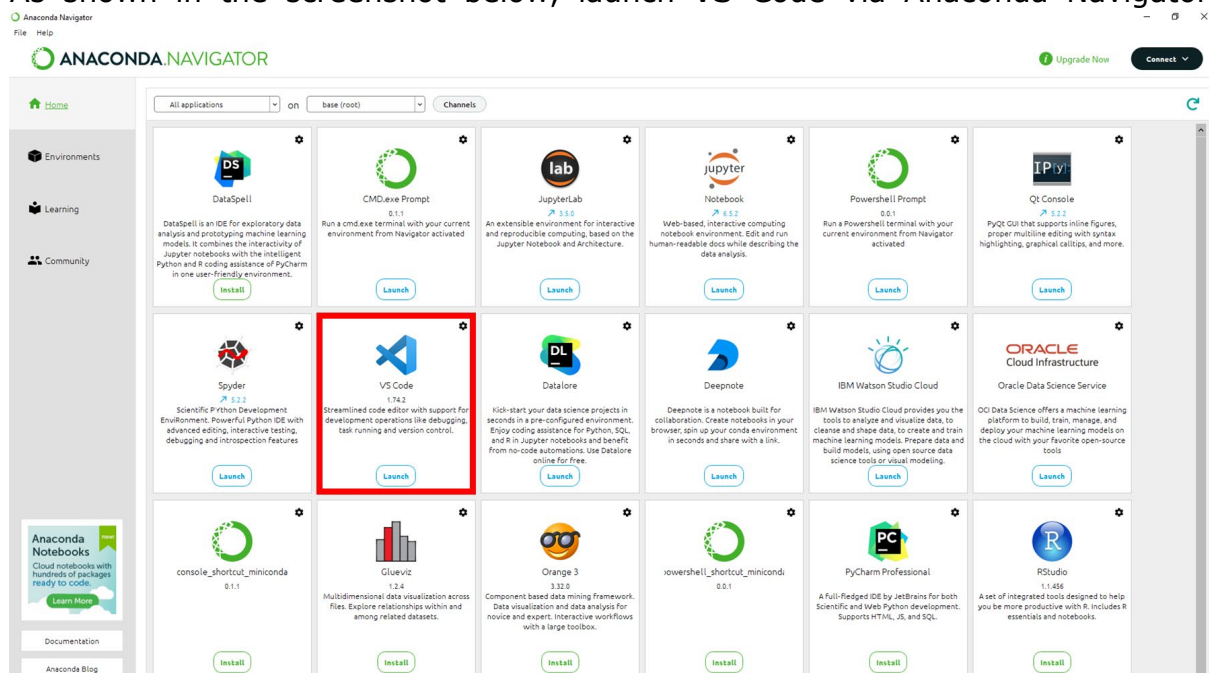
3. Launch Code-Editor via Anaconda Navigator (Anaconda3) (Engineering lab PC only)

If you are using a lab PC from the engineering building for your assignment, please make sure you start with this step. Due to the environment setting of the lab computers, you will need to start VS Code through Anaconda; if you do not do this, you may not be able to run the Pip install function properly in subsequent operations.

Find "**Anaconda Navigator (Anaconda3)**" in the Engineering lab PC and launch this application.



As shown in the screenshot below, launch VS Code via Anaconda Navigator.



4. Setup a Code-Editor

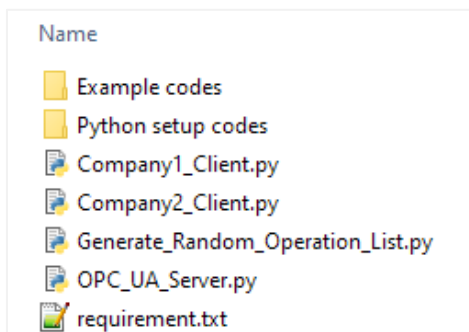
4.1. VS Code Setup for Python

Depending on the programming language you use, a proper extension needs to be installed. So, to use VS Code for Python programming, install the Python extension. You can do so from inside VS Code. Start VS Code, click the options corresponding to and in the sequence of numbers 1, 2 and 3.



4.2. Set up your assignment folder

Create a brand-new folder on your computer for this assignment, for example, "D:\Python\OPC UA Assignment\Source codes". Download your individual archived files from Canvas. Extract them into your assignment folder. You should have something like,

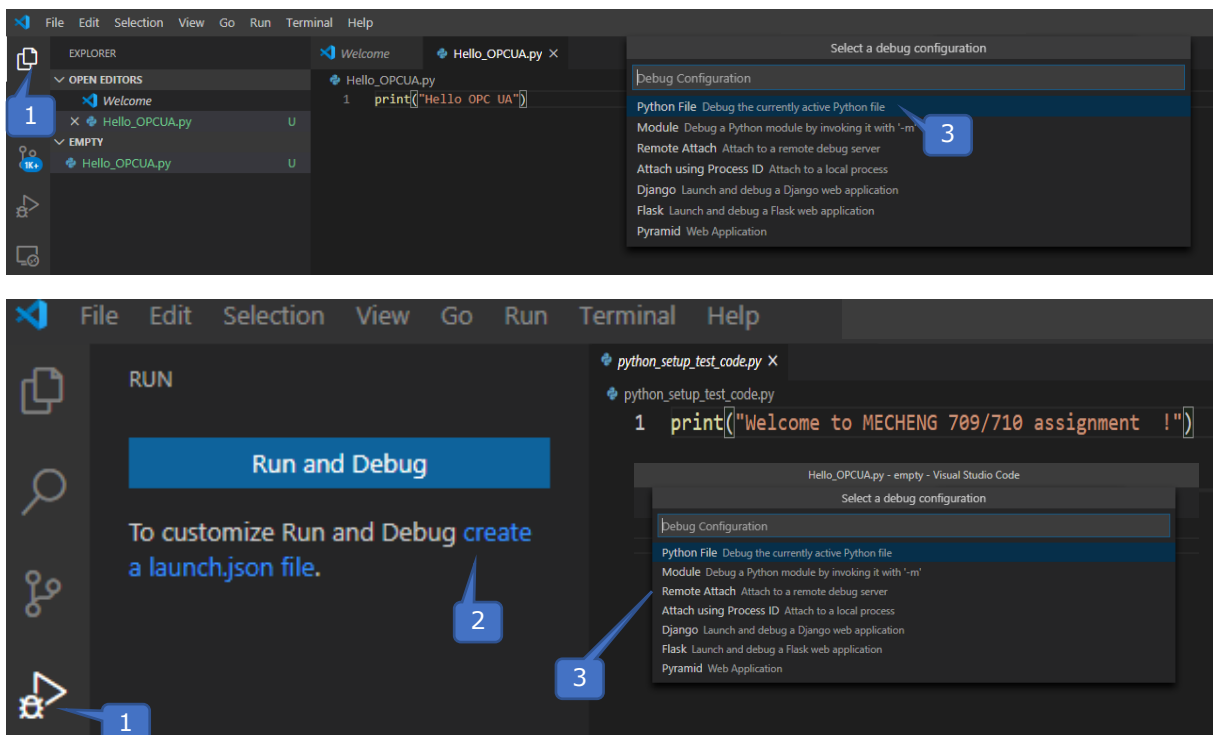


(It is inadvisable to store the assignment folder on a network drive, such as Google Drive, as doing so may lead to the occurrence of an error)

4.3. VS Code debug configuration

After installing the Python extension, there is one more step necessary before you can run and debug a Python script - you need to create a VS Code *debug configuration*.

In VS Code, go back to the 'Explorer' view and navigate to the folder where the provided "python_setup_test_code.py" Python script is located. Select the file to open, and then press 'F5', which means start debugging. Because this is the first debugging session, VS Code will ask you how exactly you want to launch it. Select "Python File". To enable this configuration permanently, click on "create a launch.json file" in the debug tab. Again, select "Python File".



If you are prompted to bring in a Python Interpreter, please follow the instruction to do so.

Now you are set up and can launch/debug Python scripts (in this particular working folder) by pressing 'F5'.

At this stage VS Code might suggest the installation of the Python Linter "pylint". You can safely ignore this message at this stage. Once you have completed Step 3 (setting up a Virtual Environment), and activated the virtual environment, you can click "Install". Without the virtual environment the installation is likely to fail due to access right. You do not need to worry, if you do not use or install the linter.

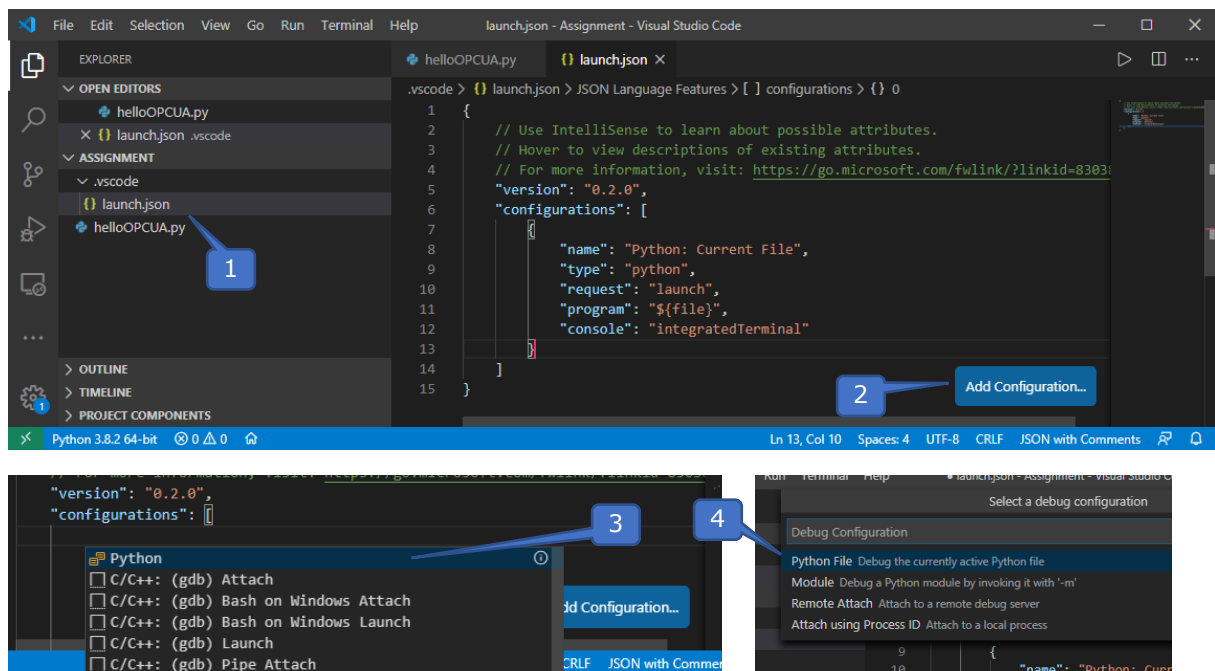
More details can be found in the VS Code documentation. Particularly at:

<https://code.visualstudio.com/docs/python/python-tutorial>

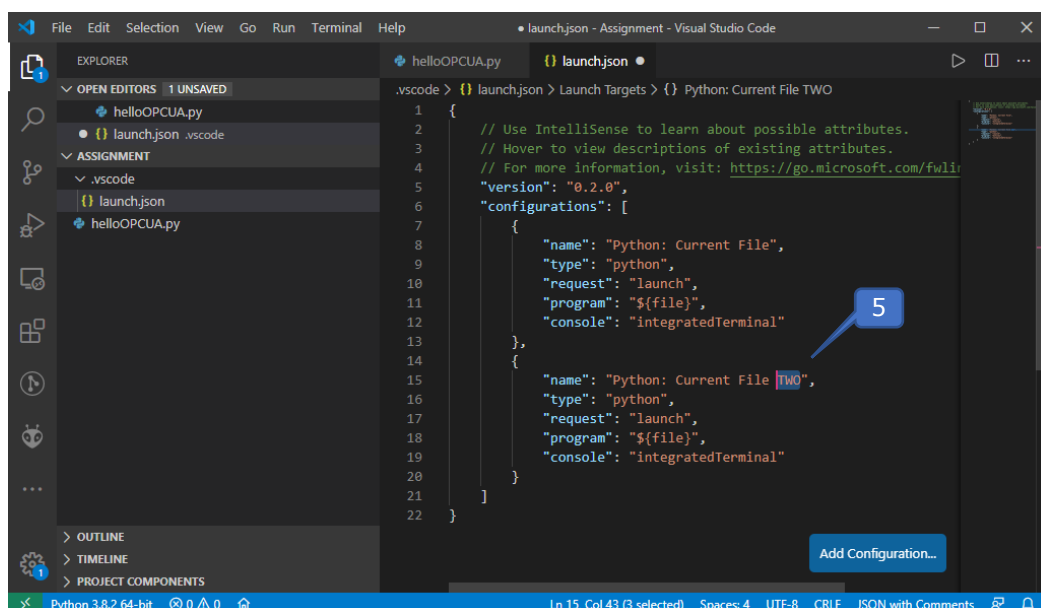
https://code.visualstudio.com/docs/python/debugging#_initialize-configurations.

4.3.1. Running two Python scripts at once

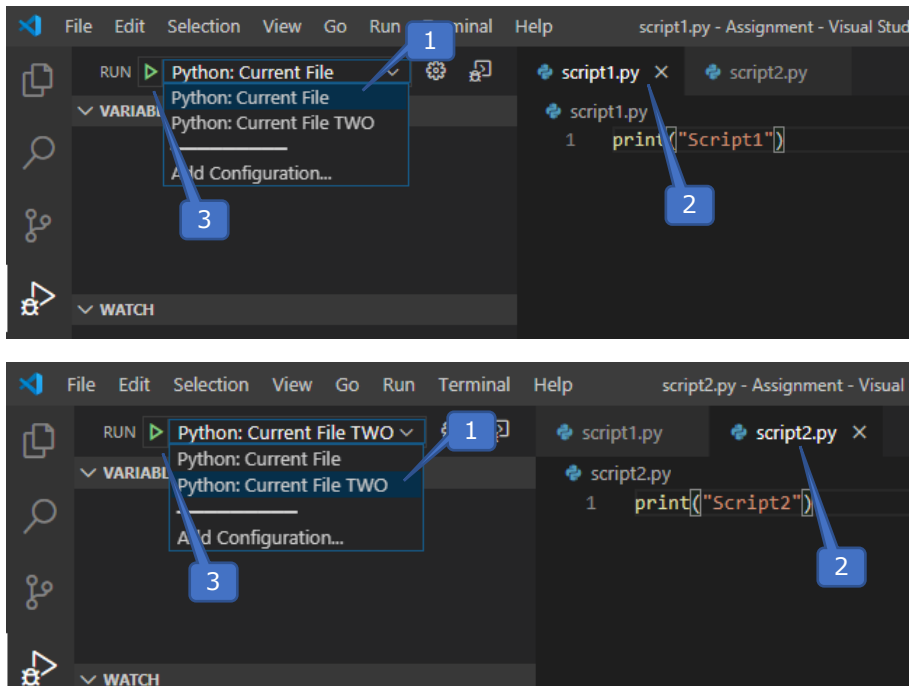
What you did in the previous step was to create a debug configuration for a single Python script file. The configuration is stored inside `“.vscode/launch.json”`. However, for this assignment, you will need to run two scripts at the same time. Opening two instances of VS Code inside the same folder does not seem to work. The solution is to create a second debug configuration.



Open the file `“.vscode/launch.json”`. Click on `“Add Configuration”` and select `“Python”`, and `“Python File”` in the menus. The assistant should add a second entry to the file that is the same as the existing entry. You can now edit the name, to distinguish it from the first element.



Save the file. To start a second script at the same time: navigate to the 'Run' tab in VS Code, select your first debug configuration and your first python script and start debugging. After that select the second debug configuration and your second python script and start debugging.

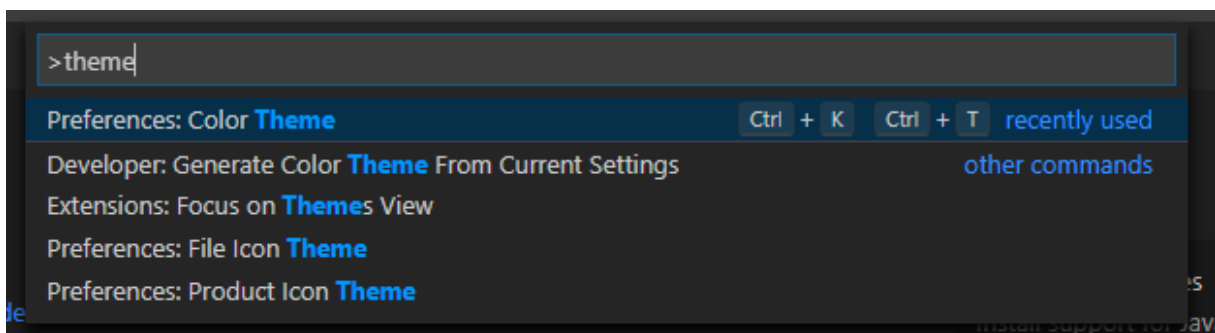


You will be able to debug both scripts at the same, which is very useful for this type of server-client applications.

4.4. Select Theme

If you do not like the default dark/black user interface/theme, you can choose from different pre-installed themes or add new ones through the extension marketplace.

Hit 'F1' to bring up the command palette in VS Code, and type "theme".

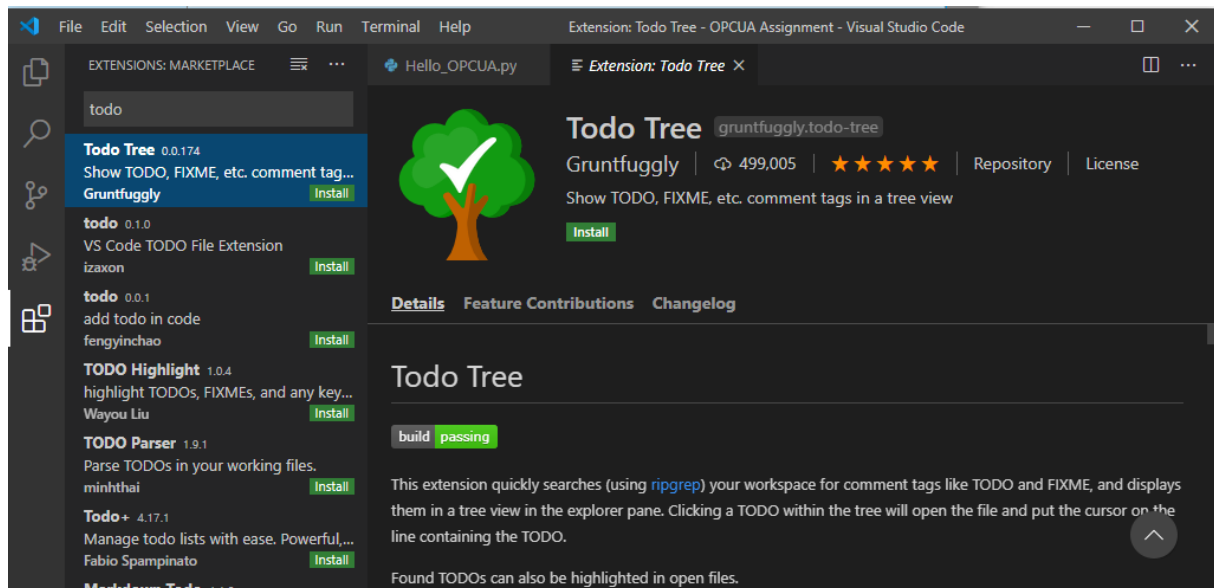


From the list select "Preferences: Colour Theme" and select your preferred theme in the appearing list.

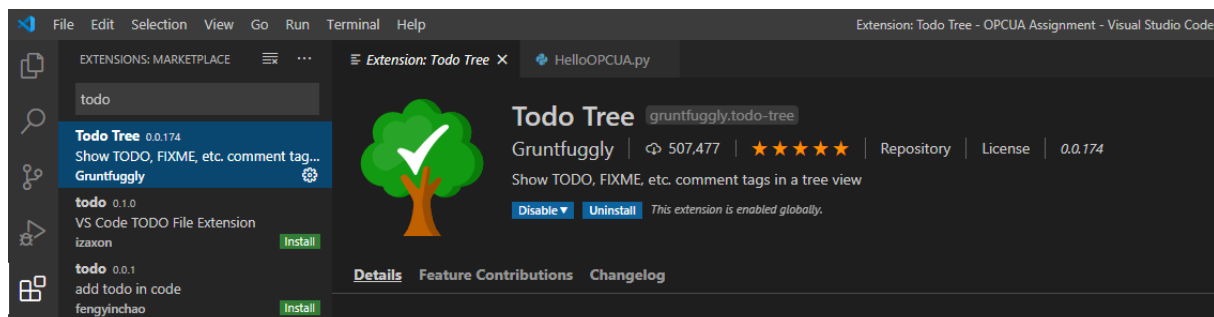
4.5. Other Plugins

You are free to install other plugins you might find helpful. In VS Code go to the 'Extension' view (The one you used to select and install the Python extension).

The “**Todo Tree**” plugin will highlight Todo-comments embedded in a script. They are used throughout the assignment code to highlight which parts you need to complete. Using this plugin will help you to not miss anything. Install Todo Tree as shown in the following screenshot.



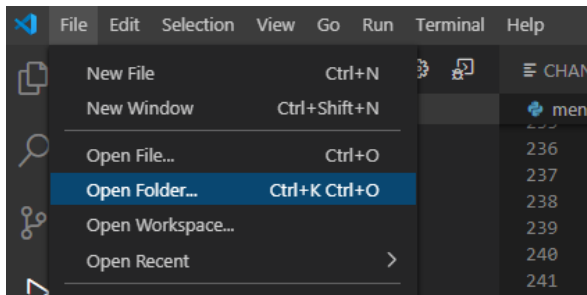
If your computer shows the following, it means that you have got the “Todo Tree” plugin. As of April 2023, we had Version v0.0.224.



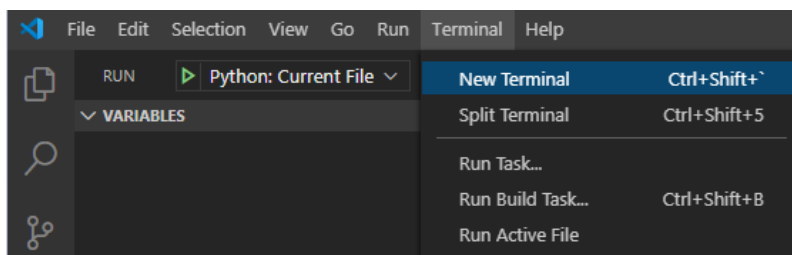
5. Setting up a Virtual Environment

A Python Virtual Environment makes your software project portable. The code libraries you download should be kept inside your workspace, but not installed globally on your computer. For this, we will use a Virtual Environment so the OPC UA libraries do not clutter up the rest of your computer.

Assume that you have downloaded and extracted your individual group zip file from the CANVAS group section into your assignment folder, e.g. D:\Python\OPC UA Assignment. Open this folder inside VS Code: "File>>Open Folder".



Next open a new Terminal. "Terminal>>New Terminal".



When you are inside your assignment folder, execute the following command: "python -m venv venv". Make sure to verify your Python installation as outlined in section 1. If you need to use "python3" or "py", your command will be "python3 -m venv venv" or "py -m venv venv", respectively. Once the virtual environment is set up and activated, you do not need to substitute the python command anymore, i.e. in your virtual environment, "python" will work.

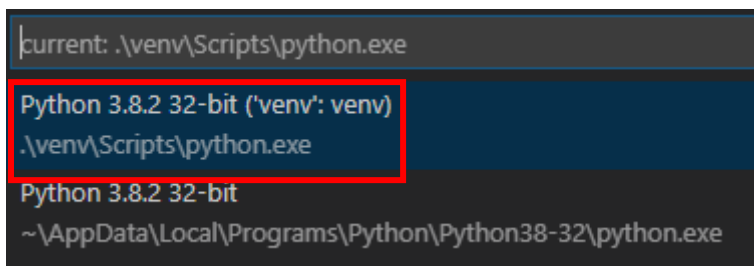
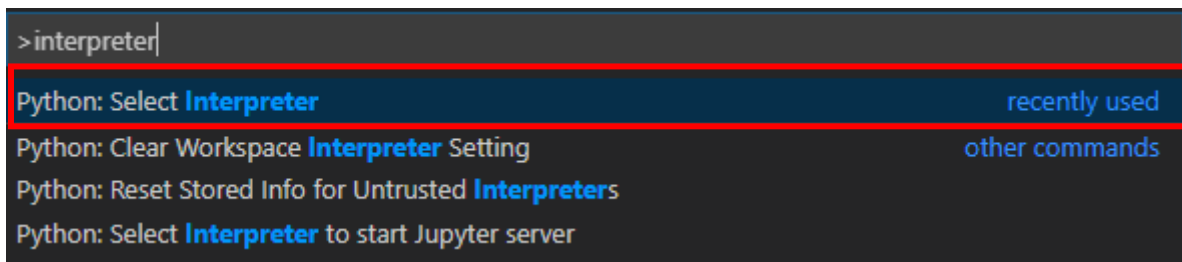
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

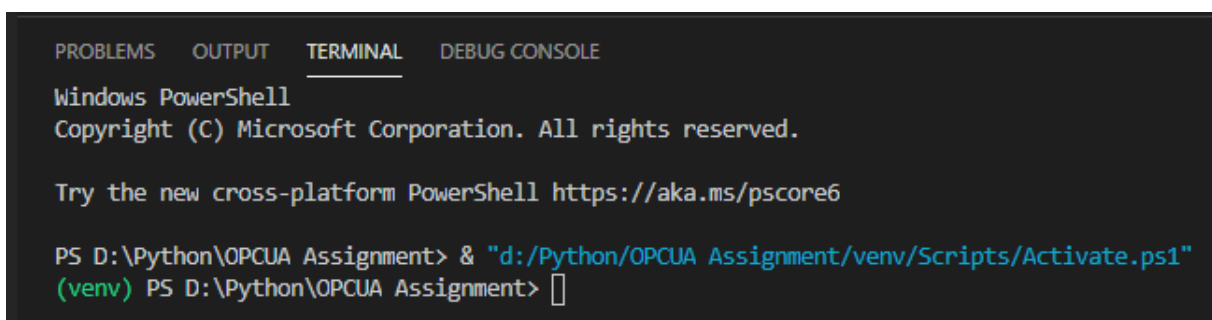
PS D:\Python\OPCUA Assignment> python -m venv venv
```

Explanation: By using python with the module (-m) parameter you start the "venv" (virtual environment) module and pass to it the parameter "venv". The python module "venv" will now create a Virtual Environment in the local folder venv.

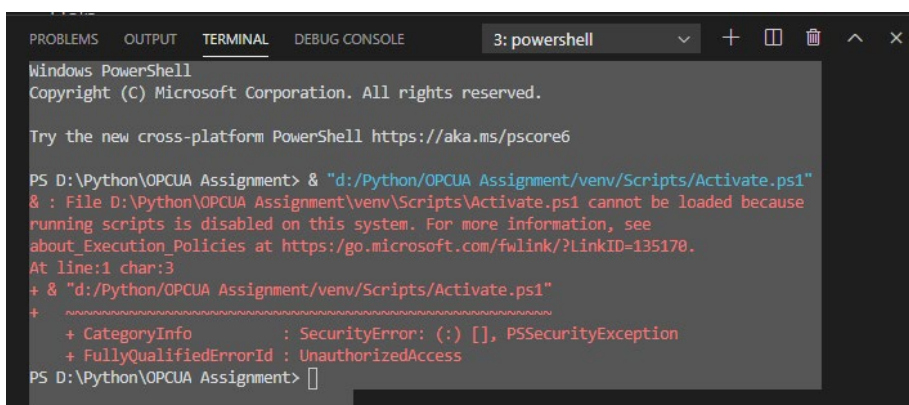
Once you created the virtual environment, press 'F1' to bring up the command palette. Type "interpreter" to start your search and select "Python: Select Interpreter"). In the dropdown menu choose the entry that refers to your Virtual Environment.



Now VS Code is configured to use the Virtual Environment you created. If you open a new terminal window it will automatically activate this Virtual Environment, which will look like this (assuming your drive is "D"):



If you are on Windows and you see this error at this stage:



This is caused by the default settings on Windows. Stay in the current terminal and execute: "Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser"

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Python\OPCUA Assignment> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

After this is executed, start a **new** terminal and you should have this,

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Python\OPCUA Assignment> & "d:/Python/OPCUA Assignment/venv/Scripts/Activate.ps1"
(venv) PS D:\Python\OPCUA Assignment> 
```

This is a known issue, unlikely to be fixed: <https://github.com/Microsoft/vscode-python/issues/2559>

More information can be found at:

<https://code.visualstudio.com/docs/python/environments>

If you do not use VS Code, or do not select the correct interpreter you will have to activate your environment by hand. This is described in section 12.2 in the python documentation <https://docs.python.org/3/tutorial/venv.html>.

6. Download OPC UA libraries

Python has a package manager called *Pip*, which enables you to easily download additional code libraries. We will use it to download the necessary OPC UA libraries.

Make sure you have activated the Virtual Environment described in the previous section. Otherwise the libraries will be installed inside your main Python folder.

The easiest way to get all the necessary files at once is to use the provided "requirements.txt" file. All it does is list the names of the necessary software packages.

You use it by executing "pip install -r requirement.txt".

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Python\OPCUA Assignment> & "d:/Python/OPCUA Assignment/venv/Scripts/Activate.ps1"
(venv) PS D:\Python\OPCUA Assignment> pip install -r requirement.txt
```

(This is the same as installing all libraries one by one:

pip install opcua

pip install numpy)

You may be asked to upgrade pip. If so, execute 'python -m pip install --upgrade pip' command.