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

April 2024

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)

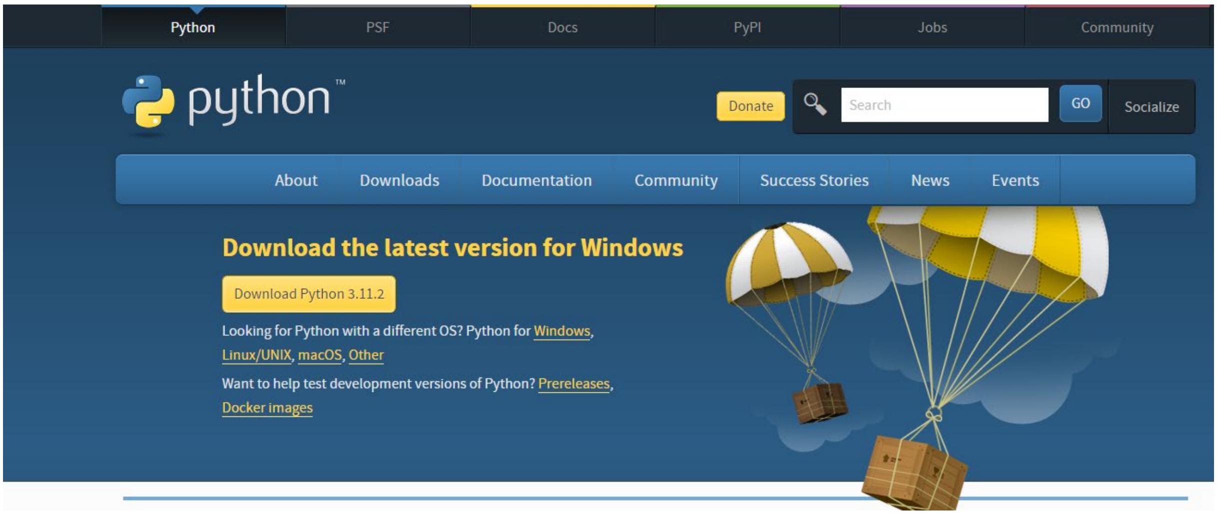
It is possible that you may have already completed some of these steps. Please read the document anyway to make sure you do not miss anything.

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). There may be newer version, but do not use the pre-release version.



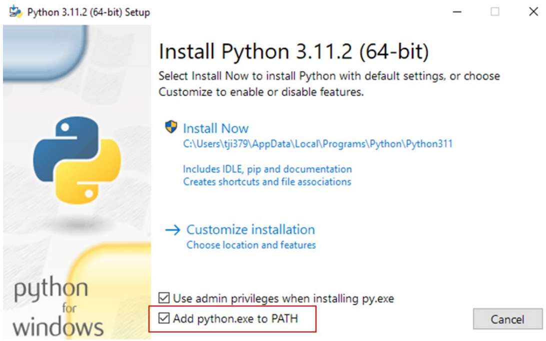
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

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".



Install Python 3.11.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

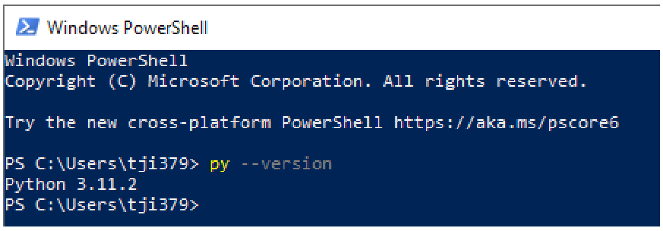
C:\Users\tji379\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

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.

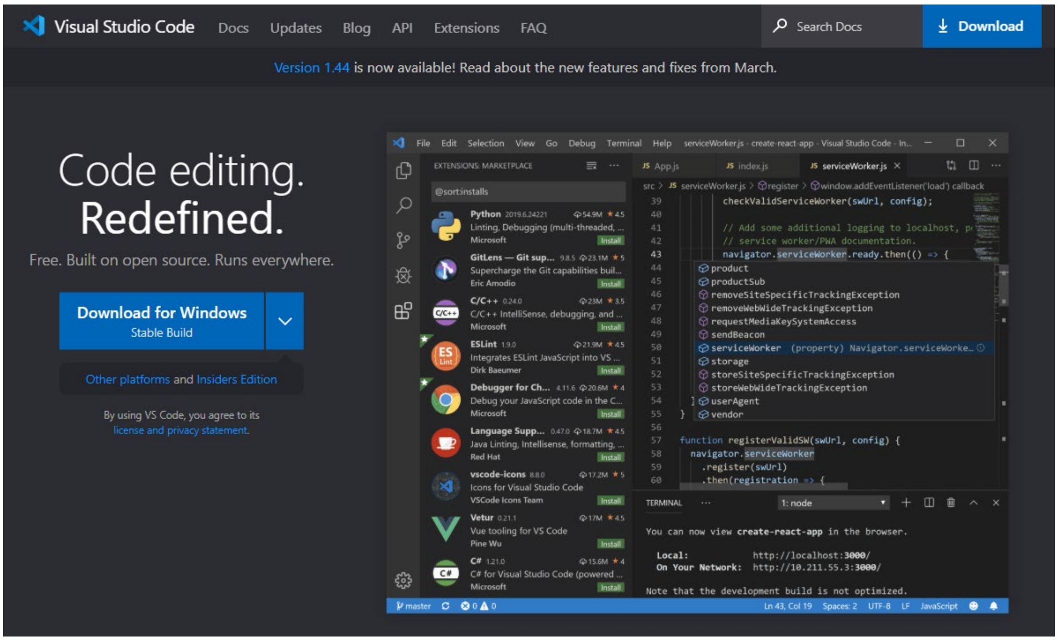
2.1. Visual Studio Code

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.


Visual Studio Code
Docs
Updates
Blog
API
Extensions
FAQ
Search Docs
Download
File
Edit
Selection
View
Go
Debug
Terminal
Help
serviceWorker.js - create-react-app - Visual Studio Code - In...
EXTENSIONS: MARKETPLACE
@sort:installs
Python 2019.6.24221 54.9M ★4.5
Linting, Debugging (multi-threaded, ...
Microsoft
Install
GitLens — Git sup... 9.8.5 23.1M ★5
Supercharge the Git capabilities buil...
Eric Amodio
Install
C/C++ 0.24.0 23M ★3.5
C/C++ IntelliSense, debugging, and ...
Microsoft
Install
ESLint 1.9.0 21.9M ★4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer
Install
Debugger for Ch... 4.11.6 20.6M ★4
Debug your JavaScript code in the C...
Microsoft
Install
Language Supp... 0.47.0 18.7M ★4.5
Java Linting, Intellisense, formatting, ...
Red Hat
Install
vscode-icons 8.8.0 17.2M ★5
Icons for Visual Studio Code
VSCode Icons Team
Install
Vetur 0.21.1 17M ★4.5
Vue tooling for VS Code
Pine Wu
Install
C# 1.21.0 15.6M ★4
C# for Visual Studio Code (powered ...
Microsoft
Install
App.js
index.js
serviceWorker.js
src > serviceWorker.js > register > window.addEventListener('load') callback
product
productSub
removeSiteSpecificTrackingException
removeWebWideTrackingException
requestMediaKeySystemAccess
sendBeacon
serviceWorker (property) Navigator.serviceWorke...
storage
storeSiteSpecificTrackingException
storeWebWideTrackingException
userAgent
vendor
TERMINAL
1: node
master
0 0
Ln 43, Col 19
Spaces: 2
UTF-8
LF
JavaScript

Version 1.44 is now available! Read about the new features and fixes from March.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its
license and privacy statement.

```
          checkValidServiceWorker(swUrl, config);

          // Add some additional logging to localhost, p
          // service worker/PWA documentation.
          navigator.serviceWorker.ready.then(() => {

      }
    }

  function registerValidSW(swUrl, config) {
    navigator.serviceWorker
      .register(swUrl)
      .then(registration => {
```

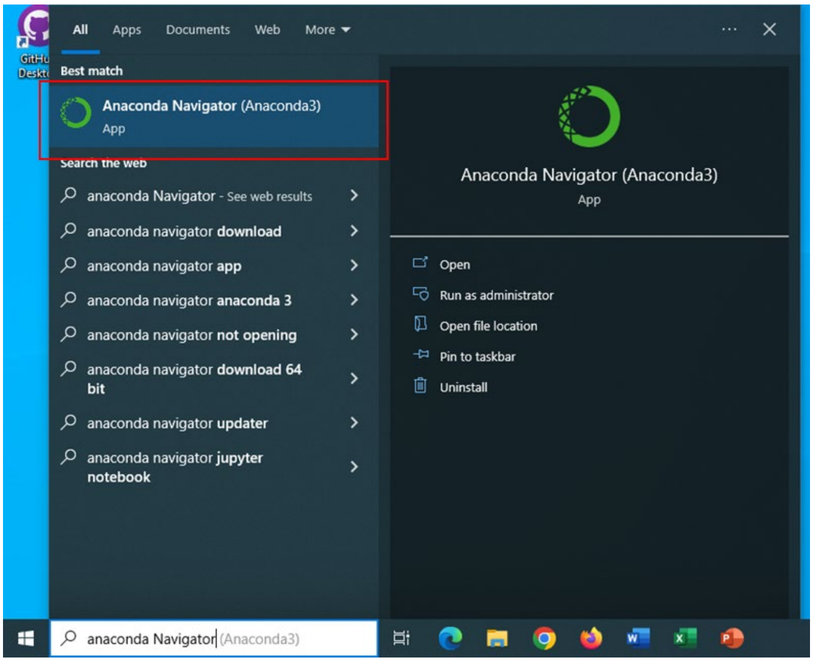
```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```

Note that different platforms will have different setup procedures. As of July 2023, we had Version 1.80 VS Code.

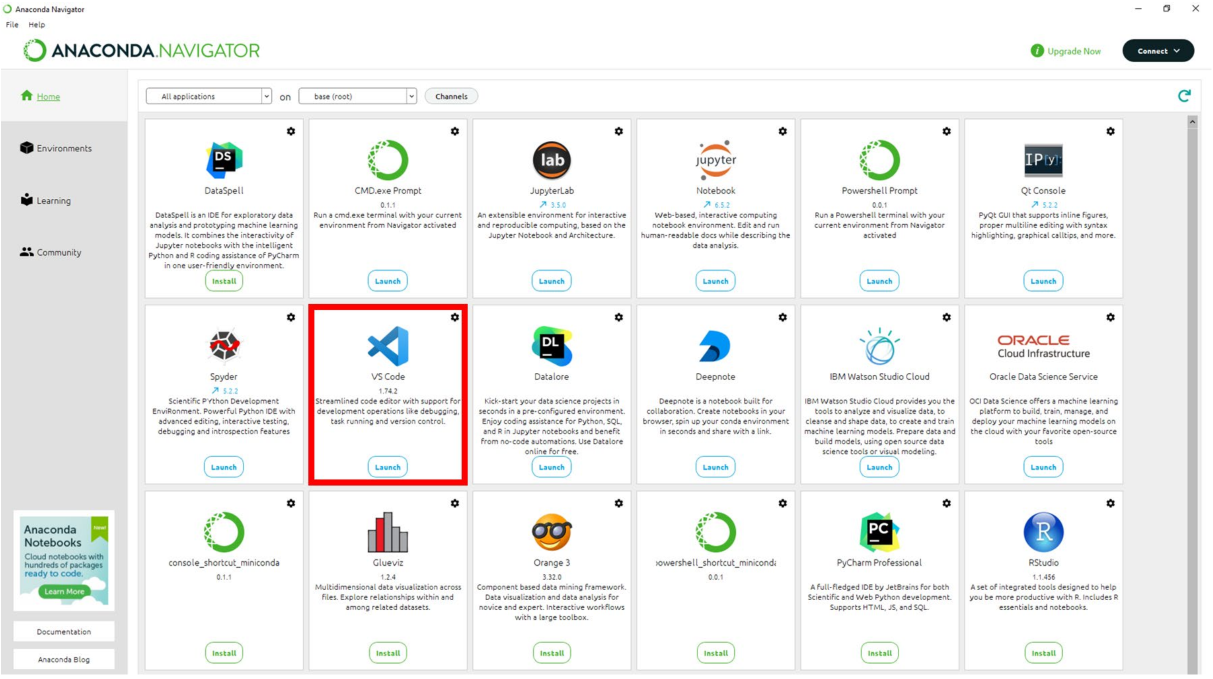
3. Launch Code-Editor via Anaconda Navigator (Anaconda3) (Engineering lab PC only)

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.


All
Apps
Documents
Web
More
GitHub Desktop
Best match
Anaconda Navigator (Anaconda3)
App
Search the web
anaconda Navigator - See web results
anaconda navigator download
anaconda navigator app
anaconda navigator anaconda 3
anaconda navigator not opening
anaconda navigator download 64 bit
anaconda navigator updater
anaconda navigator jupyter notebook
Anaconda Navigator (Anaconda3)
App
Open
Run as administrator
Open file location
Pin to taskbar
Uninstall
anaconda Navigator (Anaconda3)

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



DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch

JupyterLab
↗ 3.5.0
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
↗ 6.5.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch

Qt Console
↗ 5.2.2
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

Spyder
↗ 5.2.2
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

VS Code
1.74.2
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch

Datalore
Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
Launch

Deepnote
Deepnote is a notebook built for collaboration. Create notebooks in your browser, spin up your conda environment in seconds and share with a link.
Launch

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch

ORACLE Cloud Infrastructure
Oracle Data Science Service
OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools
Launch

console_shortcut_miniconda
0.1.1
Install

Glueviz
1.2.4
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.32.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

›owershell_shortcut_miniconda
0.0.1
Install

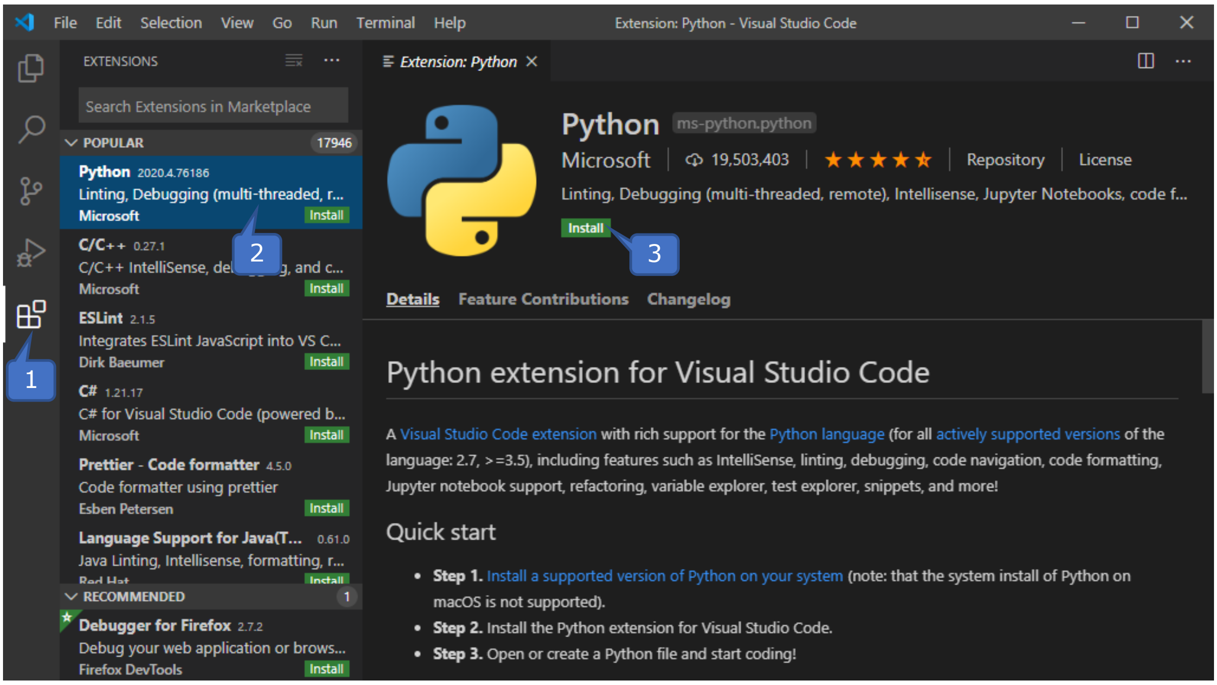
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

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.


File Edit Selection View Go Run Terminal Help
Extension: Python - Visual Studio Code
EXTENSIONS
Search Extensions in Marketplace
POPULAR 17946
Python 2020.4.76186
Linting, Debugging (multi-threaded, r...
Microsoft Install
C/C++ 0.27.1
C/C++ IntelliSense, de... g, and c...
Microsoft Install
ESLint 2.1.5
Integrates ESLint JavaScript into VS C...
Dirk Baeumer Install
C# 1.21.17
C# for Visual Studio Code (powered b...
Microsoft Install
Prettier - Code formatter 4.5.0
Code formatter using prettier
Esben Petersen Install
Language Support for Java(T... 0.61.0
Java Linting, Intellisense, formatting, r...
Red Hat Install
RECOMMENDED 1
Debugger for Firefox 2.7.2
Debug your web application or brows...
Firefox DevTools Install
Extension: Python
Python ms-python.python
Microsoft 19,503,403 ★★★★★ Repository License
Linting, Debugging (multi-threaded, remote), Intellisense, Jupyter Notebooks, code f...
Install
Details Feature Contributions Changelog
1
2
3

Python extension for Visual Studio Code

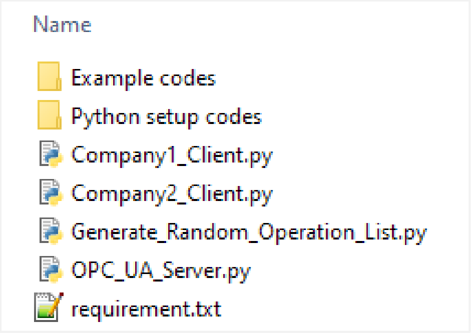
A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: 2.7, >=3.5), including features such as IntelliSense, linting, debugging, code navigation, code formatting, Jupyter notebook support, refactoring, variable explorer, test explorer, snippets, and more!

Quick start

- **Step 1.** Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- **Step 2.** Install the Python extension for Visual Studio Code.
- **Step 3.** Open or create a Python file and start coding!

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,



Name

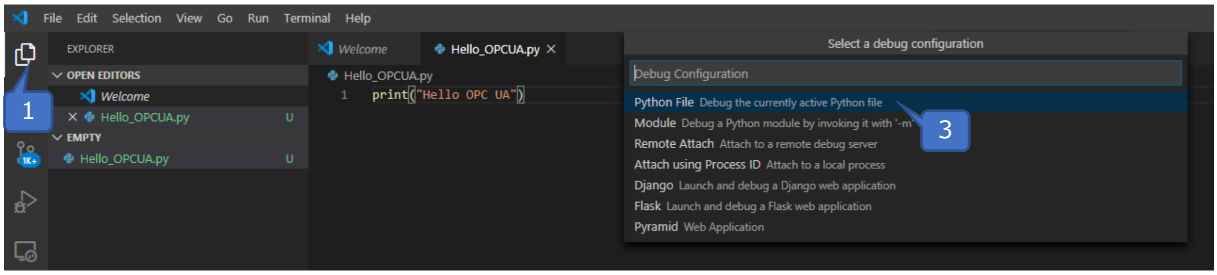
- Example codes
- Python setup codes
- Company1_Client.py
- Company2_Client.py
- Generate_Random_Operation_List.py
- OPC_UA_Server.py
- requirement.txt

<u>(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)</u>

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".

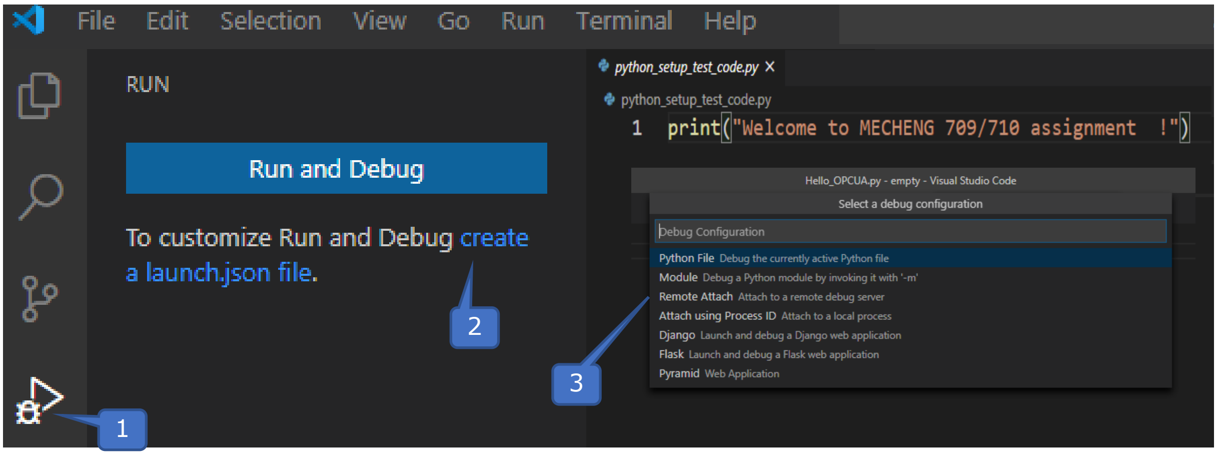


```
print("Hello OPC UA")
```

Select a debug configuration

Debug Configuration

Python File Debug the currently active Python file
Module Debug a Python module by invoking it with '-m'
Remote Attach Attach to a remote debug server
Attach using Process ID Attach to a local process
Django Launch and debug a Django web application
Flask Launch and debug a Flask web application
Pyramid Web Application


File
Edit
Selection
View
Go
Run
Terminal
Help
RUN
Run and Debug
python_setup_test_code.py
python_setup_test_code.py
Hello_OPCUA.py - empty - Visual Studio Code
1
2
3

To customize Run and Debug create a launch.json file.

```
print("Welcome to MECHENG 709/710 assignment !")
```

Select a debug configuration

Debug Configuration

Python File Debug the currently active Python file
Module Debug a Python module by invoking it with '-m'
Remote Attach Attach to a remote debug server
Attach using Process ID Attach to a local process
Django Launch and debug a Django web application
Flask Launch and debug a Flask web application
Pyramid Web Application

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.

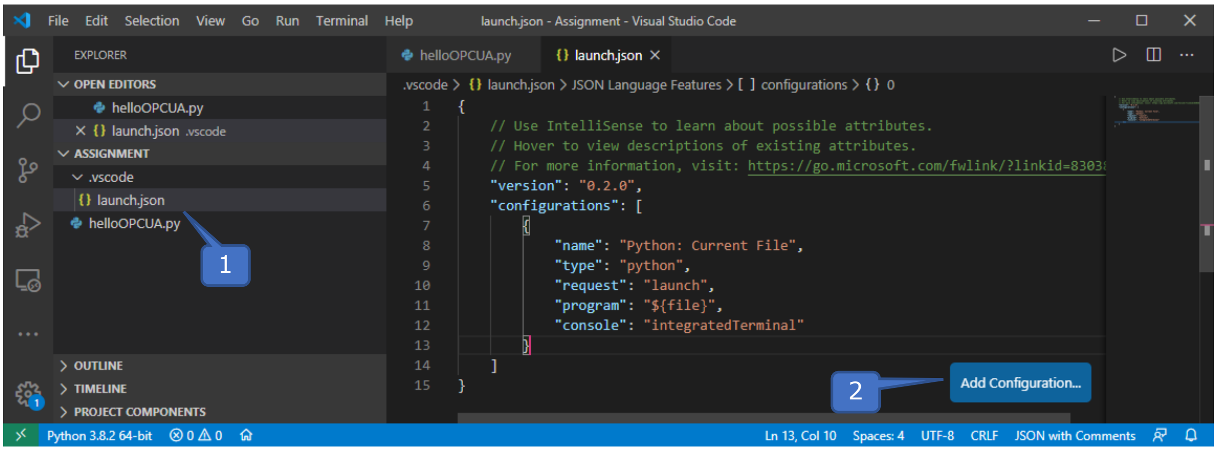
More details can be found in the VS Code documentation. Particularly at:

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

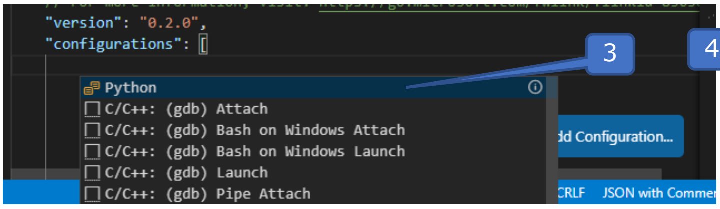
https://code.visualstudio.com/docs/python/debugging#_initialize-configurations.

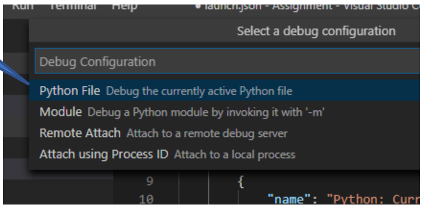
4.3.1. Running two Python scripts at once

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.

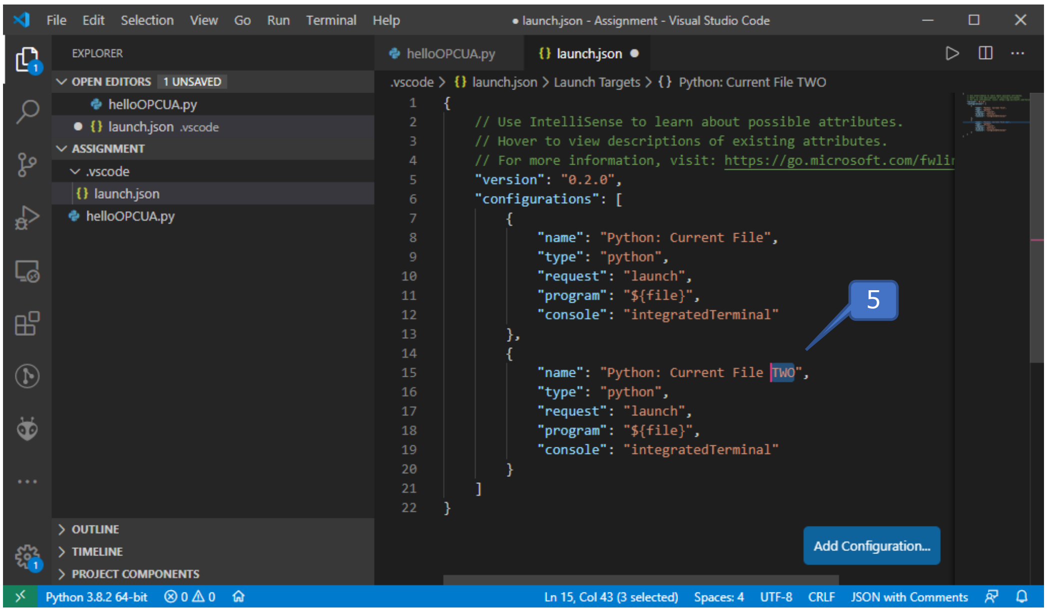

File Edit Selection View Go Run Terminal Help
launch.json - Assignment - Visual Studio Code
EXPLORER
OPEN EDITORS
helloOPCUA.py
launch.json .vscode
ASSIGNMENT
.vscode
launch.json
helloOPCUA.py
1
OUTLINE
TIMELINE
PROJECT COMPONENTS
helloOPCUA.py
launch.json
.vscode > launch.json > JSON Language Features > [ ] configurations > { } 0
Add Configuration...
2
Python 3.8.2 64-bit ⊗ 0 ⚠ 0
Ln 13, Col 10 Spaces: 4 UTF-8 CRLF JSON with Comments

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=8303
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: Current File",
            "type": "python",
            "request": "launch",
            "program": "${file}",
            "console": "integratedTerminal"
        }
    ]
}
```



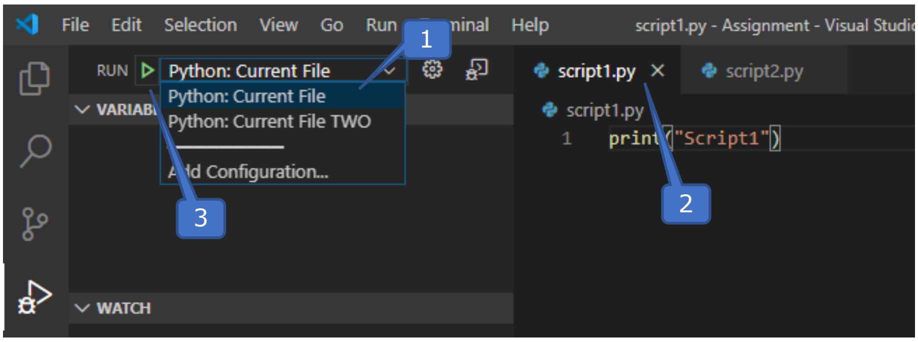

4
Select a debug configuration
Debug Configuration
Python File Debug the currently active Python file
Module Debug a Python module by invoking it with '-m'
Remote Attach Attach to a remote debug server
Attach using Process ID Attach to a local process

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.

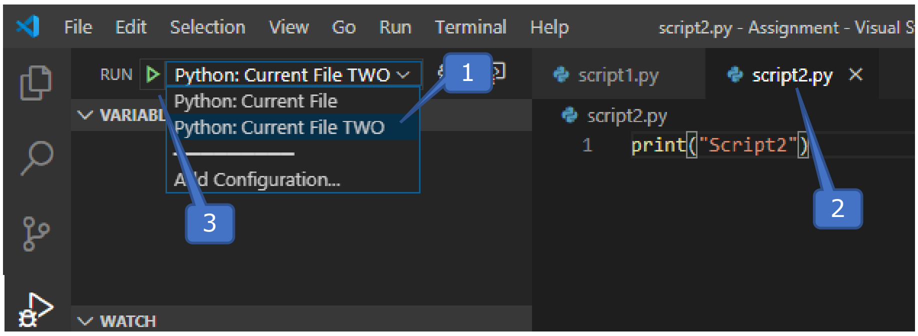

File Edit Selection View Go Run Terminal Help
● launch.json - Assignment - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
helloOPCUA.py
launch.json .vscode
ASSIGNMENT
.vscode
launch.json
helloOPCUA.py
OUTLINE
TIMELINE
PROJECT COMPONENTS
helloOPCUA.py
launch.json
.vscode > launch.json > Launch Targets > { } Python: Current File TWO
5
Add Configuration...
Python 3.8.2 64-bit ⊗ 0 ⚠ 0
Ln 15, Col 43 (3 selected) Spaces: 4 UTF-8 CRLF JSON with Comments

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwli
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Python: Current File",
            "type": "python",
            "request": "launch",
            "program": "${file}",
            "console": "integratedTerminal"
        },
        {
            "name": "Python: Current File TWO",
            "type": "python",
            "request": "launch",
            "program": "${file}",
            "console": "integratedTerminal"
        }
    ]
}
```

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.



```
print("Script1")
```



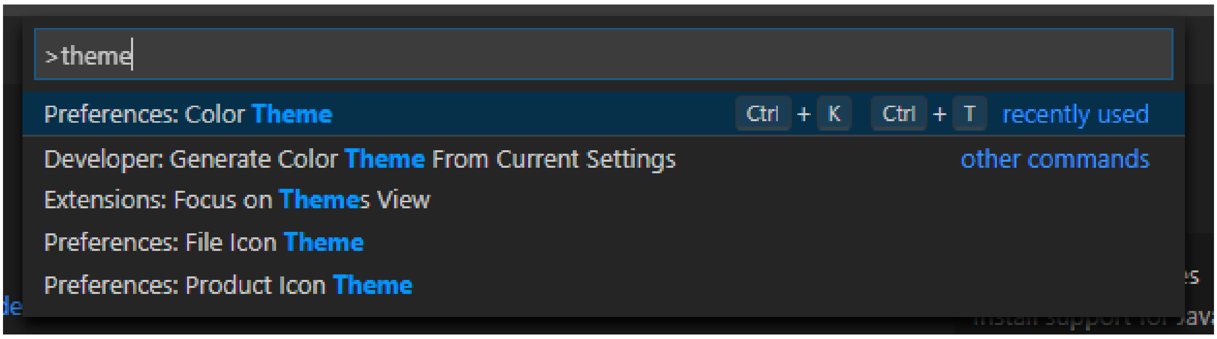
```
print("Script2")
```

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”.



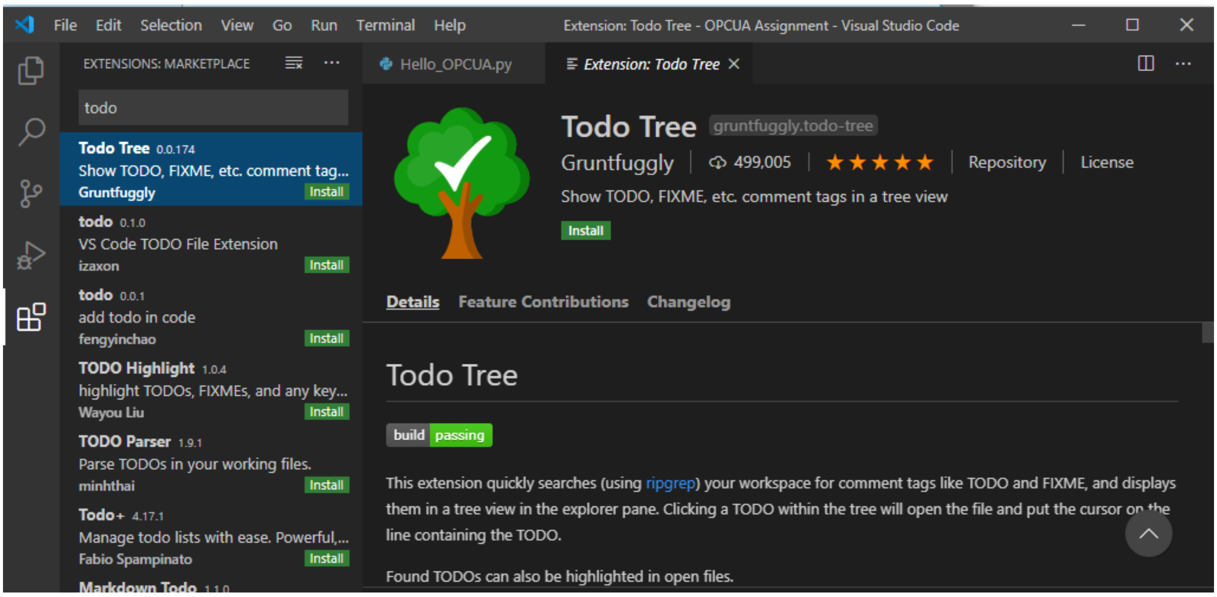
```
>theme
Preferences: Color Theme                                   Ctrl + K  Ctrl + T  recently used
Developer: Generate Color Theme From Current Settings                          other commands
Extensions: Focus on Themes View
Preferences: File Icon Theme
Preferences: Product Icon Theme
```

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.



Todo Tree 0.0.174
Show TODO, FIXME, etc. comment tag...
Gruntfuggly Install

todo 0.1.0
VS Code TODO File Extension
izaxon Install

todo 0.0.1
add todo in code
fengyinchao Install

TODO Highlight 1.0.4
highlight TODOs, FIXMEs, and any key...
Wayou Liu Install

TODO Parser 1.9.1
Parse TODOs in your working files.
minhthai Install

Todo+ 4.17.1
Manage todo lists with ease. Powerful,...
Fabio Spampinato Install

Markdown Todo 1.1.0

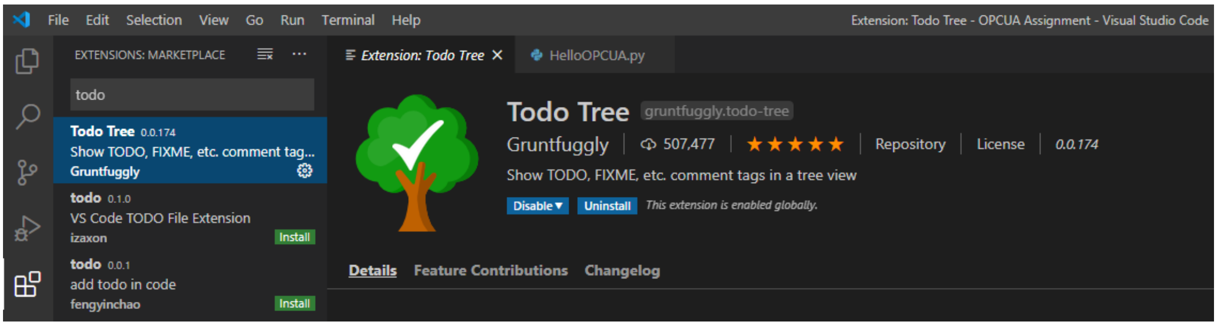
Todo Tree gruntfuggly.todo-tree
Gruntfuggly | 499,005 | ★★★★★ | Repository | License
Show TODO, FIXME, etc. comment tags in a tree view
Install

Todo Tree

This extension quickly searches (using ripgrep) your workspace for comment tags like TODO and FIXME, and displays them in a tree view in the explorer pane. Clicking a TODO within the tree will open the file and put the cursor on the line containing the TODO.

Found TODOs can also be highlighted in open files.

If your computer shows the following, it means that you have got the "Todo Tree" plugin. As of April 2023, we had Version v0.0.226.


File Edit Selection View Go Run Terminal Help
Extension: Todo Tree - OPCUA Assignment - Visual Studio Code
EXTENSIONS: MARKETPLACE
Extension: Todo Tree
HelloOPCUA.py
todo
Details
Feature Contributions
Changelog

Todo Tree 0.0.174
Show TODO, FIXME, etc. comment tag...
Gruntfuggly

todo 0.1.0
VS Code TODO File Extension
izaxon Install

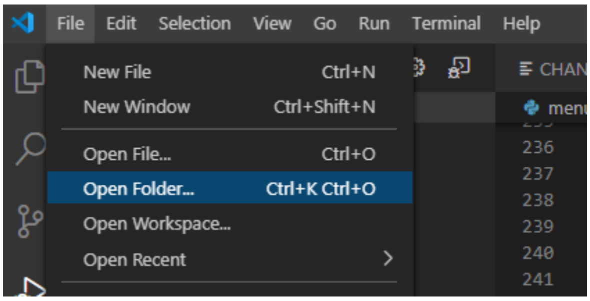
todo 0.0.1
add todo in code
fengyinchao Install

Todo Tree gruntfuggly.todo-tree
Gruntfuggly | 507,477 | ★★★★★ | Repository | License | 0.0.174
Show TODO, FIXME, etc. comment tags in a tree view
Disable ▾ Uninstall This extension is enabled globally.

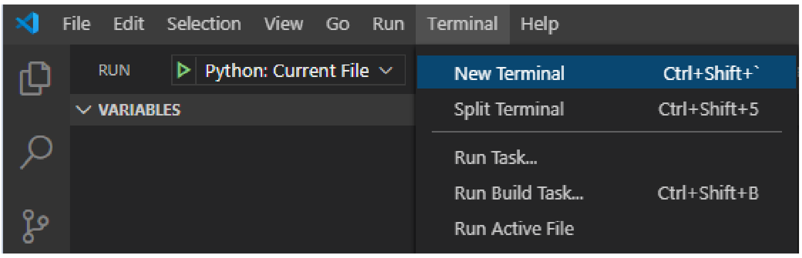
5. Setting up a Virtual Environment

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.

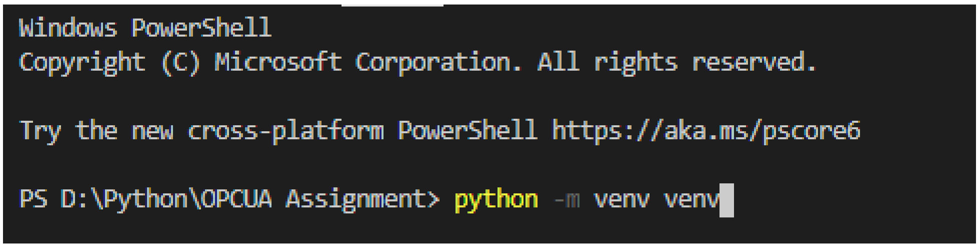
Assume that you have downloaded and extracted your individual group zip file from the CANVAS group section into your assignment folder, e.g. D:\Python\OPC UA Assignment. Open this folder inside VS Code: “File>>Open Folder”.



Next open a new Terminal. “Terminal>>New Terminal”.



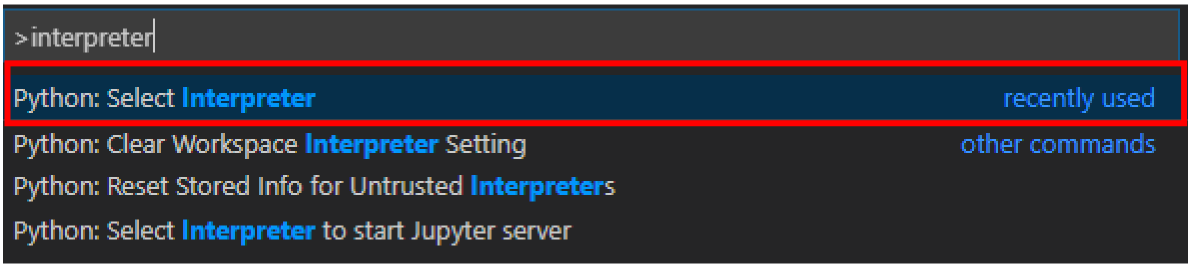
When you are inside your assigment folder, execute the following command: “python -m venv venv”. Make sure to verify your Python installation as outlined in section 1. If you need to use “python3” or “py”, your command will be “python3 -m venv venv” or “py -m venv venv”, respectively. Once the virtual 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.

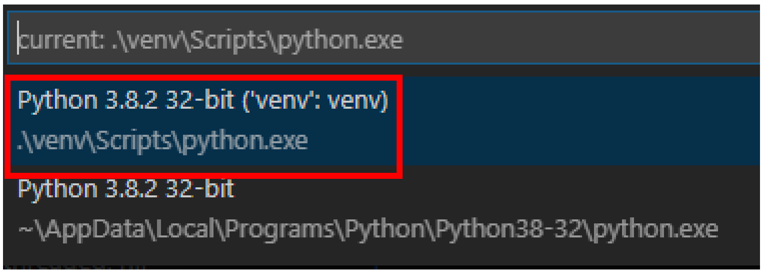


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

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

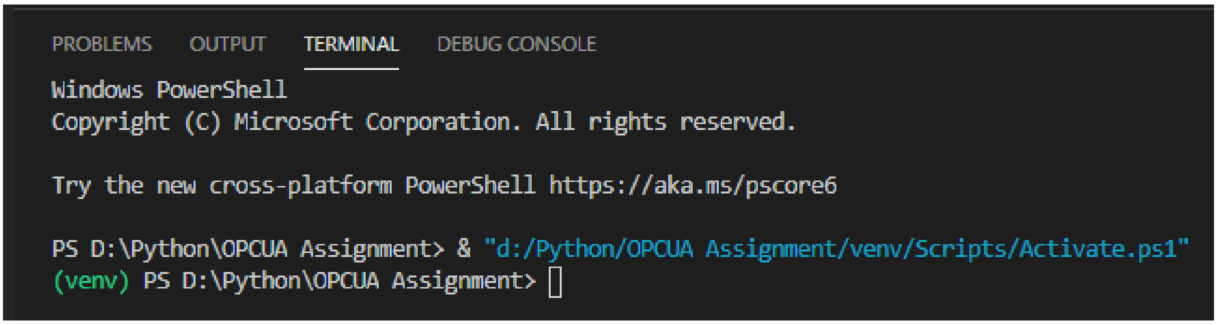
Explanation: By using python with the module (-m) parameter you start the “venv” (virtual environment) module and pass to it the parameter “venv”. The python module “venv” will now create a Virtual Environment in the local folder venv.

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.


>interpreter
Python: Select Interpreter
recently used
Python: Clear Workspace Interpreter Setting
other commands
Python: Reset Stored Info for Untrusted Interpreters
Python: Select Interpreter to start Jupyter server

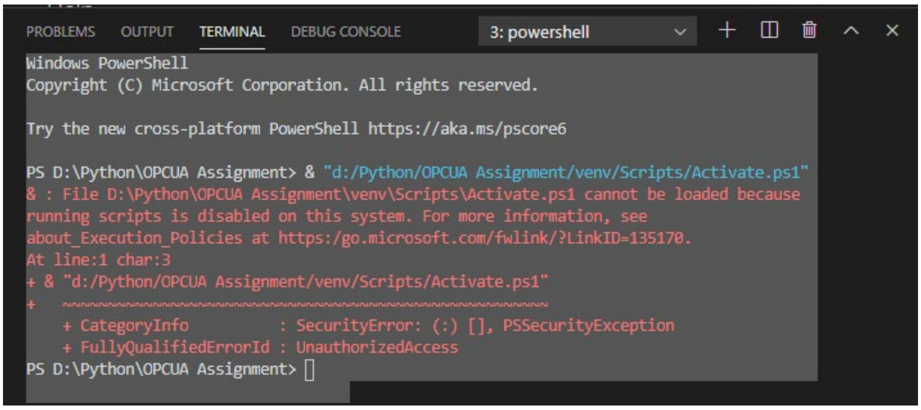

current: .\venv\Scripts\python.exe
Python 3.8.2 32-bit ('venv': venv)
.\venv\Scripts\python.exe
Python 3.8.2 32-bit
~\AppData\Local\Programs\Python\Python38-32\python.exe

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”):



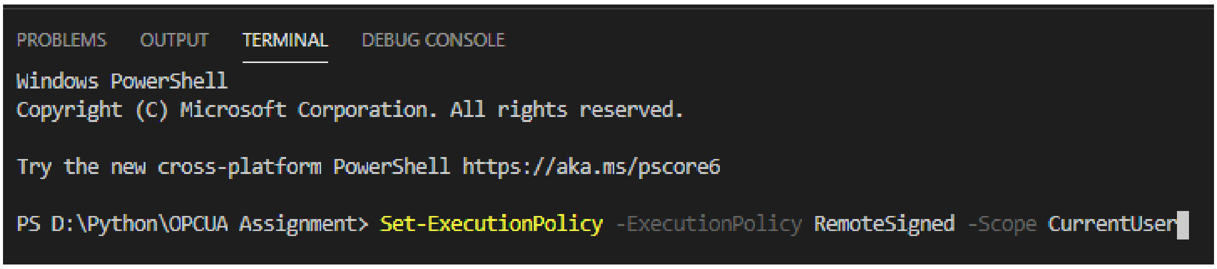
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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>
```

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



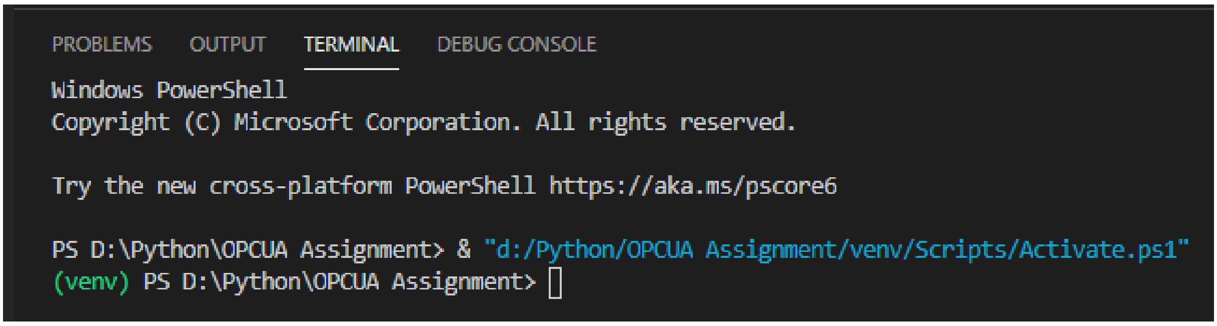
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS D:\Python\OPCUA Assignment> & "d:/Python/OPCUA Assignment/venv/Scripts/Activate.ps1"
& : File D:\Python\OPCUA Assignment\venv\Scripts\Activate.ps1 cannot be loaded because
running scripts is disabled on this system. For more information, see
about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:3
+ & "d:/Python/OPCUA Assignment/venv/Scripts/Activate.ps1"
+   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS D:\Python\OPCUA Assignment>
```

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



```
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 -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

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



```
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.

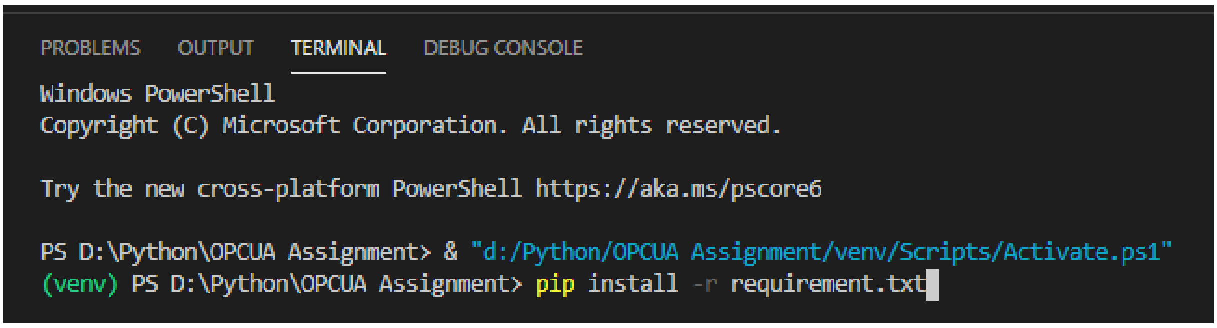
6. Download OPC UA libraries

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".



```
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.