Dr Pouria Sarhadi
School of Physics, Engineering and Computer Science

Python and Visual Studio Code installation to Run Control Functions and Simulation (in 30 minutes):

Python is an open-source and powerful programming language as an alternative to MATLAB. Although Python is comparatively simple and a high-level programming language, MATLAB is easier! especially in control engineering applications. But being open-source and its supportive strong community are advantages of Python. To build an environment like MATLAB in Python, we need to go through some extra installation steps that might seem straightforward for programmers; but not for people who only work with MATLAB. Therefore, building this environment can facilitate the process of control system analysis and design in Python. This document shows how can we build a MATLAB-like environment in Python to analyse and design control systems.

To this end we need some items:

1) Python interpreter;

2) A Graphical User Interface GUI to run Python codes. IDEs (Integrated Development Environments) are GUIs to be used as code editors, compilers or interpreters, and debuggers. Python comes with an IDE called IDLE, but it is not as powerful and interactive as other more professional IDEs. Pycharm, Spyder, and Visual Studio Code (VS Code) are commonly used by programmers. We have selected VS Code as a low size and fast IDE for our core development systems.

3) Python library packages which provide MATLAB-like facilities which can be compared to MATLAB toolboxes.

After setting up those items, we can start our coding in Python. In order to avoid being verbose, it is written in short sentences. The following sections explain the process.

1- Python installation:

Search install Python in google or go to https://www.python.org/downloads/, and download the latest version of Python as shown in the following figure:

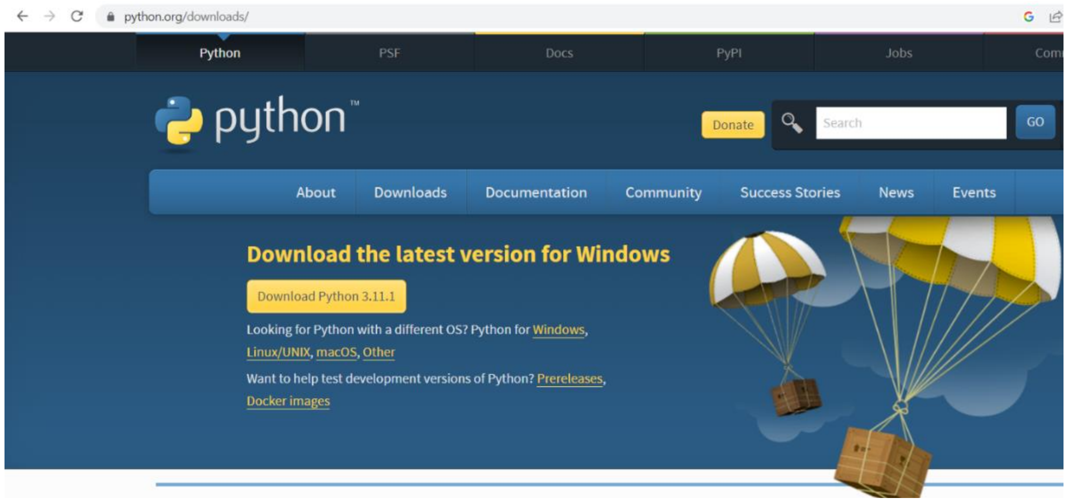
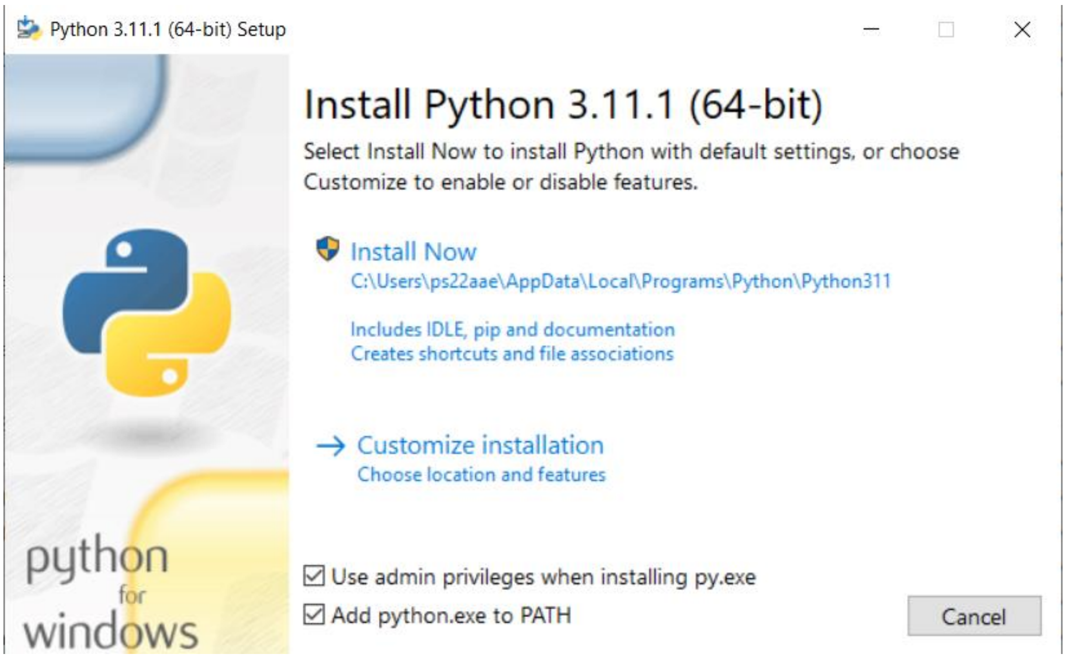

python.org/downloads/
Python
PSF
Docs
PyPI
Jobs
python
Donate
Search
GO
About
Downloads
Documentation
Community
Success Stories
News
Events
Download the latest version for Windows
Download Python 3.11.1
Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other
Want to help test development versions of Python? Prereleases, Docker images

Figure 1- Download Python latest version

Run the downloaded file and install Python by checking “add python.exe to PATH” box as shown in Figure 2. After some seconds, Python will be installed on your computer.



Python 3.11.1 (64-bit) Setup

Install Python 3.11.1 (64-bit)

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

Install Now
C:\Users\ps22aae\AppData\Local\Programs\Python\Python311

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

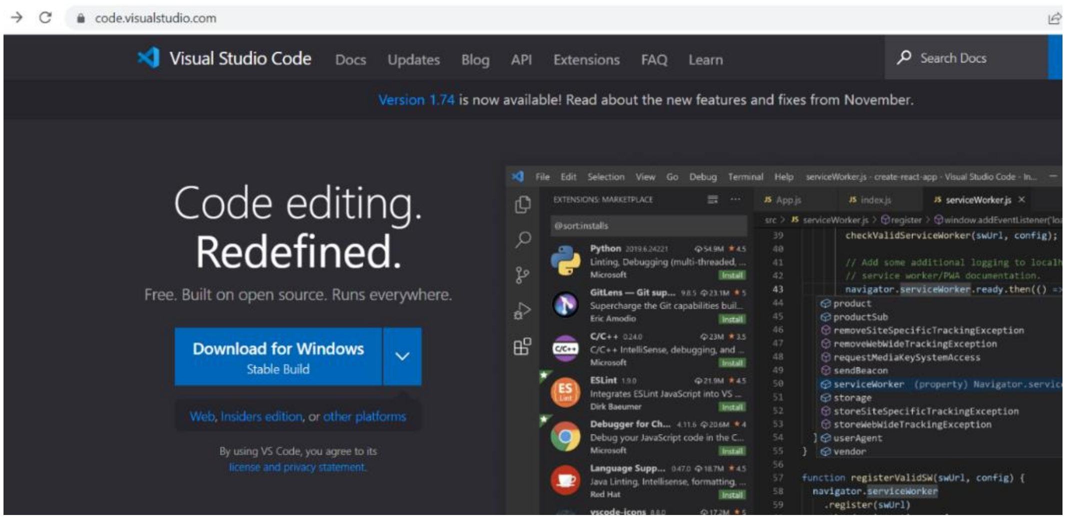
☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

Figure 2- Python installation (don’t forget to tick add to path)

2- Install VS Code

We can google it, or go to the following addresses (https://code.visualstudio.com/download or https://code.visualstudio.com/) as shown in the following figures and download VS Code for your operating system.

*** Note:** This document is only tested for Windows 10.



code.visualstudio.com

Version 1.74 is now available! Read about the new features and fixes from November.

Code editing.
Redefined.

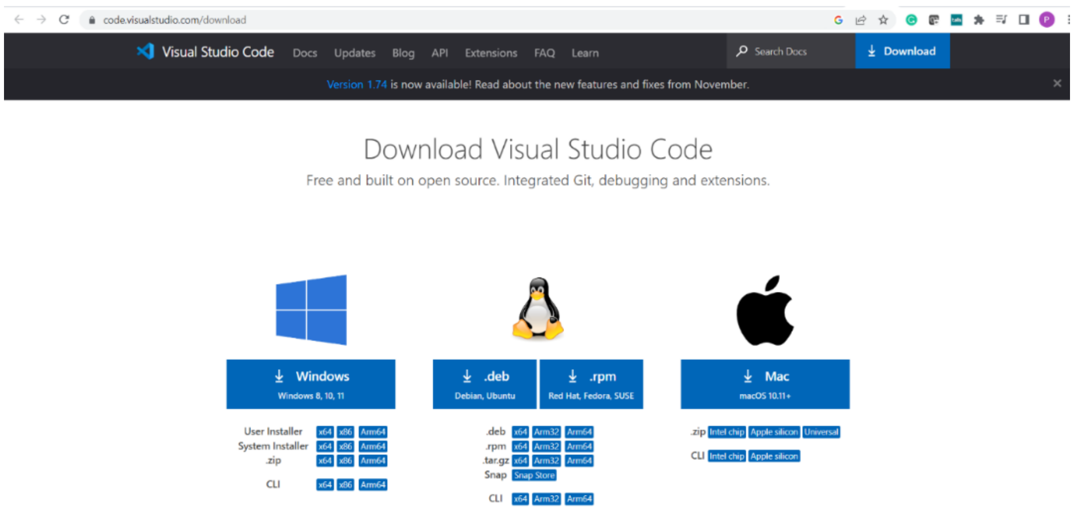
Free. Built on open source. Runs everywhere.

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

```
checkValidServiceWorker(swUrl, config);

// Add some additional logging to localh
// service worker/PWA documentation.
navigator.serviceWorker.ready.then(() =>

function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
```

Figure 3- VS Code auto selection webpage

or:



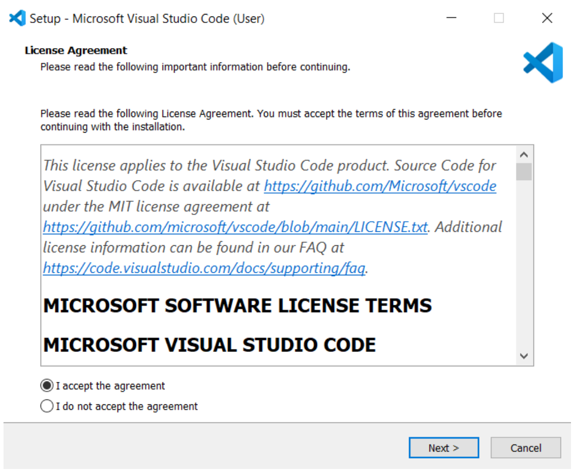
Version 1.74 is now available! Read about the new features and fixes from November.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Figure 4- VS Code download for a specific operating system

After downloading the *.exe file, (might be named VSCodeUserSetup-x64-1.74.3) double click on it and start setup in the following 6 steps and settings:



Setup - Microsoft Visual Studio Code (User)

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

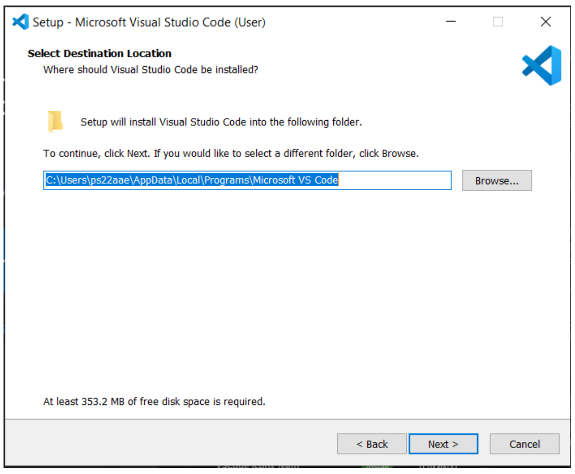
This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

[✓ I accept the agreement] I do not accept the agreement

1



Setup - Microsoft Visual Studio Code (User)

Select Destination Location
Where should Visual Studio Code be installed?

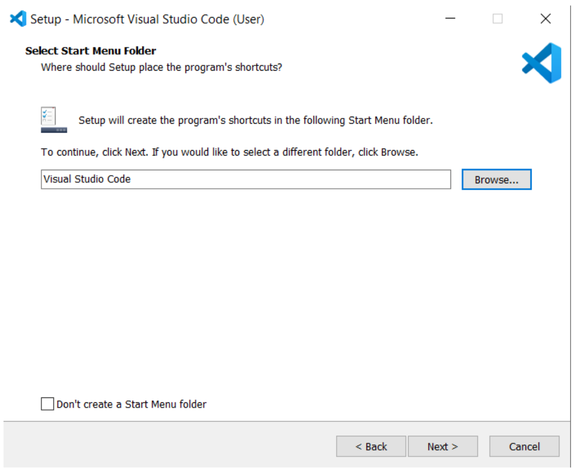
Setup will install Visual Studio Code into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Users\ps22aae\AppData\Local\Programs\Microsoft VS Code

At least 353.2 MB of free disk space is required.

2



Setup - Microsoft Visual Studio Code (User)

Select Start Menu Folder
Where should Setup place the program's shortcuts?

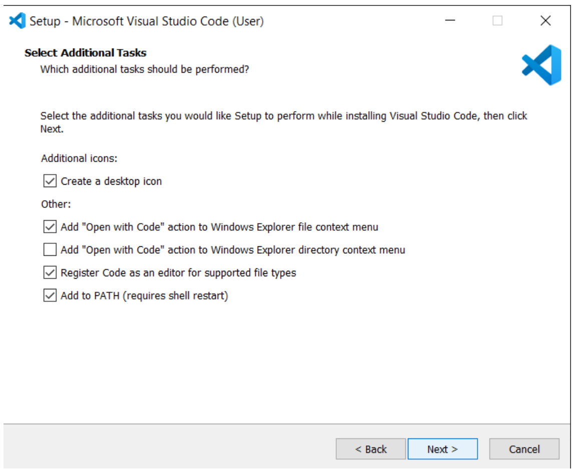
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Visual Studio Code

☐ Don't create a Start Menu folder

3



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

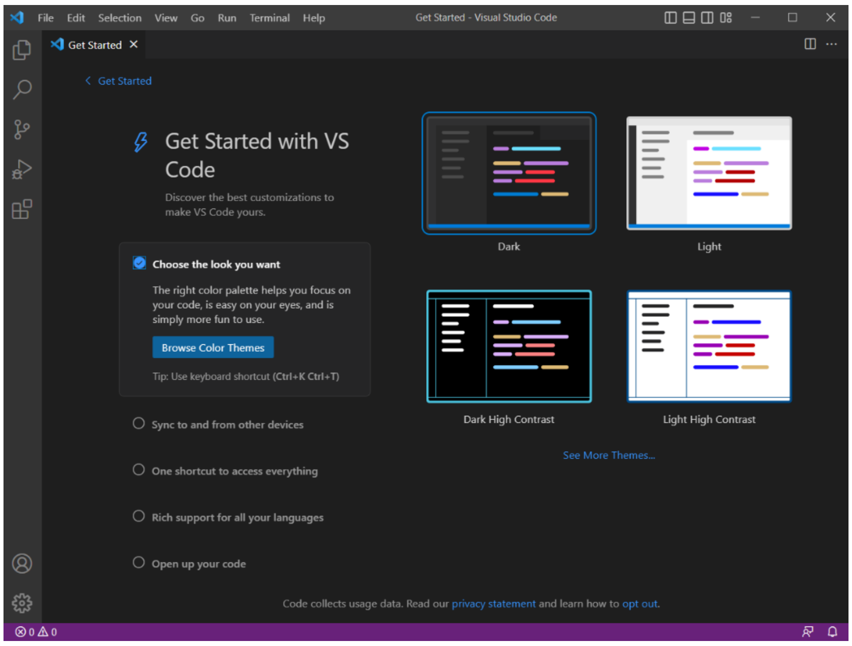
Additional icons:
- ☑ Create a desktop icon

Other:
- ☑ Add "Open with Code" action to Windows Explorer file context menu
- ☐ Add "Open with Code" action to Windows Explorer directory context menu
- ☑ Register Code as an editor for supported file types
- ☑ Add to PATH (requires shell restart)

4

Figure 5- VS Code installation steps

After completion of all steps, VS Code will be installed. Select your custom look and click on get started:


File Edit Selection View Go Run Terminal Help
Get Started - Visual Studio Code
Get Started
Get Started with VS Code
Choose the look you want
Browse Color Themes
Sync to and from other devices
One shortcut to access everything
Rich support for all your languages
Open up your code
Dark
Light
Dark High Contrast
Light High Contrast
See More Themes...

Discover the best customizations to make VS Code yours.

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

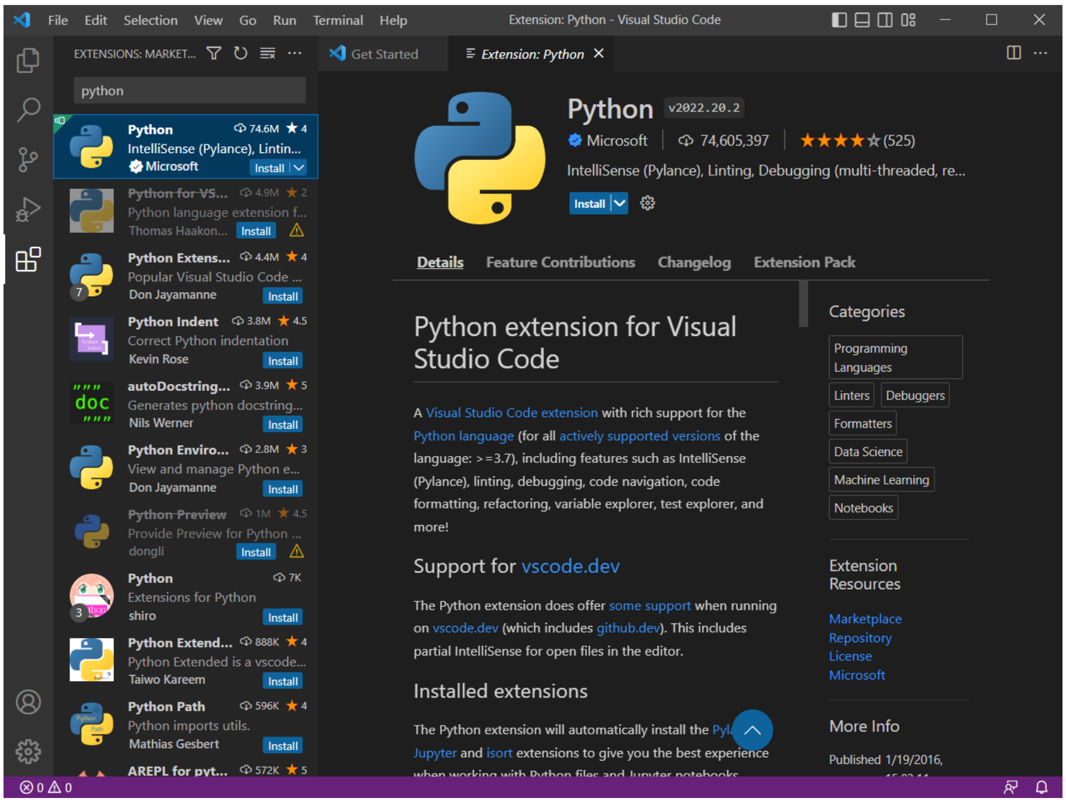
Tip: Use keyboard shortcut (Ctrl+K Ctrl+T)

Code collects usage data. Read our privacy statement and learn how to opt out.

Figure 6- Selecting the look in VS Code (the default dark selected here)

3- Install VS Code Python extension and required Python Libraries

Now the VS Code is ready. However, to run Python in VS Code, we need to install a Python extension. To install this extension, we navigate to the left-hand side “extensions” tab ( ) and search Python. Install the first item (provided by IntelliSense Pylance) with a green tick or corner on its top left as shown in Figure 7.



Python	74.6M	4	IntelliSense (Pylance), Lintin... — Microsoft — Install
~~Python for VS...~~	4.9M	2	Python language extension f... — Thomas Haakon... — Install
Python Extens...	4.4M	4	Popular Visual Studio Code ... — Don Jayamanne — Install
Python Indent	3.8M	4.5	Correct Python indentation — Kevin Rose — Install
autoDocstring...	3.9M	5	Generates python docstring... — Nils Werner — Install
Python Enviro...	2.8M	3	View and manage Python e... — Don Jayamanne — Install
~~Python Preview~~	1M	4.5	Provide Preview for Python ... — dongli — Install
Python	7K		Extensions for Python — shiro — Install
Python Extend...	888K	4	Python Extended is a vscode... — Taiwo Kareem — Install
Python Path	596K	4	Python imports utils. — Mathias Gesbert — Install
AREPL for pyt...	572K	5	

IntelliSense (Pylance), Linting, Debugging (multi-threaded, re...

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: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

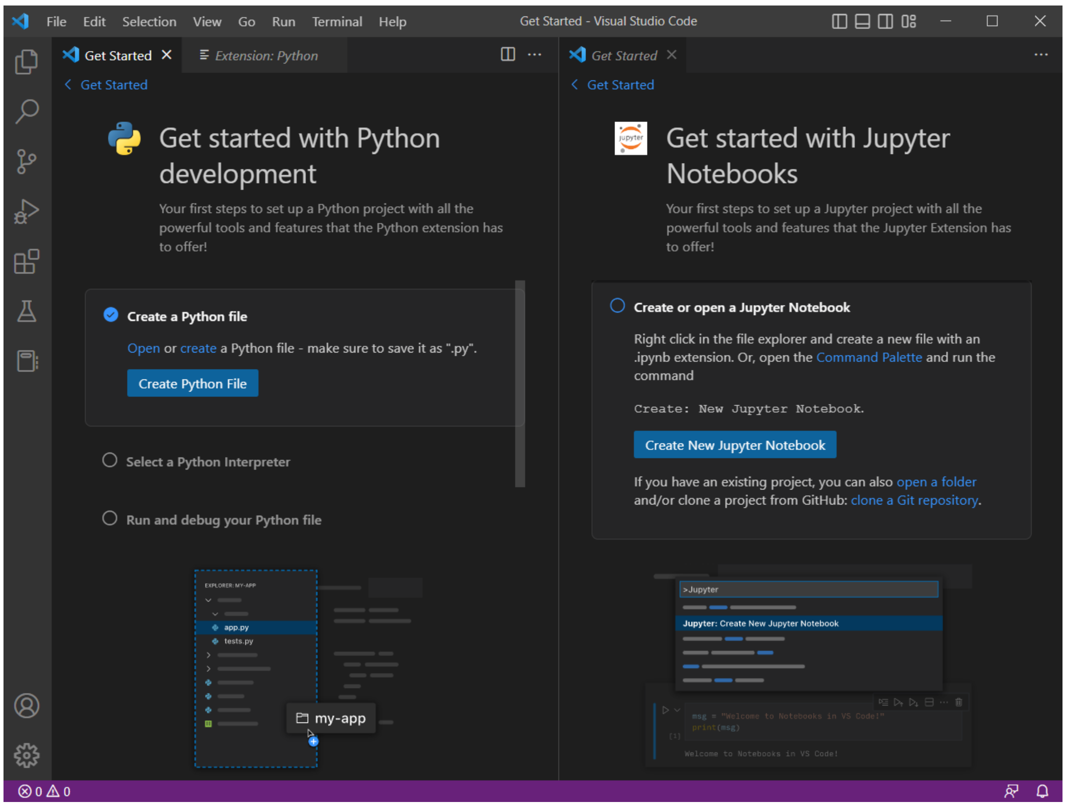
The Python extension does offer some support when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the Pyl... Jupyter and isort extensions to give you the best experience when working with Python files and Jupyter notebooks

Figure 7- Python extension installation in VS Code

Now we have a Python extension but the Python interpreter (that we installed in step 1) should be defined for VS Code. After completing the Python extension installation, the following figure will appear, it is better to select “Create a Python file”.


File
Edit
Selection
View
Go
Run
Terminal
Help
Get Started - Visual Studio Code
Get Started
Extension: Python
Get Started
Create Python File
Select a Python Interpreter
Run and debug your Python file
my-app
Get Started
Create New Jupyter Notebook
>Jupyter
Jupyter: Create New Jupyter Notebook

Get started with Python development

Your first steps to set up a Python project with all the powerful tools and features that the Python extension has to offer!

Create a Python file

Open or create a Python file - make sure to save it as ".py".

Get started with Jupyter Notebooks

Your first steps to set up a Jupyter project with all the powerful tools and features that the Jupyter Extension has to offer!

Create or open a Jupyter Notebook

Right click in the file explorer and create a new file with an .ipynb extension. Or, open the Command Palette and run the command

`Create: New Jupyter Notebook.`

If you have an existing project, you can also open a folder and/or clone a project from GitHub: clone a Git repository.

Figure 8- Select create a Python file to start programming

After these steps, VS Code is ready for programming incorporating Python. Let’s try an example:

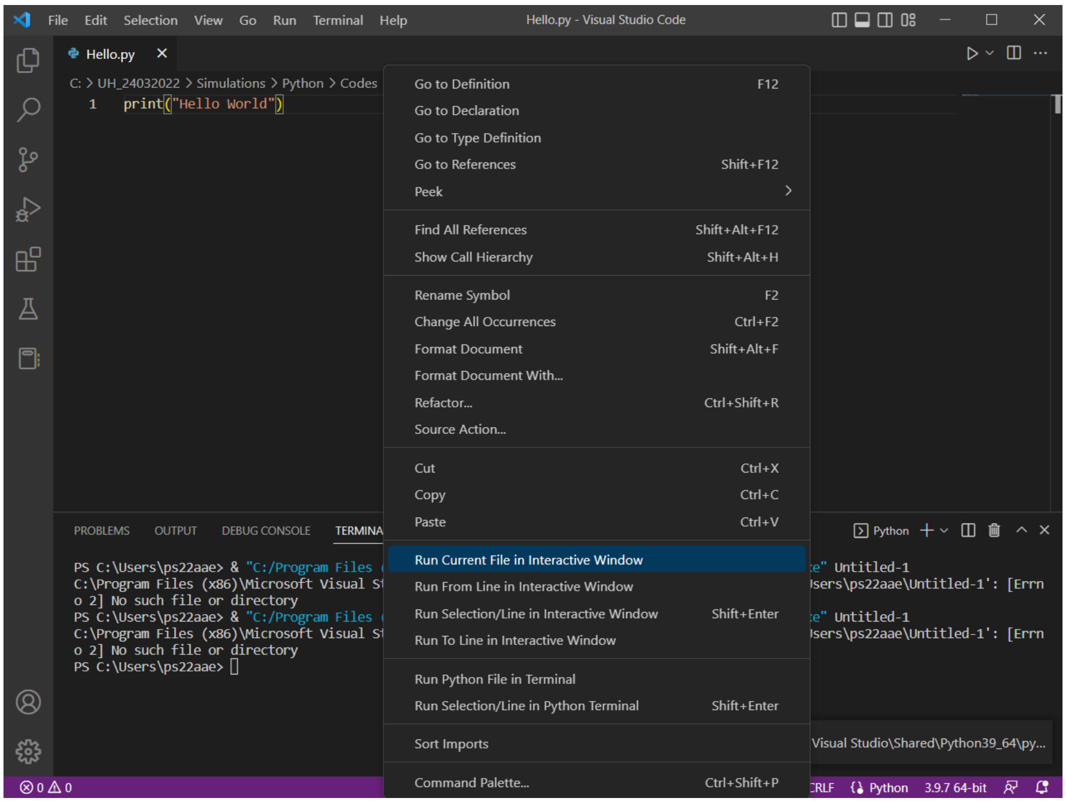
Example 1: Hello World!

In this file, write:

```
print("HelloWorld")
```

Right click and select “Run Current File in In Interactive Window” as shown in Figure 9.

A pop-up will appear and ask/ to install ipykernel, accept and install it as shown in Figure 10.

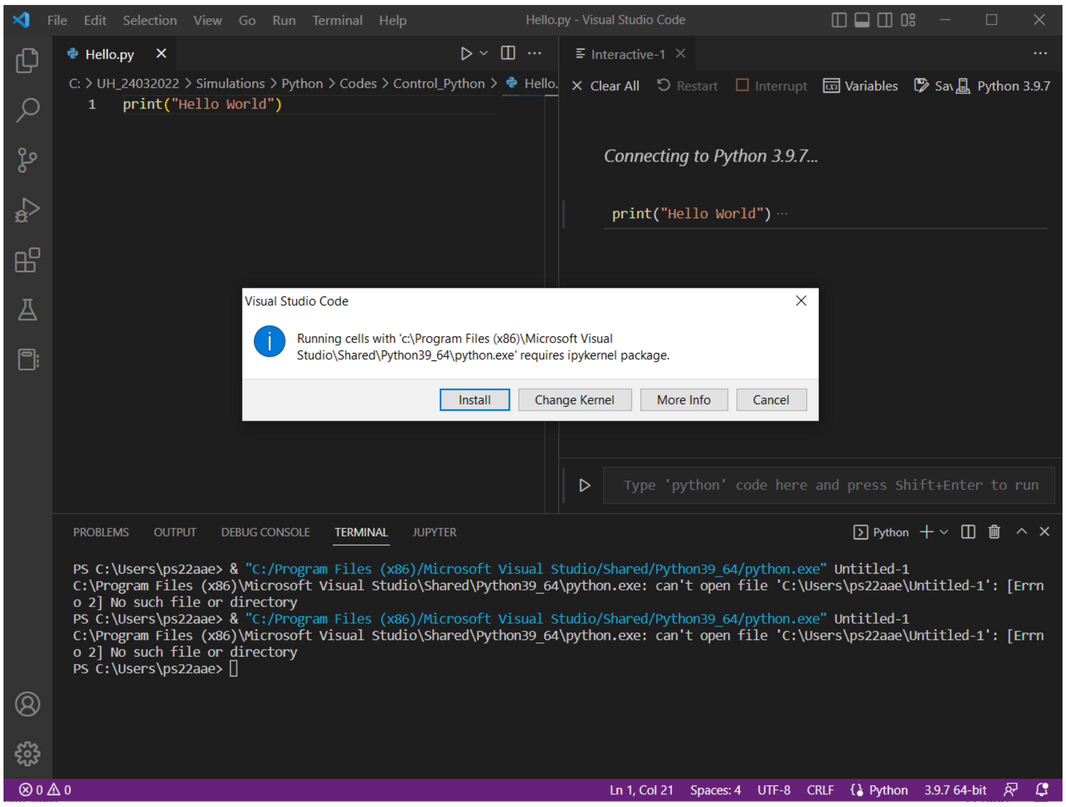

File Edit Selection View Go Run Terminal Help
Hello.py - Visual Studio Code
Hello.py
C: > UH_24032022 > Simulations > Python > Codes
Go to Definition F12
Go to Declaration
Go to Type Definition
Go to References Shift+F12
Peek
Find All References Shift+Alt+F12
Show Call Hierarchy Shift+Alt+H
Rename Symbol F2
Change All Occurrences Ctrl+F2
Format Document Shift+Alt+F
Format Document With...
Refactor... Ctrl+Shift+R
Source Action...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Run Current File in Interactive Window
Run From Line in Interactive Window
Run Selection/Line in Interactive Window Shift+Enter
Run To Line in Interactive Window
Run Python File in Terminal
Run Selection/Line in Python Terminal Shift+Enter
Sort Imports
Command Palette... Ctrl+Shift+P
PROBLEMS OUTPUT DEBUG CONSOLE TERMINA
Python
CRLF Python 3.9.7 64-bit

```
print("Hello World")
```

```
PS C:\Users\ps22aae> & "C:/Program Files (
C:\Program Files (x86)\Microsoft Visual St
o 2] No such file or directory
PS C:\Users\ps22aae> & "C:/Program Files (
C:\Program Files (x86)\Microsoft Visual St
o 2] No such file or directory
PS C:\Users\ps22aae> []
```

```
e" Untitled-1
Jsers\ps22aae\Untitled-1': [Errn
e" Untitled-1
Jsers\ps22aae\Untitled-1': [Errn
Visual Studio\Shared\Python39_64\py...
```

Figure 9- Run a Python file in VS code


File Edit Selection View Go Run Terminal Help
Hello.py - Visual Studio Code
Hello.py
C: > UH_24032022 > Simulations > Python > Codes > Control_Python > Hello.
Interactive-1
Clear All
Restart
Interrupt
Variables
Sav
Python 3.9.7
Visual Studio Code
Install
Change Kernel
More Info
Cancel
Type 'python' code here and press Shift+Enter to run
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python
Ln 1, Col 21 Spaces: 4 UTF-8 CRLF Python 3.9.7 64-bit

```
print("Hello World")
```

Connecting to Python 3.9.7...

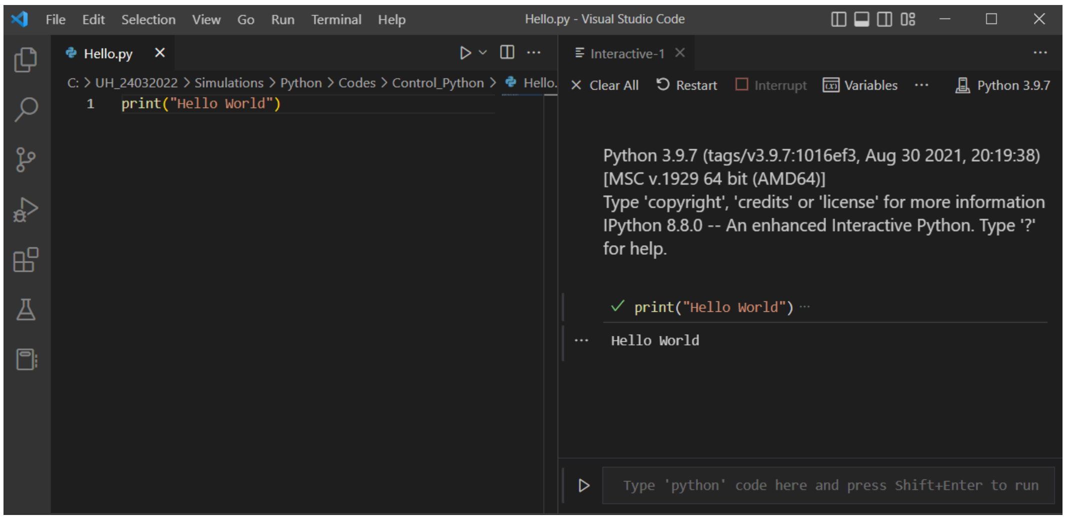
```
print("Hello World")
```

Running cells with 'c:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe' requires ipykernel package.

```
PS C:\Users\ps22aae> & "C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python39_64/python.exe" Untitled-1
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe: can't open file 'C:\Users\ps22aae\Untitled-1': [Errn
o 2] No such file or directory
PS C:\Users\ps22aae> & "C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python39_64/python.exe" Untitled-1
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe: can't open file 'C:\Users\ps22aae\Untitled-1': [Errn
o 2] No such file or directory
PS C:\Users\ps22aae> []
```

Figure 10- ipykernel installation

Here we go! we can observe Hello World on the right-hand side interactive tool as shown in Figure 11.


File Edit Selection View Go Run Terminal Help
Hello.py - Visual Studio Code
Hello.py
Interactive-1
C: > UH_24032022 > Simulations > Python > Codes > Control_Python > Hello.
Clear All
Restart
Interrupt
Variables
Python 3.9.7

```
print("Hello World")
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38)
[MSC v.1929 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.8.0 -- An enhanced Interactive Python. Type '?'
for help.

✓ print("Hello World")
Hello World

Type 'python' code here and press Shift+Enter to run
```

Figure 11- Hello world (Example1) output

Now, we sprightly jump over to package installation.

3- Install Python Libraries

Why do we need to install library packages for Python? As mentioned, even though, Python is one of the easiest programming languages, MATLAB is easier! In MATLAB, there are some toolboxes and functions already installed in the early setup. When we open MATLAB, all those functions are loaded and for example, we can easily define a vector and plot it with the following functions:

A=[1 2 3 4 5]
Plot(A)

In Python, only some basic functions are ready. Therefore, to use a function like "plot" we need to add relevant packages to our code. Fortunately, the Python community has provided lots of effort to mimic Matlab's best tools like plot. Hence, we need to add some extra packages to enable programming. To do this, we need to first install those libraries and import them inside the code. If we want to proceed very quickly, to conduct our daily coding we need to install the following packages in Python:

Matplotlib: a plotting library similar to MATLAB, https://matplotlib.org/.

NumPy: for mathematic functions like handling arrays and matrices, https://numpy.org/.

Control: basic control system functions, https://python-control.readthedocs.io/en/0.9.3.post2/.

Pandas: for array and matrix manipulation.

SciPy: for scientific computing and technical computing like: optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing.

Seaborn: for statistical analysis and data visualisation based on matplotlib

Note: matplotlib, numpy, and control are enough to run most control analysis functions in Python.

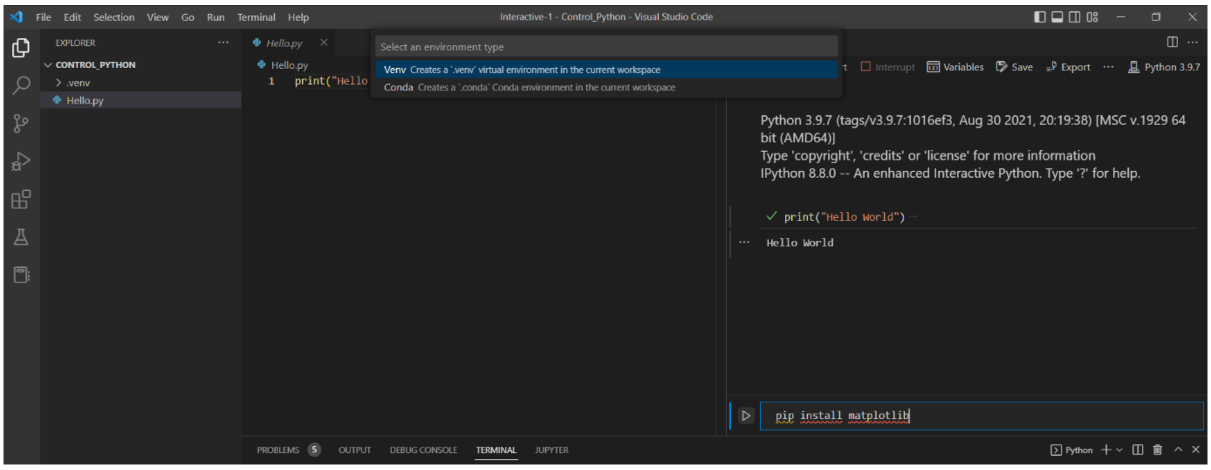
How do we install those libraries?

They could be installed with some simple commands. On the right-hand side window (called interactive window), there is a command executer, just type:

```
pip install matplotlib
```

The process is shown in Figure 12.

and run it with shift+enter or ▷ tab. In some seconds matplotlib will be installed.



```
print("Hello
```

Select an environment type
Venv Creates a `.venv` virtual environment in the current workspace
Conda Creates a `.conda` Conda environment in the current workspace

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64
bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.8.0 -- An enhanced Interactive Python. Type '?' for help.

✓ print("Hello World") ···
··· Hello World

pip install matplotlib
```

Figure 12- Installing Python packages (matplotlib case)

To install other packages, just try the following commands:

```
pip install numpy
```

and:

```
pip install control
```

As well as:

```
pip install scipy
```

After some seconds we should have matplotlib, NumPy, SciPy and Control installed, and we should have access to basic MATAB-like tools.

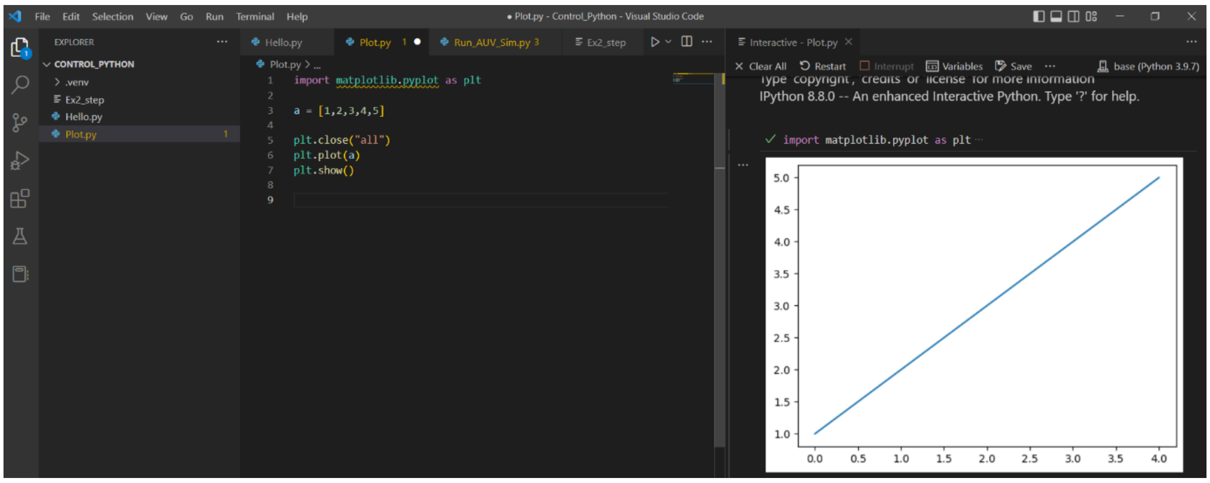
4- Test and run some examples

Example 2: defining a vector and plotting:

Simply try the following functions, right-click in the coding window and select “Run Current File in In Interactive Window”:

```
import matplotlib.pyplot as plt

a = [1,2,3,4,5]
plt.plot(a)
plt.show()
```

It will plot the defined vector as shown in Figure 13.

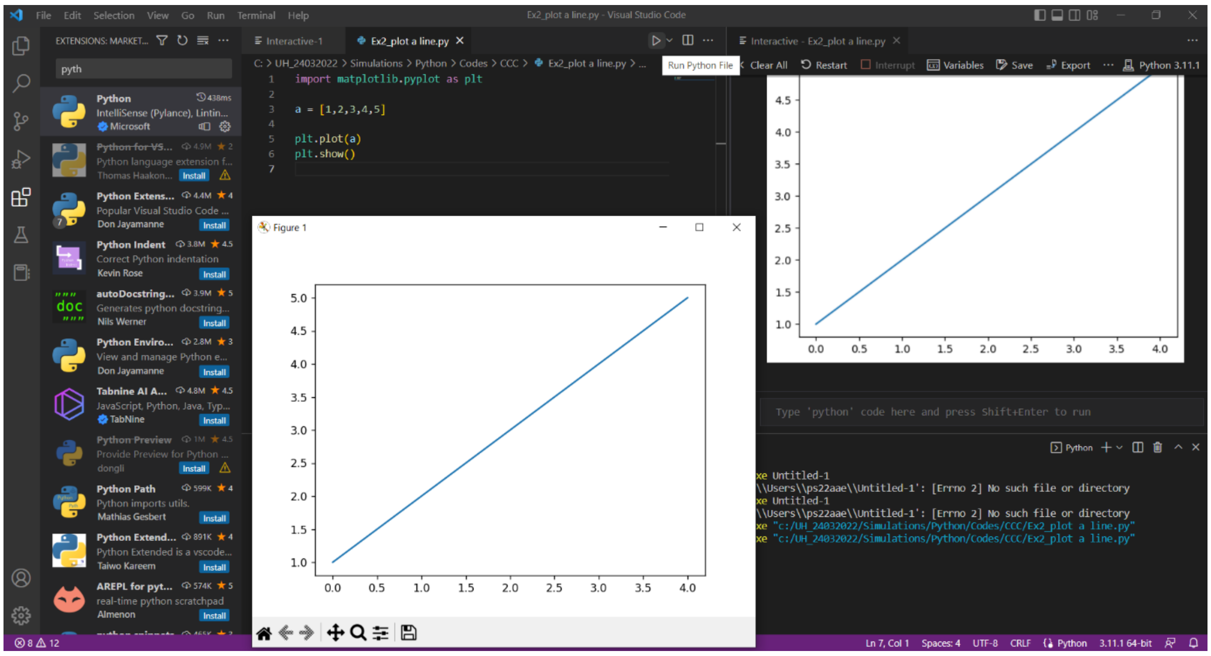


```
import matplotlib.pyplot as plt

a = [1,2,3,4,5]

plt.close("all")
plt.plot(a)
plt.show()
```

```
Type 'copyright', 'credits' or 'license' for more information
IPython 8.8.0 -- An enhanced Interactive Python. Type '?' for help.

✓ import matplotlib.pyplot as plt ···
```

Figure 13- Plotting a vector in the interactive window

Alternatively, one can save the file and try run on top by clicking ▷ as depicted in Figure 14. It will run the code in the terminal and the figure will pop up (as in Figure 14).



```
import matplotlib.pyplot as plt

a = [1,2,3,4,5]

plt.plot(a)
plt.show()
```

```
xe Untitled-1
\\Users\\ps22aae\\Untitled-1': [Errno 2] No such file or directory
xe Untitled-1
\\Users\\ps22aae\\Untitled-1': [Errno 2] No such file or directory
xe "c:/UH_24032022/Simulations/Python/Codes/CCC/Ex2_plot a line.py"
xe "c:/UH_24032022/Simulations/Python/Codes/CCC/Ex2_plot a line.py"
```

Figure 14- Plotting a vector by running the code in the terminal

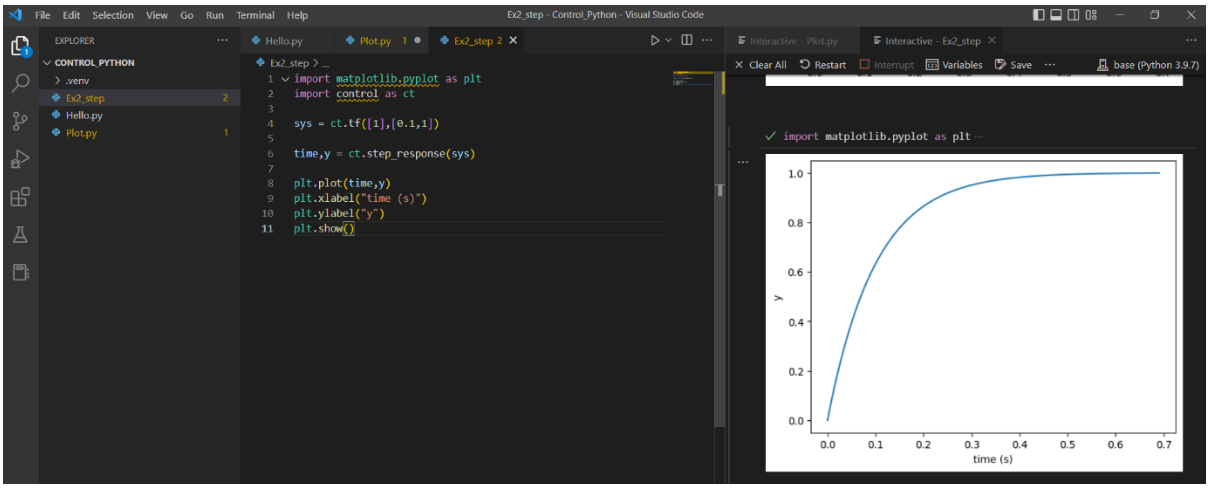
Example 3: defining a system and step plot

Now, we can try a control system simulation. Open a new file, and write the following lines:

```
import matplotlib.pyplot as plt
import control as ct

sys = ct.tf([1],[0.1,1])
```

```
time,y = ct.step_response(sys)

plt.plot(time,y)
plt.xlabel("time (s)")
plt.ylabel("y")
plt.show()
```

Run it with the approaches explained in the previous example, and the result will be as shown in Figure 15:

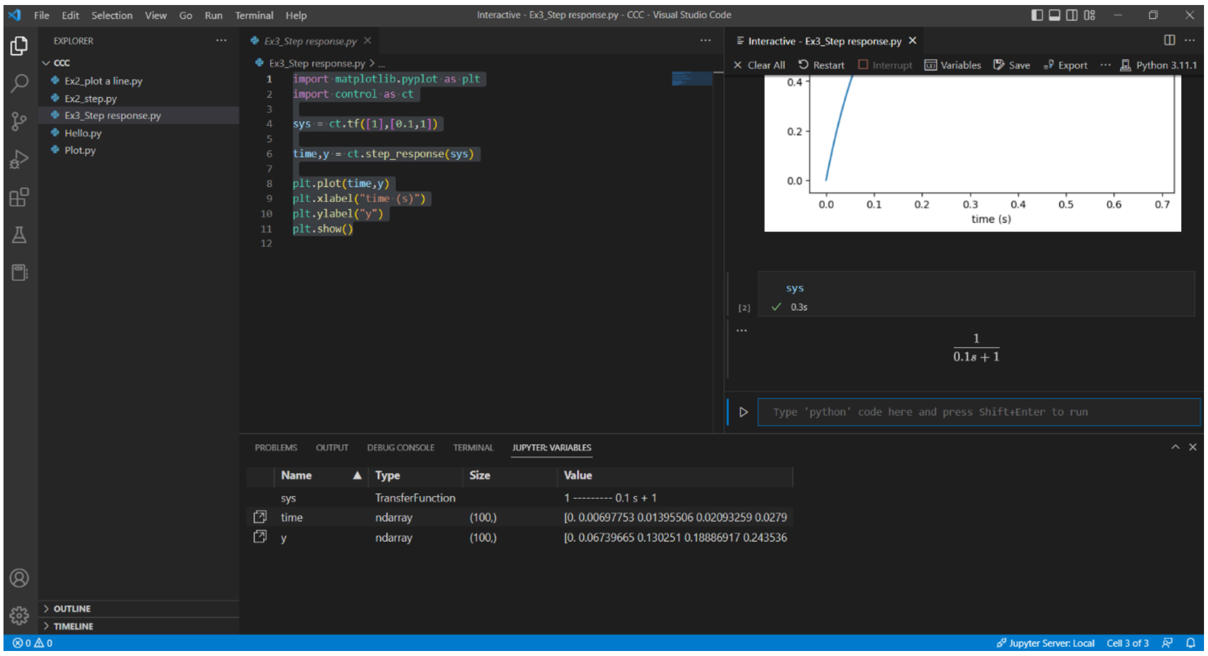


Ex2_step - Control_Python - Visual Studio Code

```
import matplotlib.pyplot as plt
import control as ct

sys = ct.tf([1],[0.1,1])

time,y = ct.step_response(sys)

plt.plot(time,y)
plt.xlabel("time (s)")
plt.ylabel("y")
plt.show()
```

Figure 15- Plotting step response of a system

By clicking on variables on top right, we can see our defined variables (like workspace in MATLAB). In the interactive window, if we type "sys" and run it, we will get the following:



Interactive - Ex3_Step response.py - CCC - Visual Studio Code

```
import matplotlib.pyplot as plt
import control as ct

sys = ct.tf([1],[0.1,1])

time,y = ct.step_response(sys)

plt.plot(time,y)
plt.xlabel("time (s)")
plt.ylabel("y")
plt.show()
```

```
[2]  sys
     0.3s
```

$$\frac{1}{0.1s + 1}$$

JUPYTER: VARIABLES

Name	Type	Size	Value
sys	TransferFunction		1 --------- 0.1 s + 1
time	ndarray	(100,)	[0. 0.00697753 0.01395506 0.02093259 0.0279
y	ndarray	(100,)	[0. 0.06739665 0.130251 0.18886917 0.243536

Figure 16- Defined variables (bottom) and the transfer function illustration (right)

Example 4: linear analysis of the system

We can easily plot root-locus, Bode, Nyquist, Nichols charts or other analysis tools. For instance, this time open a new file and try the following code:

```
import matplotlib.pyplot as plt

import control as ct

sys = ct.tf([0.5,1],[1,0.5,1])

time,y = ct.step_response(sys)


plt.plot(time,y)
plt.xlabel("time (s)")
plt.ylabel("y")
plt.show()

ct.rlocus(sys)
plt.show()


ct.bode_plot(sys)
plt.show()

ct.nyquist_plot(sys)
plt.show()

ct.nichols_plot(sys)
plt.show()
```

It will plot all those mentioned charts. An example of a root locus plotted by Python is illustrated in Figure 17.

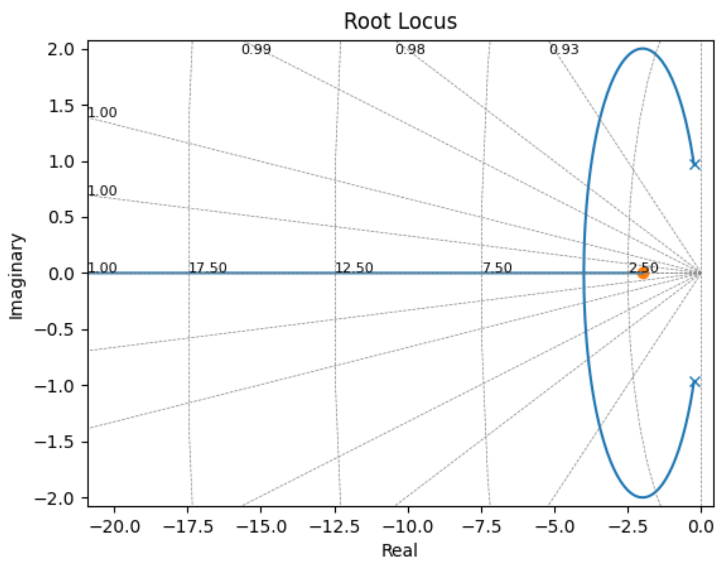


Figure 17- Root Locus plot of a system generated by Python

5- Complete uninstallation of VS Code:

In the early stages of programming, some small changes in IDE or settings could result in errors which are not easy to fix. In this case, my recommendation is to uninstall and re-installing whole software by repeating all steps explained in 1-4 (it does not take more than 30 minutes). However, sometimes uninstallation and deleting VS Code can be challenging since the preferences remain. To completely uninstall and delete all relevant files, the following steps should be undertaken:

1- Go to the folder VS Code installed, run unins000.exe file to uninstall the software.

2- Delete the directory C:\Users\{username}\AppData\Roaming and delete the folder named Code. If you do not see the AppData folder, showing hidden folders should be activated.

3- Similarly, delete the folder C:\Users\{username}\ (you may see 2 and 3 may contain many files).

4- Restart your computer and maybe install VS Code again.