# CREATE VIRTUAL ENVIRONMENT IN WINDOWS 10/11 USING POWERSHELL TERMINAL

1. Create a parent folder where you want to store your Python code and virtual environment, eg. (myProject)
2. Open Powershell Terminal and navigate to this (myProject) folder
3. When you are inside the (myProject) folder within the Powershell Terminal environment, use the Powershell Terminal command:   
      
    python -m venv <NAME OF ENVIRONMENT>  
    example: python -m venv simple
4. In Powershell or in your File Explorer, make sure a new folder for your virtual environment (simple) was added to your parent folder (myProject)
5. Once this runs successfully (it assumes you have [Python](https://www.python.org/) running and the [venv](https://docs.python.org/3/library/venv.html) installed on the computer), you can activate the virtual environment using the following code:

.\<NAME OF ENVIRONMENT\Scripts\Activate.ps1

example: .\simple\Scripts\Activate.ps1

* 1. NOTE: If you get a security ERROR, you need to execute the following command in your Powershell:

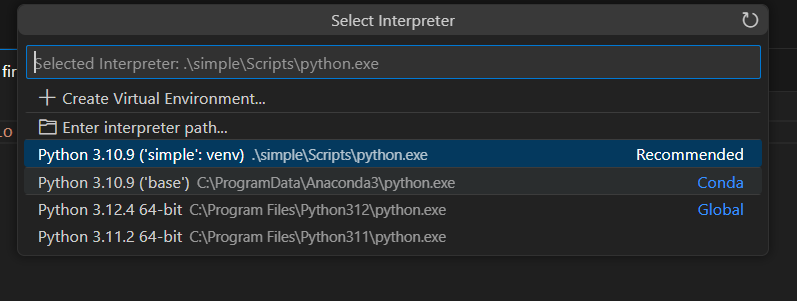
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

After this, now go back to try to ACTIVATE the virtual environment

example: .\simple\Scripts\Activate.ps1

1. Now, in your Powershell, you should see on the far LEFT a (NAME OF ENVIRONMENT) designation as you are interacting with the terminal instead of the typical “PS”; mine now says “(simple) “

# CHOOSE VIRTUAL ENVIRONMENT IN WINDOWS 10/11 USING VS CODE

1. Open [VS Code](https://code.visualstudio.com/)
2. Open the parent folder from the above process in VS Code by choosing **FILE** > **Open Folder** from the VS Code toolbar option at top; in our example, I would open the (myProject) folder
3. Create a new file in this folder and ensure it is a Python file (firstFile.py). Remember, a Python file has a \*.py extension.
4. Open this file (firstFile.py) in VS Code
5. In the Toolbar at the top of the software, click on the **VIEW** option then choose the **COMMAND PALLET** option ---- Note, if you do NOT have a Python file opened in VS Code when you are at this step, you may not see the appropriate command in the COMMAND PALLET
6. In the COMMAND PALLET search box, type: **Python: Select Interpreter**. You will then be presented options for which Python kernel you want to employ (which virtual environment) … I have a few, you may have only two options. Choose your recently created virtual environment from the drop-down list. This will assign whichever Python environment you want to use to VS Code for this session (while you have the program open) and execute ALL your Python code using this environment.
7. Run your code to test.

# INSTALLING NEW MODULES IN YOUR RECENTLY CREATED VIRTUAL ENVIRONMENT IN WINDOWS 10/11 USING VS CODE

1. Open Powershell Terminal and activate your virtual environment (see above)
2. In Powershell environments, you can simply use [pip](https://pip.pypa.io/en/stable/cli/pip_install/):

pip install <NAME OF MODULE>

1. Find appropriate modules by searching online -or- visiting PyPip at <https://pypi.org/>

# USING A REQUIREMENTS.TXT FILE TO LOAD MODULES IN YOUR RECENTLY CREATED VIRTUAL ENVIRONMENT IN WINDOWS 10/11 USING VS CODE

1. Open Powershell Terminal and activate your virtual environment (see above)
2. You must have a requirements.txt file in this directory.
3. The requirements.txt file can list your required modules line by line; you can also specify versions needed for *each* module in the list. NOTE: You do NOT have to list the versions if you don’t need to do so. Examples:

chromadb==0.5.5

huggingface==0.0.1

huggingface-hub==0.24.5

langchain==0.2.14

langchain-community==0.2.12

langchain-core==0.2.32

langchain-text-splitters==0.2.2

langsmith==0.1.99

sentence-transformers==3.0.1

transformers==4.44.0

1. In Powershell, you can simply use [pip](https://pip.pypa.io/en/stable/cli/pip_install/):

pip install --upgrade -r requirements.txt

## NOTE: You can FREEZE a PROJECT USING PIP so that you can automatically create a “requirements.txt” for your project from the virtual environment…

pip freeze > requirements.txt

This will create a “requirements.txt” file for you in the directory. You can look at it to see ALL the modules installed in your Python virtual environment. NOTE – these will show all the base modules -and- modules you have installed.