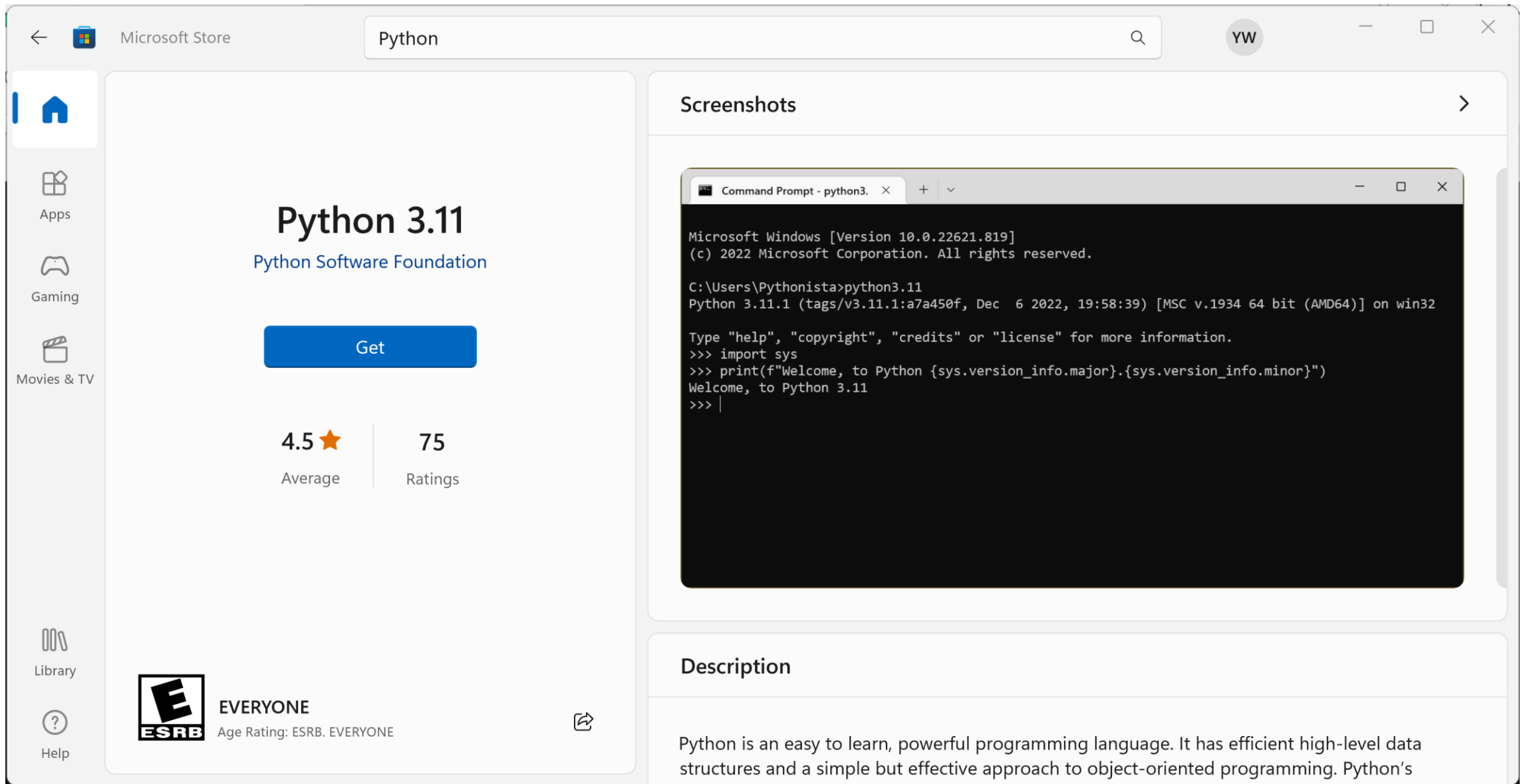


CLI vs GUI

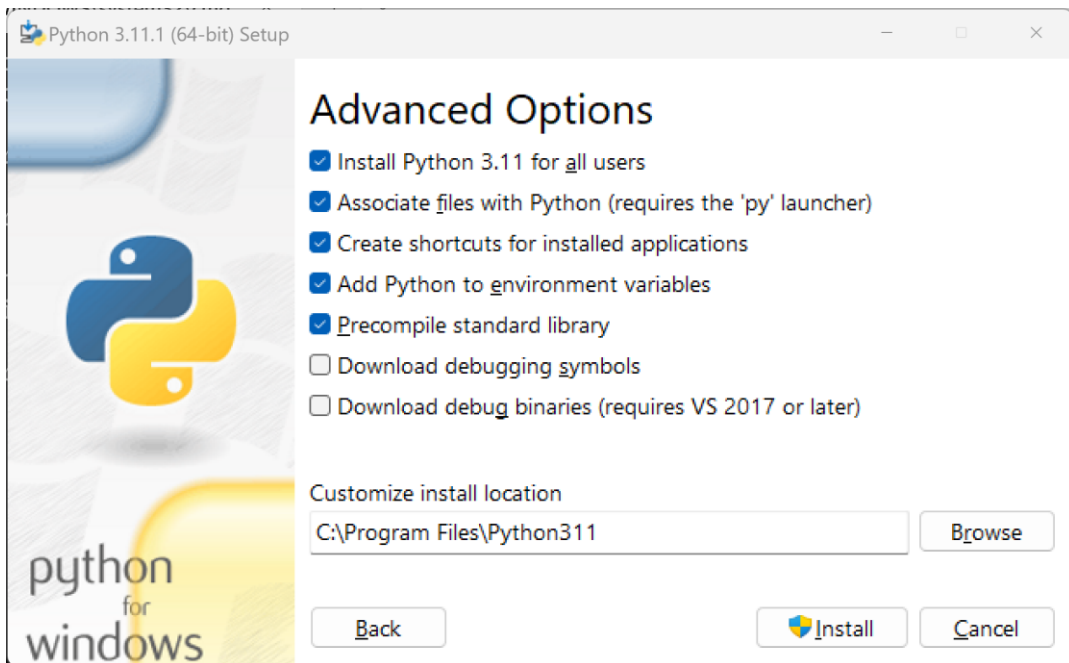
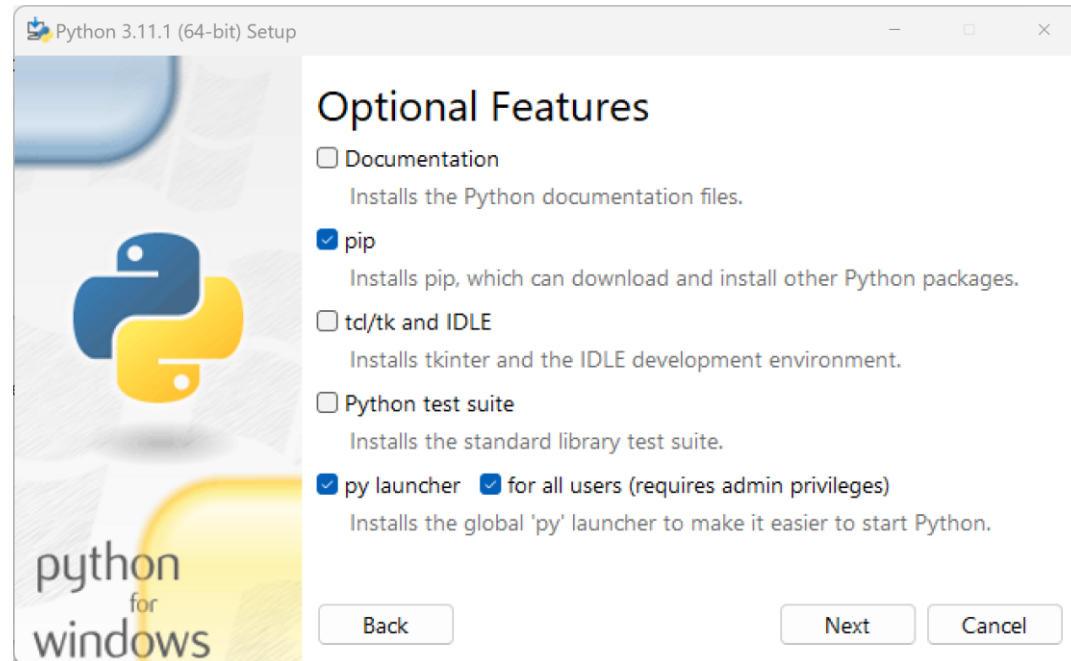
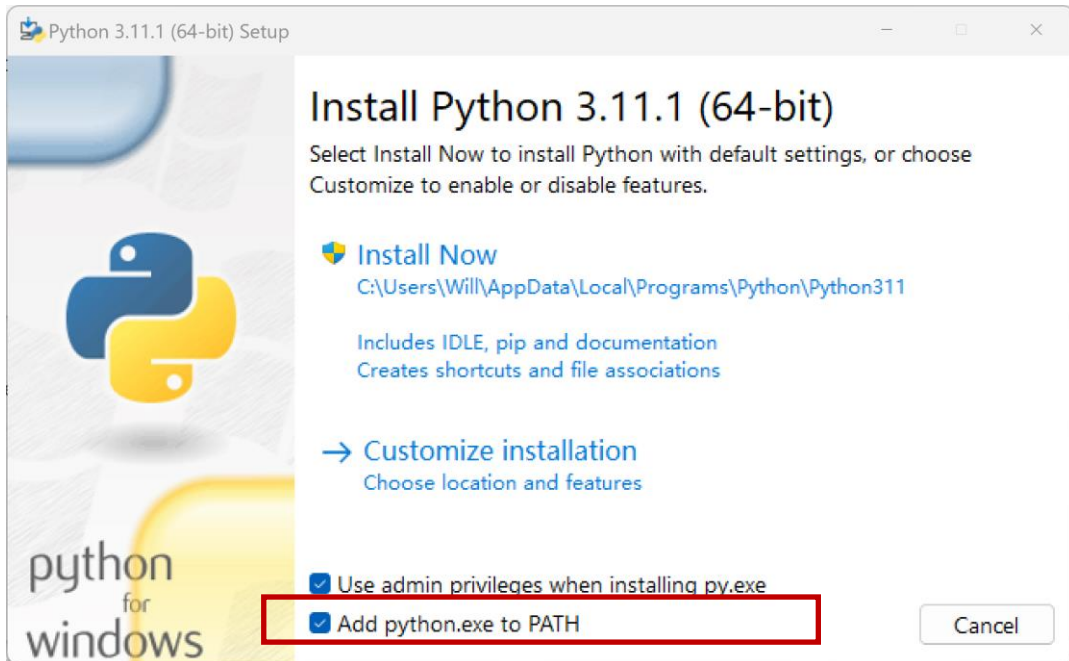
- CLI: command line interface
 - Traditional, old school, back in 1960s
 - But it's required in all programming work and in this course
 - On Windows: CMD / PowerShell / Windows Terminal / WSL
 - On MacOS: Terminal
- GUI: graphical user interface
 - Modern, since 1980s: Apple Lisa / Windows 1.0

How to install Python

- Windows:
 - You may already install it.
 - Windows store. If you type python in command line on a computer without Python installed, a windows store installation page should pop up.
 - Official Installer from python.org
 - Using [conda](https://conda.io). Only recommend if you already know it. It's more complex
- MacOS:
 - It's a built-in function.
 - You may install again using the official installer or conda, but NOT recommended:
 - It will lead to 2 Python on your computer, just creating trouble for yourself.

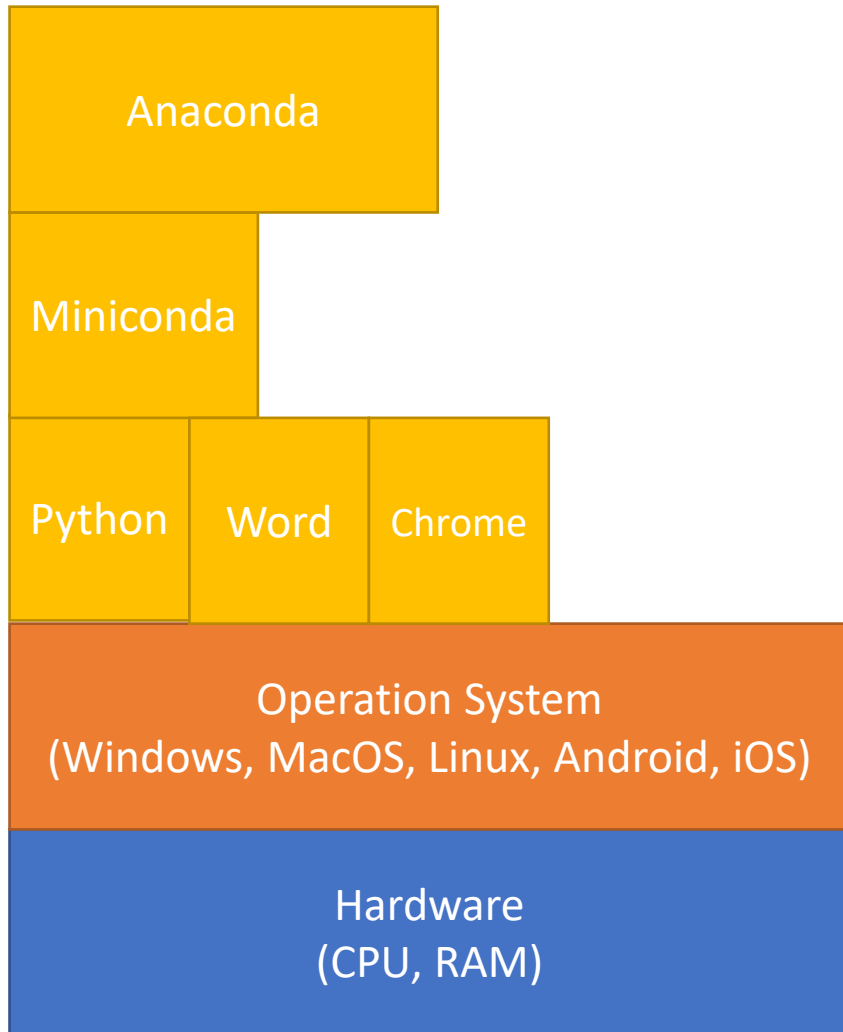


Python on Windows Store



Recommended Setting for Python Installer
Select: Add python.exe to PATH

What's the difference between Python/Miniconda/ Anaconda



Python: Python is an application running on top of operation system. You could download it from python.org. The installer size is about 28MB.

Miniconda: Miniconda is Python + package manager (AppStore, Google Play). It allow you to conveniently install other python libraries. It could be downloaded from [here](#). The installer size is about 72MB.

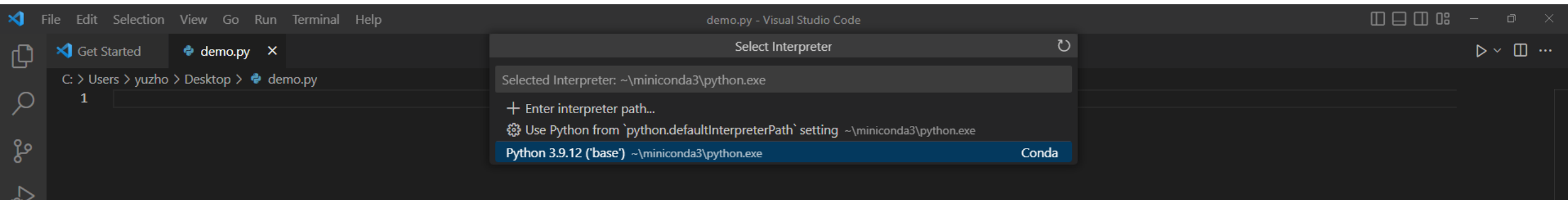
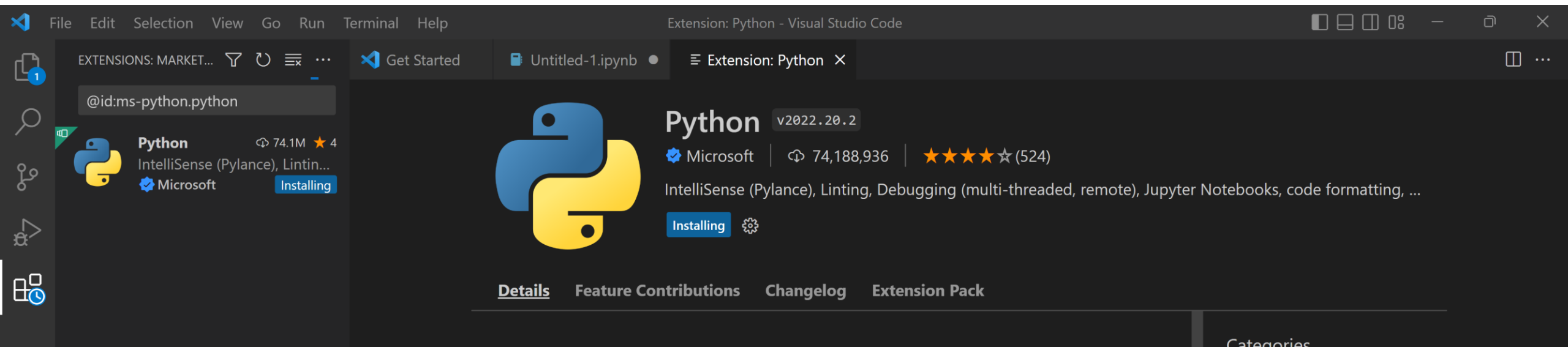
Anaconda: Anaconda is Miniconda with all the common packages installed. It could be downloaded from [here](#). The installer size is about 600MB.

Conclusion:

1. Either one is sufficient for the course. If your system already have built-in Python (MacOS, Linux), just use the system provided Python.

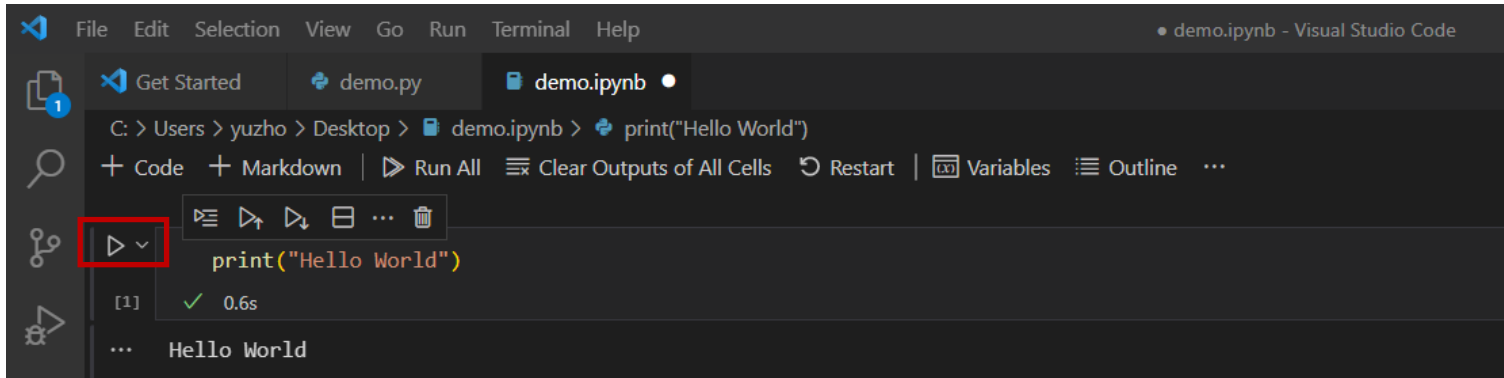
How to Install VS Code

1. Just download & install. It's free and open source. [Link](#)
 1. Open you open/create the first Python file, it will prompt you to install Python extension. Click yes to install.
 2. Alternatively, search for Python extension in the Extension page.
 3. Remember to select Python interpreter. It will prompt automatically.
2. Why VS Code?
 1. It support open Jupyter notebook file
 2. Have a lot of extensions to help you (to be explored later)
 3. It has been used in many tech companies: Meta/Google.
 4. Alternative: Sublime Text / PyCharm



How to run your code?

- In Jupyter notebook:
 - Just click the run arrow



- In Python file:
 - In VS code: Run->Run without Debugging
 - (Recommended) In command line:
 - 1. Change the directory to where your Python file is saved
 - 2. Run command: `python filename.py`

What's the difference between Python Shell / Editor / Jupyter Notebook

Python REPL Shell:

- REPL stands for: Read, Evaluate, Print, Loop
- Good for simple tasks

```
(base) PS C:\Users\yuzho> python
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+1
2
>>> |
```

Code Editor:

- First write code, then run code: `python <filename>.py`
- Good for complex tasks: multiple files / functions / classes

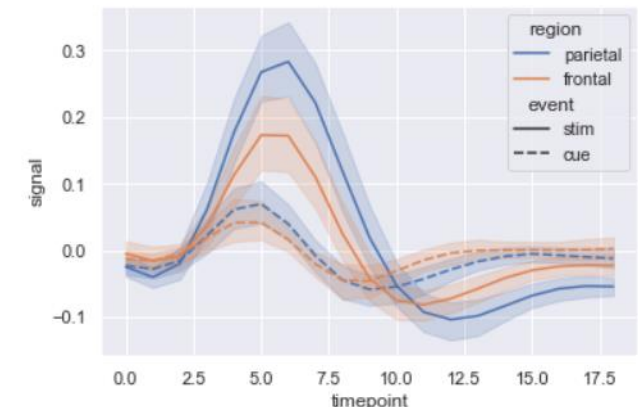
```
for fn_name, tests in all_tests.items():
    fn = getattr(solution, fn_name)
    for args, expected_outputs in tests.items():
        actual_outputs = fn(*args)
        if not isinstance(actual_outputs, tuple):
            actual_outputs = (actual_outputs, )

        assert len(actual_outputs) == len(expected_outputs)
        matches = [abs(actual - expected) < epsilon for actual, expected in zip(actual_outputs, expected_outputs)]
        if all(matches):
            score += each_score
        else:
            print(f"Found Error: Your code output {fn_name} is incorrect")

print(f"Your score for Lab1 Assignment: {score:.0f}")
```

Jupyter Notebook:

- A combination of both
- Good for experiments, analysis, visualization



What's the benefit of using a code editor?

1. Syntax Highlight
2. Auto completion
3. Auto formatting
4. Auto Lint

```
(for fn_name, tests in all_tests.items():  
    fn = getattr(solution, fn_name)  
    for args, expected_outputs in tests.items():  
        actual_outputs = fn(*args)  
        if not isinstance(actual_outputs, tuple):  
            actual_outputs = (actual_outputs, )  
  
        assert len(actual_outputs) == len(expected_outputs)  
        matches = [abs(actual - expected) < epsilon for actual, expected in zip(actual_outputs, expected_outputs)]  
        if all(matches):  
            score += each_score  
        else:  
            print(f"Found Error: Your code output {fn_name}{args} = {actual_outputs}. Expected output: {expected_outputs}")  
  
print(f"Your score for Lab1 Assignment: {score:.0f}")
```

How to install & launch Jupyter notebook

Recommend: Install [VS Code](#) and you could open Jupyter notebook directly.

Alternative: Install Jupyter notebook python package

- not recommend unless you really want, it's complex
- Due to different system configuration, use one of the command below

Install Jupyter notebook:

```
conda install jupyter
pip install jupyter
pip3 install jupyter
python -m pip install jupyter
python3 -m pip install jupyter
```

Launch Jupyter notebook:

```
jupyter notebook
python -m jupyter notebook
python3 -m jupyter notebook
```