

Tutorial 1 – Week 1

We'll be starting at the 10 minute mark

Agenda

1. TA Introductions
2. Logistics
3. Install Anaconda Navigator
4. Set up a Folder Structure for APS106
5. VSCode/Jupyter Notebook
6. UofT JupyterHub
7. Questions?

Introduction – TA



Minori Narita

(TUT02, TUT05)

Current studies: 4th year PhD MIE Student

Research/other interests: Machine Learning, Scheduling Problems, Basketball

Tutorial Logistics

- Tutorials are for your benefit – **no grading**
 - We will review previous weeks labs & lecture content
 - From Week 2, we will focus on **coding problems!**
- Be sure to **ask lots of questions** and have Python open.
We are here to help you!
- Questions outside of tutorial time?
 - Post to **Piazza** – all TAs/instructors and your peers can answer questions quickly
 - **Coffee Time** – drop-in hours for 1on1 help

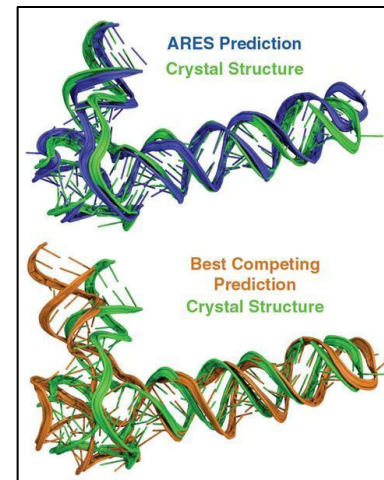
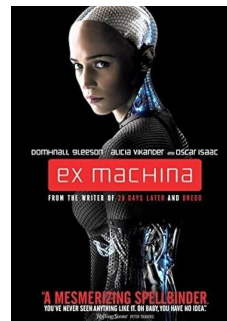
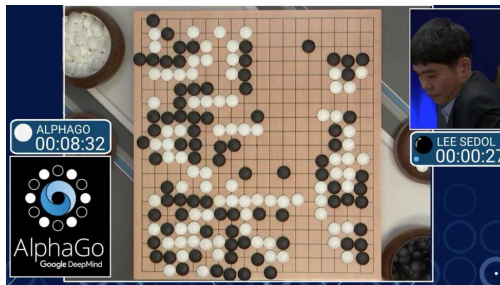
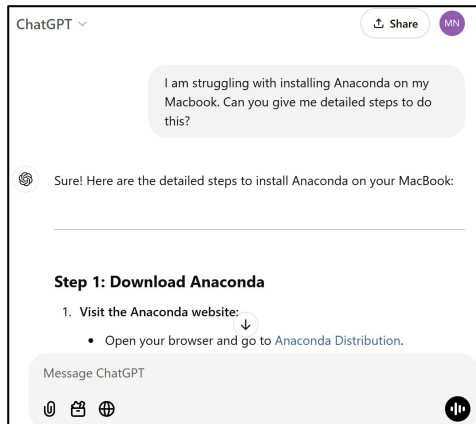
Online Tutorials/Office Hours Survey

- URL – <https://forms.office.com/r/SrPxYxARNu>



Why code?¹

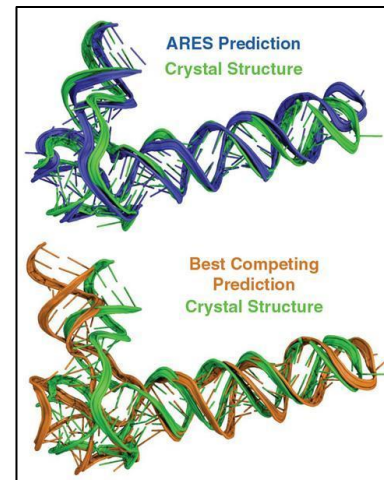
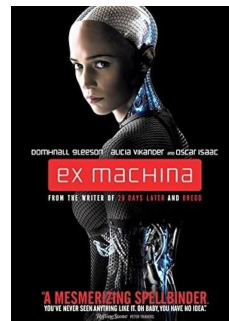
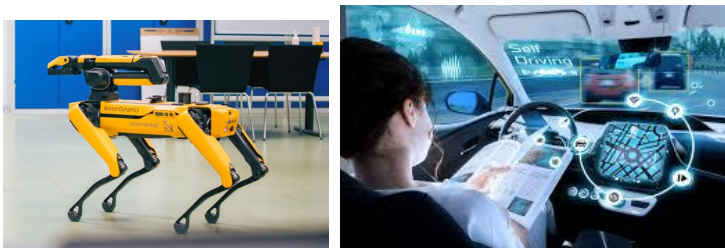
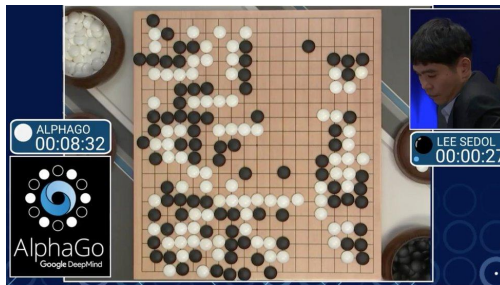
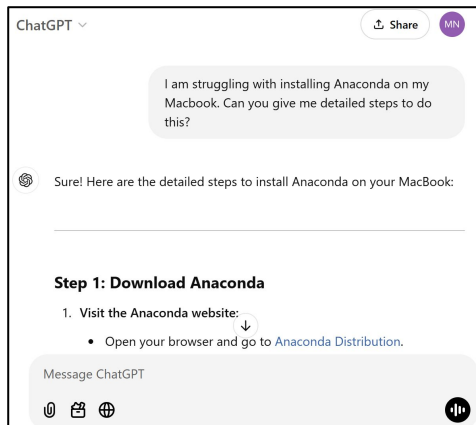
- Programming is awesome!
- Computers are everywhere, programming is a boundless opportunity
- Software engineering can be applied to almost any context in the world



Source: © Townshend et al, 2021 Science

Why code?²

- Regardless of background, coding is always an achievable skill
- Software engineering can be applied to almost any context in the world
- **NEVER THINK THAT YOU “CAN'T UNDERSTAND” CODING**



Source: © Townshend et al, 2021 Science



What would you want to ideally learn how to do with computers?

slido



What would you want to ideally learn how to do with computers?

① Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.



Coding experience?

slido



Coding experience?

① Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

Install Anaconda Navigator

Anaconda

- Anaconda is a distribution of **Python** that includes tools and packages geared towards scientific computing (such as data science and machine learning)
- **Anaconda Navigator** is the graphical user interface (GUI) allowing users to install and manage their programming environment without command line (terminal) prompts



Install Anaconda

- You can install Anaconda from the following link:
 - <https://www.anaconda.com/download>

[Products](#)[Solutions](#)[Resources](#)[Partners](#)[Company](#)[Free Download](#)[Sign Up](#)[Sign In](#)

Distribution

Free Download*

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

- ✓ Easily search and install thousands of data science, machine learning, and AI packages
- ✓ Manage packages and environments from a desktop application or work from the command line
- ✓ Deploy across hardware and software platforms
- ✓ Distribution installation on Windows, MacOS, or Linux

Provide email to download Distribution

Email Address:

☐

Agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent [here](#) at any time.

By continuing, I agree to Anaconda's [Privacy Policy](#) and [Terms of Service](#).

Submit >

Skip registration



Install Anaconda

- You can install Anaconda from the following link:
 - <https://www.anaconda.com/download>



[Products](#) [Solutions](#) [Resources](#) [Partners](#) [Company](#)

[Sign Up](#)

[Sign In](#)

Download Now

For installation assistance, refer to [Troubleshooting](#).

Download Anaconda Distribution or [Miniconda](#) by choosing the proper installer for your machine. Learn the difference from our [Documentation](#).



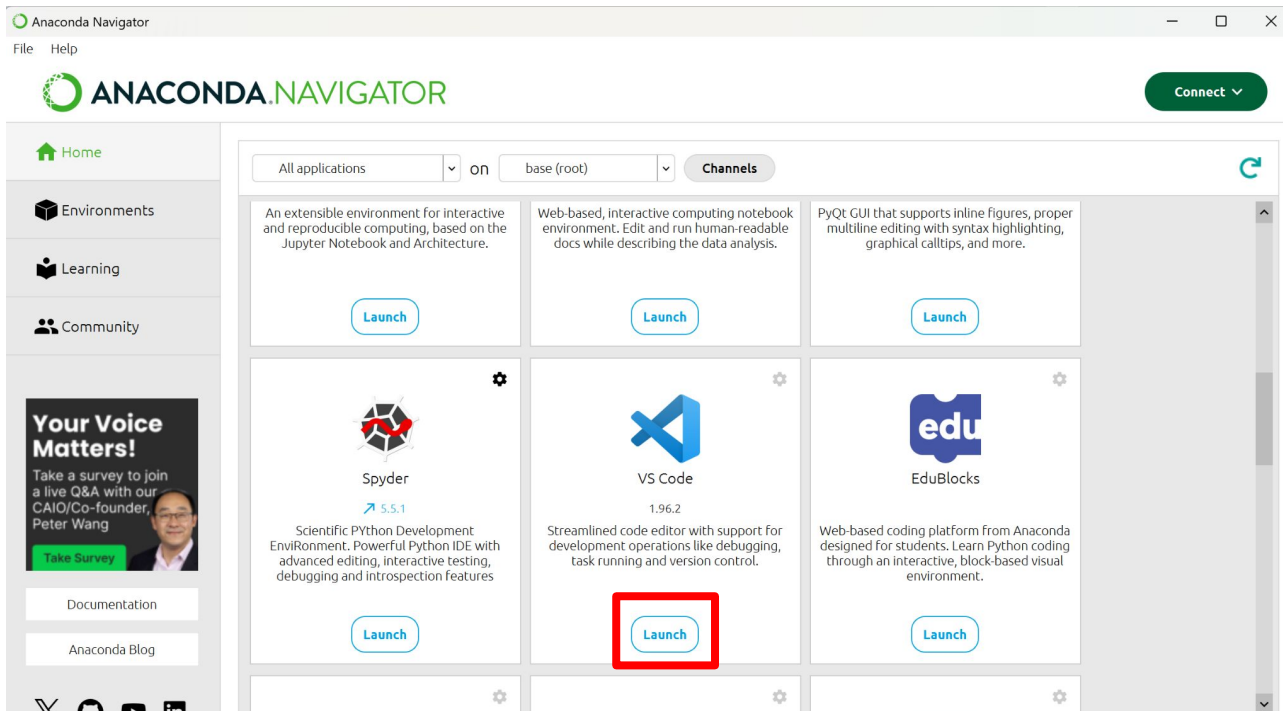
Anaconda Installers

 [Download](#)



Anaconda Navigator

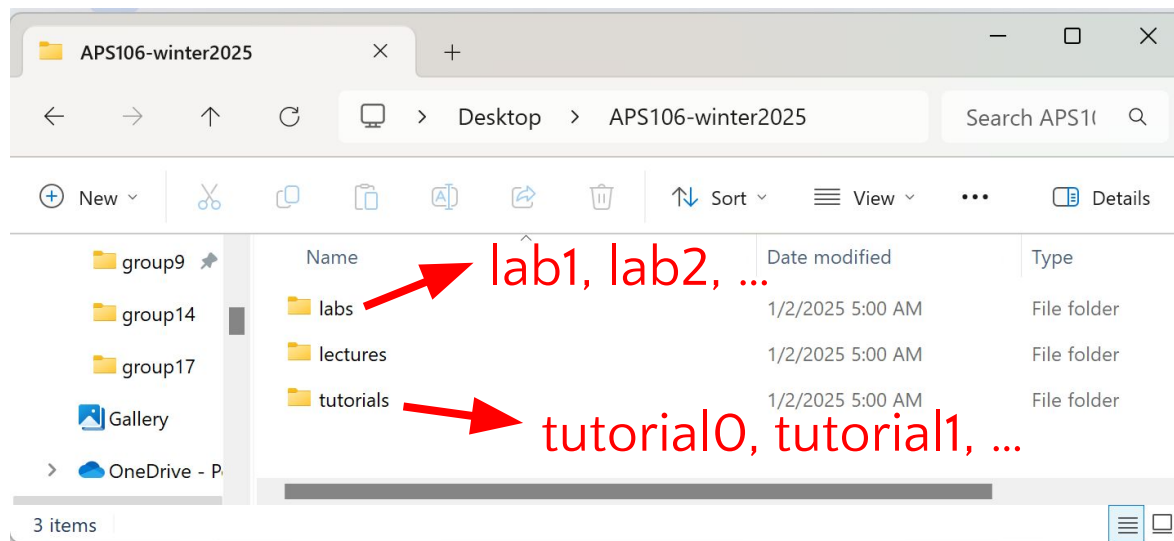
- We will be mainly using “**VS Code**” in this course. Let’s click on “Launch”!



Set up a Folder Structure for APS106

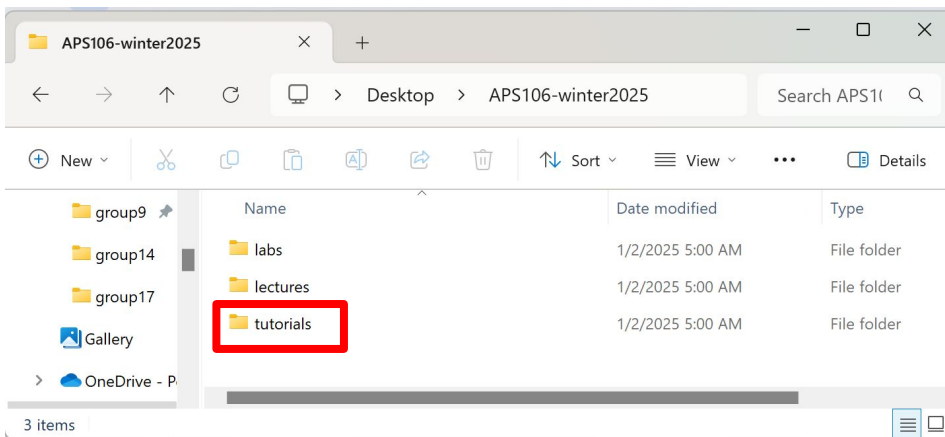
Stay Organized, Stay Efficient

- It's not a good idea to store all your files in the Downloads folder!
- Let's create a folder for APS106 (e.g., on Desktop) and organize your files there



Let's download files from Quercus!

- Go to Quercus → Modules → **Tutorial Homepage**
 - Click on **Tutorial 0 – Setting Up A Development Environment** in week1
 - Download “**tutorial0.zip**”
 - Unzip the zip file and move it in your folder for APS106



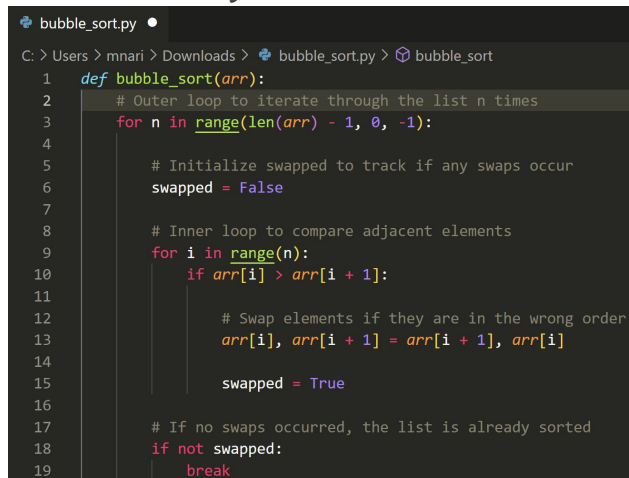
VSCode/Jupyter Notebook

VSCode is an IDE

- Integrated Development Environments (IDEs) are programs that provides tools and features to programmers in a unified environment
- IDEs often include:
 - **A code editor**
 - A place to type and edit code, usually with colour-coded syntax highlighting to improve readability
 - **Code compilers or interpreters**
 - Turns the readable Python code into something the machine can understand
 - **Debuggers**
 - Pause the code at pre-determined locations and go line-by-line through your code
- So IDEs basically contain **everything** you need to code!

VSCode is an IDE

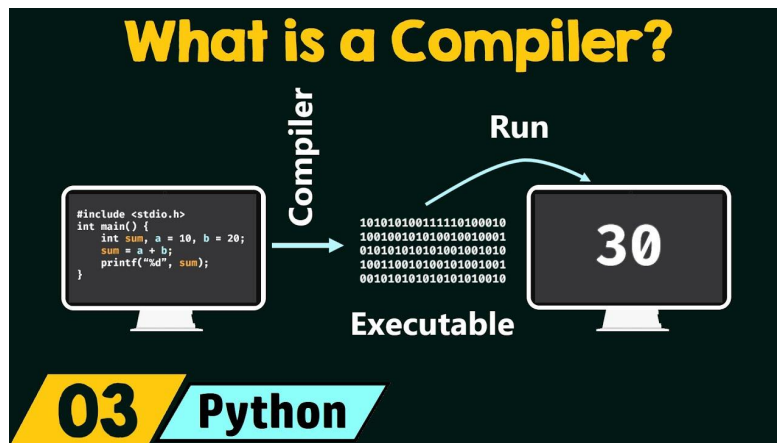
- Integrated Development Environments (IDEs) are programs that provides tools and features to programmers in a unified environment
- IDEs often include:
 - **A code editor**
 - A place to type and edit code, usually with colour-coded syntax highlighting to improve readability



```
bubble_sort.py •
C: > Users > mnari > Downloads > bubble_sort.py > bubble_sort
1  def bubble_sort(arr):
2      # Outer loop to iterate through the list n times
3      for n in range(len(arr) - 1, 0, -1):
4
5          # Initialize swapped to track if any swaps occur
6          swapped = False
7
8          # Inner loop to compare adjacent elements
9          for i in range(n):
10             if arr[i] > arr[i + 1]:
11
12                 # Swap elements if they are in the wrong order
13                 arr[i], arr[i + 1] = arr[i + 1], arr[i]
14
15                 swapped = True
16
17             # If no swaps occurred, the list is already sorted
18             if not swapped:
19                 break
```

VSCode is an IDE

- Integrated Development Environments (IDEs) are programs that provides tools and features to programmers in a unified environment
- IDEs often include:
 - **Code compilers or interpreters**
 - Turns the readable Python code into something the machine can understand

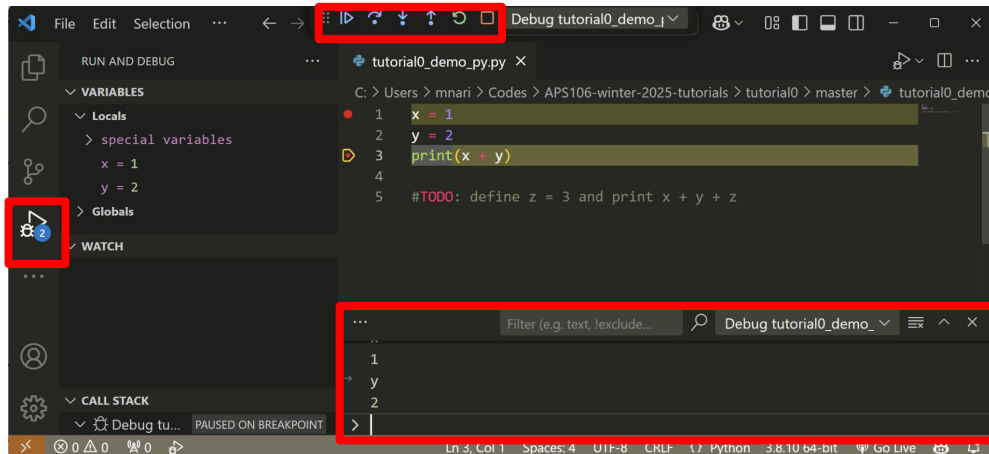


VSCoDe is an IDE

- Integrated Development Environments (IDEs) are programs that provides tools and features to programmers in a unified environment
- IDEs often include:
 - **Debuggers**
 - Pause the code at pre-determined locations and go line-by-line through your code

Continue, step over, step in/out, restart, stop

Run and Debug



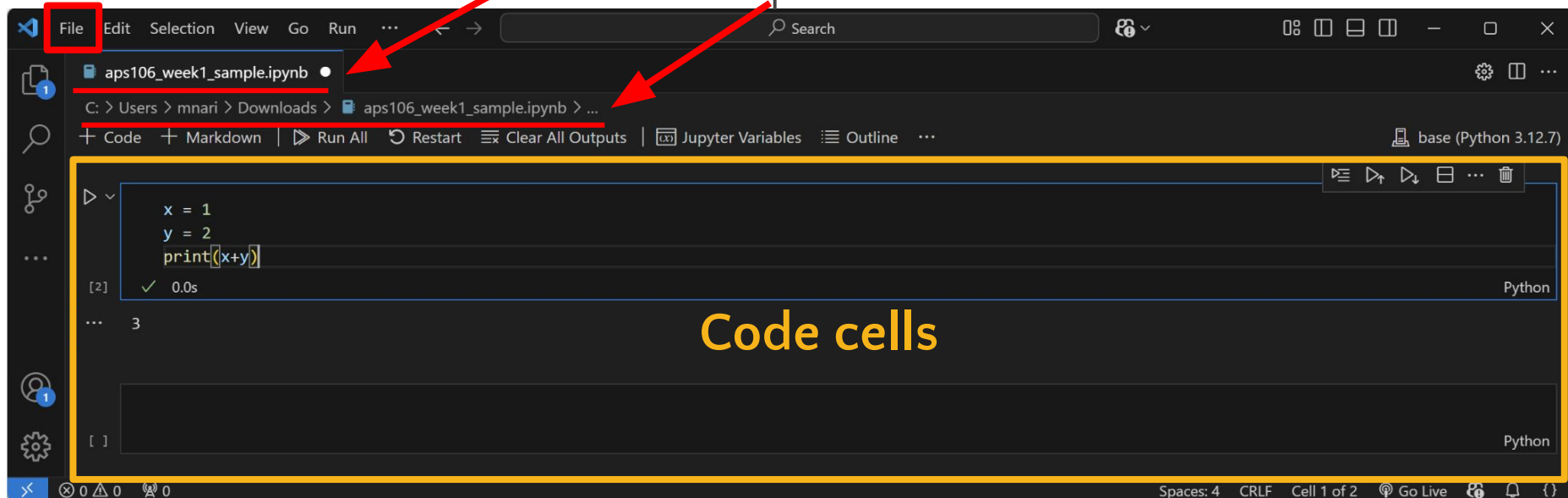
Debug console

VS Code (Jupyter Notebook)

Open/Save files

Filename

File path

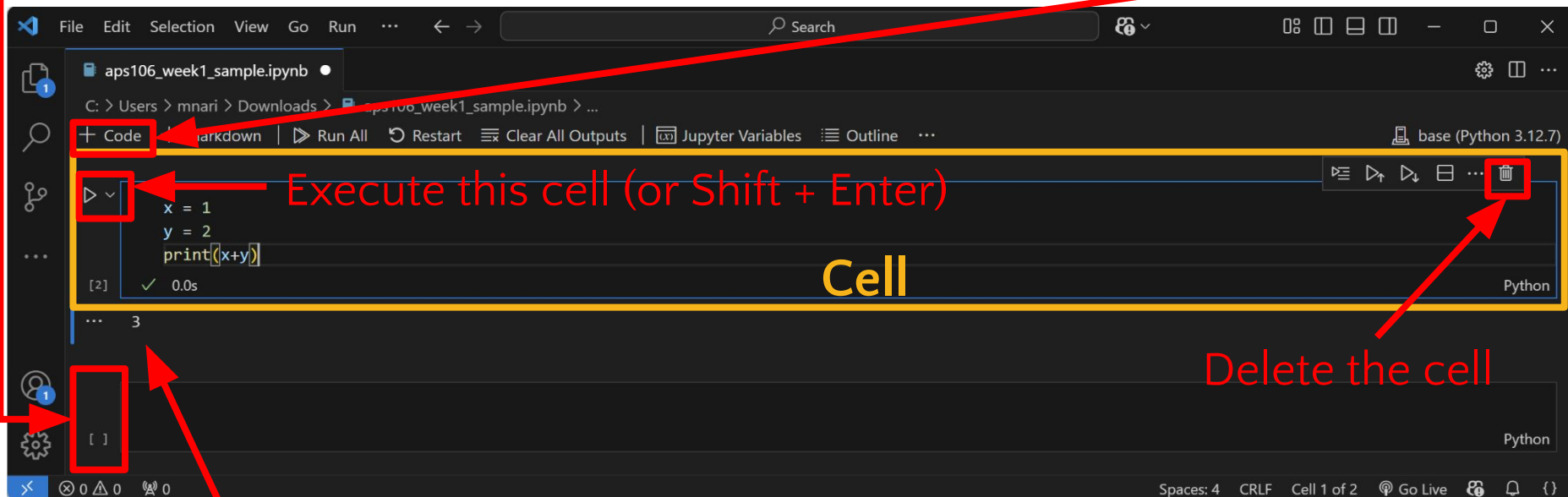


VS Code (Jupyter Notebook)

(drag&drop or Alt + ↑ / Alt + ↓)

Move up/down the cell

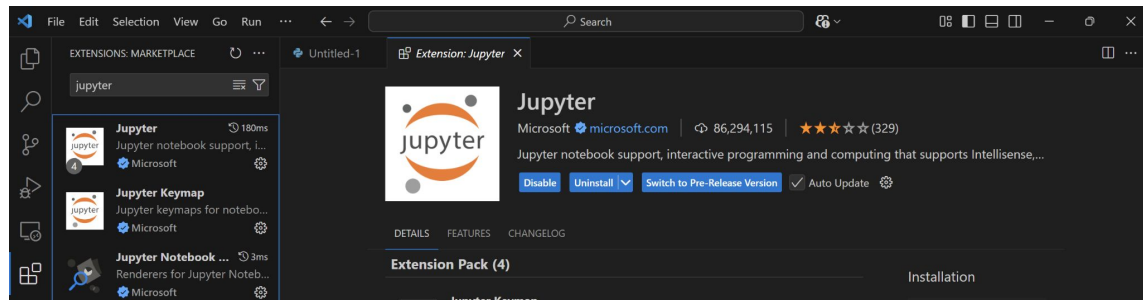
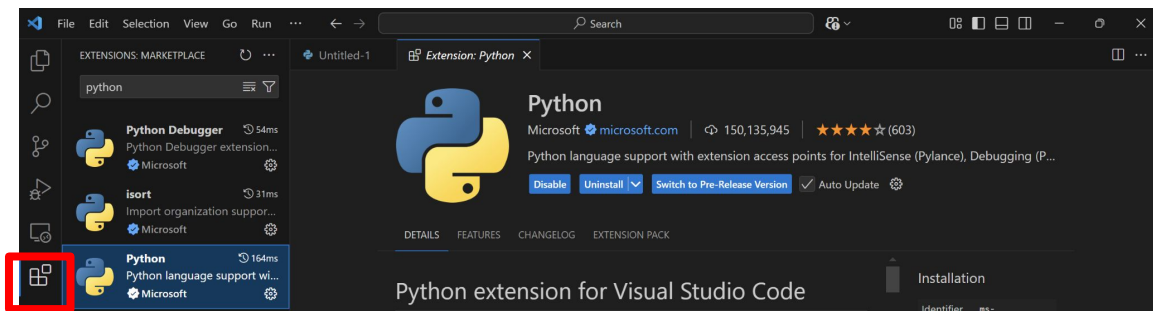
Create a new cell under the current one



Cell output

VS Code (Jupyter Notebook)

- Make sure you have “Python” and “Jupyter” extensions installed!

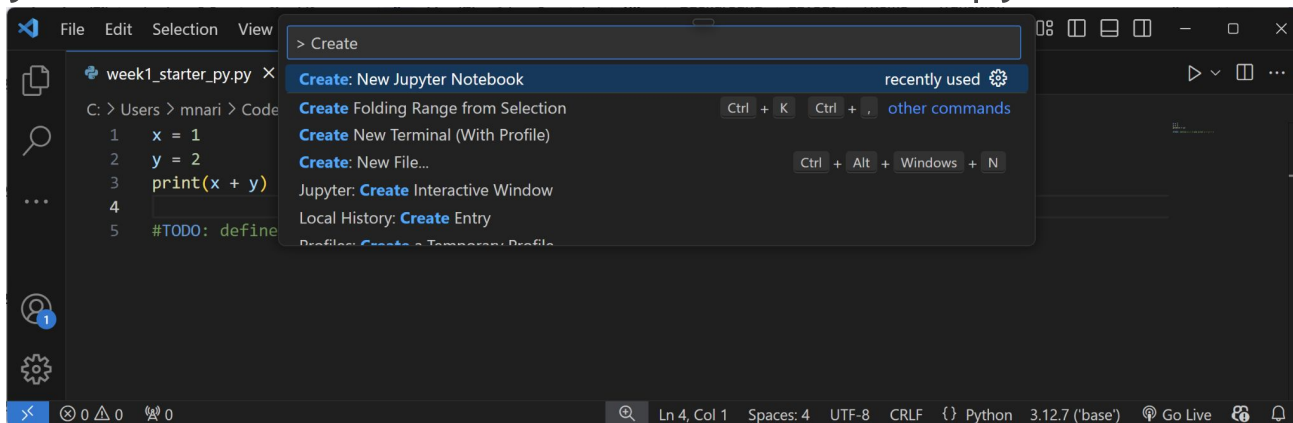


Let's practice!¹

- Let's open **tutorial0_demo_jupyter.ipynb**, write our first program, and save it!
- Steps:
 1. Launch VS Code through Anaconda Navigator
 2. File -> Open File -> Go to the APS106 folder and select **tutorial0_demo_jupyter.ipynb**
 3. In the first code cell, write
print("Hello World!")
 4. Execute the cell (if asked, select "base (Python 3.12.7)")
 5. File -> Save as -> save it as **"tutorial0_demo_jupyter.ipynb"**

Create a new Jupyter Notebook

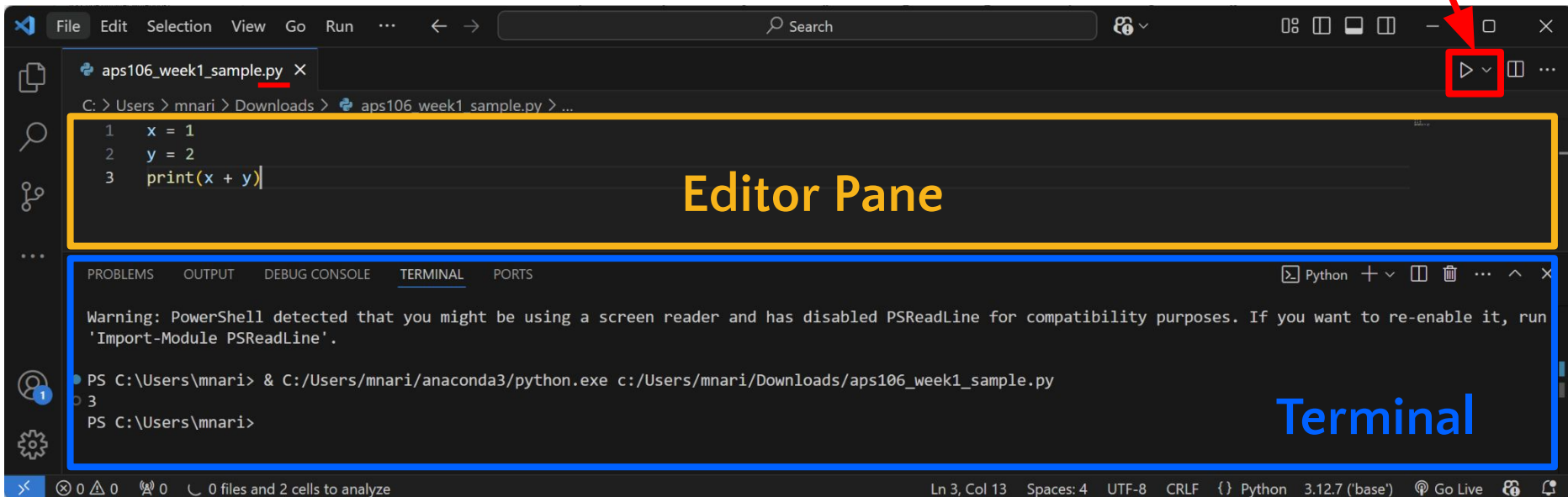
- You can create a new notebook on VS Code too!
- Steps:
 1. Launch VS Code through Anaconda Navigator
 2. In the command palette (“Search” on top), type in “> **create new jupyter notebook**” and select “Create: New Jupyter Notebook”



VS Code (Python script)

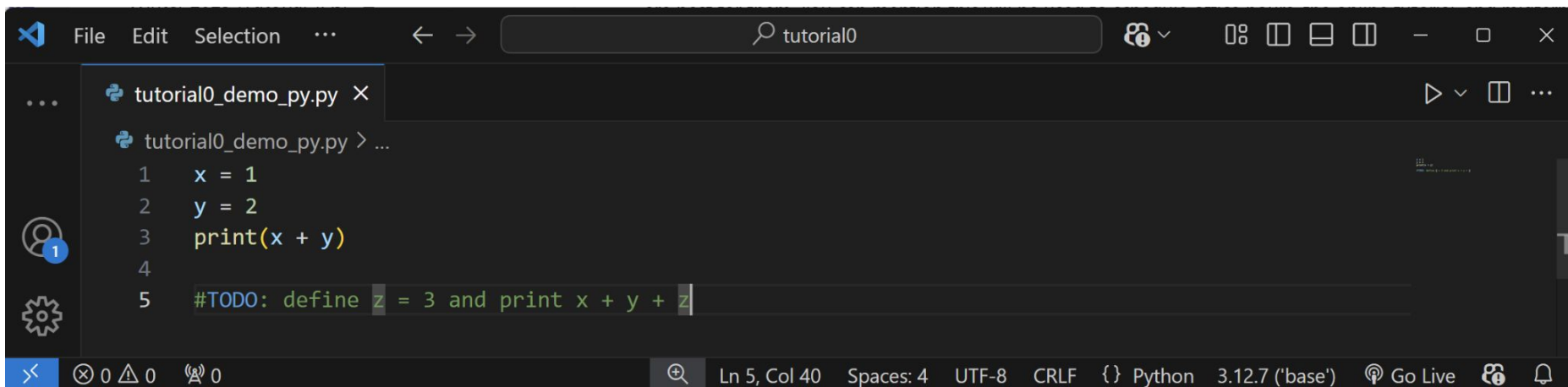
- When you open a `.py` file, an editor with a different layout appears

Execute the Python script



Let's practice!²

- Let's open **tutorial0_demo_py.py** in VS Code, modify the code based on the instruction, and run it!



```
File Edit Selection ... tutorial0
```

```
... tutorial0_demo_py.py X
```

```
... tutorial0_demo_py.py > ...
```

```
1 x = 1
```

```
2 y = 2
```

```
3 print(x + y)
```

```
4
```

```
5 #TODO: define z = 3 and print x + y + z
```

```
< 0 0 0 Ln 5, Col 40 Spaces: 4 UTF-8 CRLF {} Python 3.12.7 ('base') Go Live
```

We'll switch from .ipynb to .py files later

- Jupyter Notebooks (.ipynb) are interactive and beginner-friendly
- Python scripts (.py) are **the standard format** for Python programs and are suited for building and running **larger projects**
- We'll switch to .py files in **lab5!**

APS106



UNIVERSITY OF
TORONTO

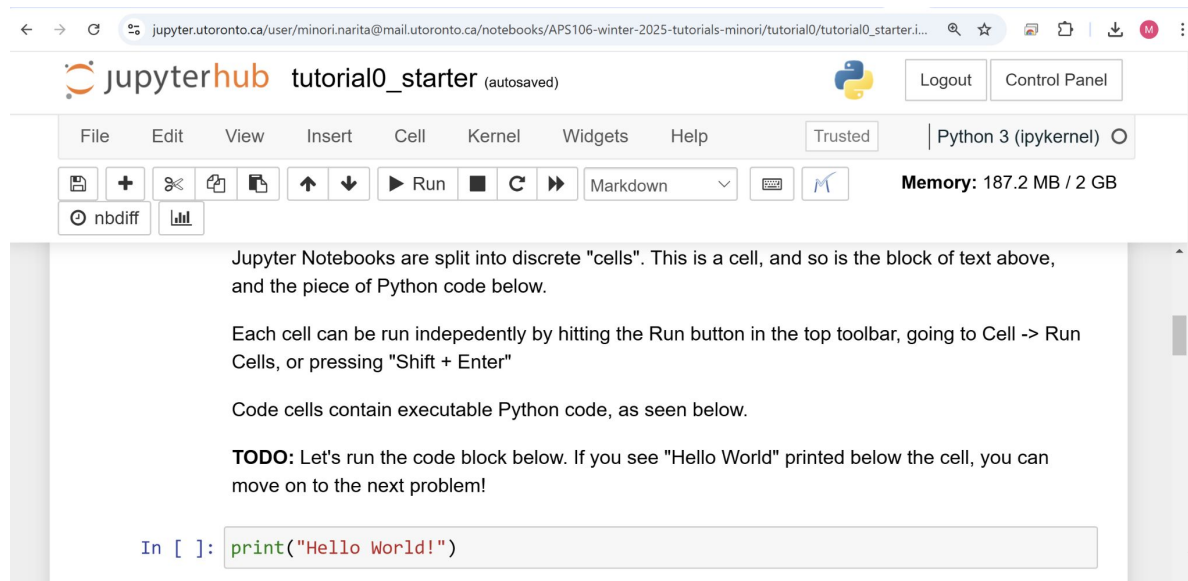
UofT JupyterHub

What is UofT JupyterHub?

- Access at <https://jupyter.utoronto.ca/>
- Cloud-based Jupyter Notebook service that allows us to run Jupyter Notebooks (.ipynb files) directly from a web browser
 - Don't need to install anything
- Linked to your UofT account, all lecture notes are stored as a copy in JupyterHub!

Let's try UofT JupyterHub!

- Go to **Tutorial Homepage** in APS106 Modules on Quercus
 - Click on **Tutorial 0 – Setting Up A Development Environment** in week1
 - Click on **JupyterHub Starter Link**



The screenshot shows a web browser window displaying a JupyterHub interface. The address bar shows the URL: `jupyter.utoronto.ca/user/minori.narita@mail.utoronto.ca/notebooks/APS106-winter-2025-tutorials-minori/tutorial0/tutorial0_starter.i...`. The page title is "jupyterhub tutorial0_starter (autosaved)". The interface includes a top navigation bar with "Logout" and "Control Panel" buttons. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". A "Trusted" status indicator and "Python 3 (ipykernel)" are also visible. The main toolbar contains icons for saving, creating a new notebook, opening a recent notebook, uploading a file, downloading a file, running a cell, and other actions. The memory usage is shown as "Memory: 187.2 MB / 2 GB". The main content area displays text explaining Jupyter Notebooks and a code cell with the following text:

Jupyter Notebooks are split into discrete "cells". This is a cell, and so is the block of text above, and the piece of Python code below.

Each cell can be run independently by hitting the Run button in the top toolbar, going to Cell -> Run Cells, or pressing "Shift + Enter"

Code cells contain executable Python code, as seen below.

TODO: Let's run the code block below. If you see "Hello World" printed below the cell, you can move on to the next problem!

```
In [ ]: print("Hello World!")
```