Files needed today:
`python_practice.py` [canvas]

Introduction to Coding in Python

Dr. Groves
Dr. Peirce-Cottler

Module 0 Lecture 1

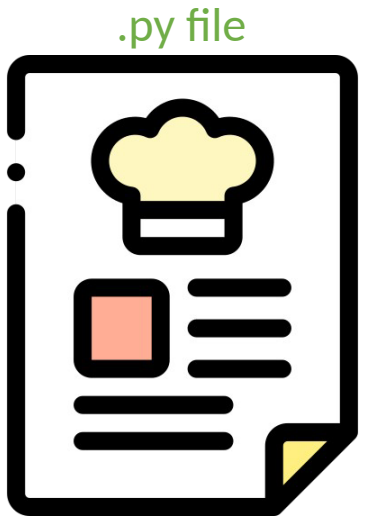
Learning objective: I can do basic tasks in Python (loading a file, writing a for loop, calling a function) using VS code or another

Setting up your computer for this class

- You should have:
  - Anaconda installed
  - VS Code installed (or an IDE you like)
  - A GitHub account
- In module 0, we will add:
  - A folder location to organize your files (code files and data files for this class)
  - VS Code extensions:
    - Jupyter notebook extensions
    - Optional extensions: CSV viewer
  - Python packages needed for this class
  - Github repositories for each project

How do coding tools fit together?

- Writing code ≠ running code
- Think of coding like running a **restaurant**
  - .py files
  - Jupyter notebooks
  - VS Code (IDE)
  - Python interpreter
  - Terminal/Command Prompt
  - Anaconda & environments
  - GitHub and git
- Your code = the recipe
  - It describes ingredients and steps, but a recipe does not cook itself!





Folder

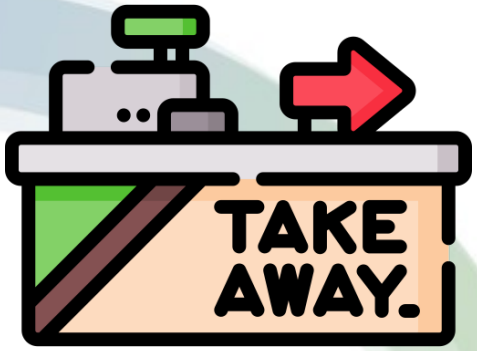
Organization: The recipe book

- Pick one location to make a folder for this class (Documents? UVA Box? Home?)
  - Each Module will have its own subfolder that we will put in this folder
  - Any extra data/files from Canvas should go here
- Think about what you want to name your files!



Python interpreter: The Chef

- The chef reads the recipe line by line and executes each instruction
- Different chefs exist:
  - Python 3.7
  - Python 3.9
  - Python 3.11 …
- Same recipe with different chef → slightly different outcomes!



Terminal & Command Prompt: The Order Counter

- The terminal is where orders are placed

```
python analysis.py
```

Means "Chef Python, make the analysis.py recipe"

- We can access the terminal through stand-alone apps (terminal, command prompt) or *within VS code*
  - Like different windows at the same order counter

The Restaurant: The Big Picture

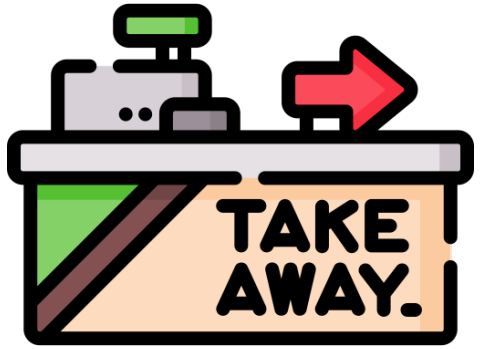
You write recipes (code)…



… and put them in a recipe book (folder)…



… which you can send to the order counter (terminal)…



I need an analysis.py, stat!

…which tells the chef (Python) what to cook

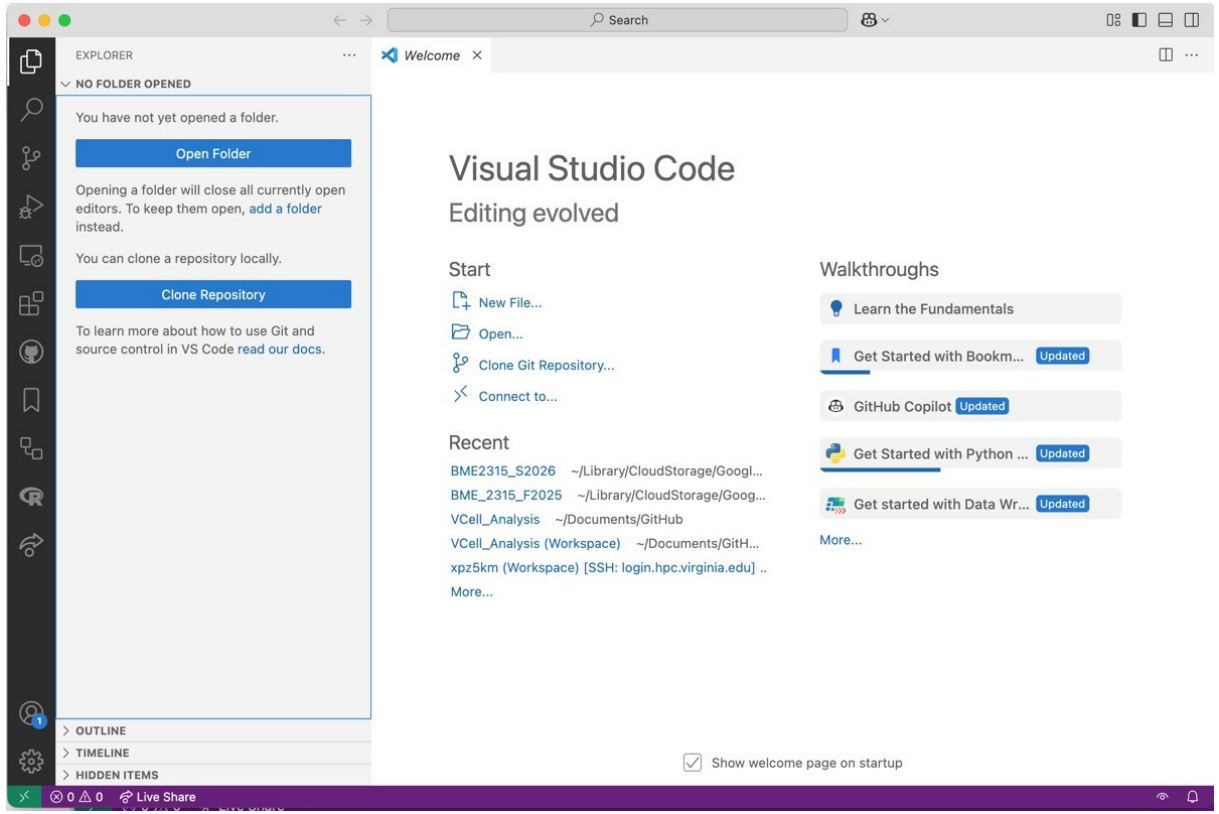


The Restaurant (VS Code)

All of this happens inside the restaurant, VS Code!

VS Code & Integrated Development Environments: The Restaurant

- VS code is a place to work on code (write recipes), view data files (peek in the kitchen), and run your code through a terminal (tell the chef what to cook)
- Integrated: many tools in one place
- Development: place to build code
- Environment: workspace



Let’s practice!

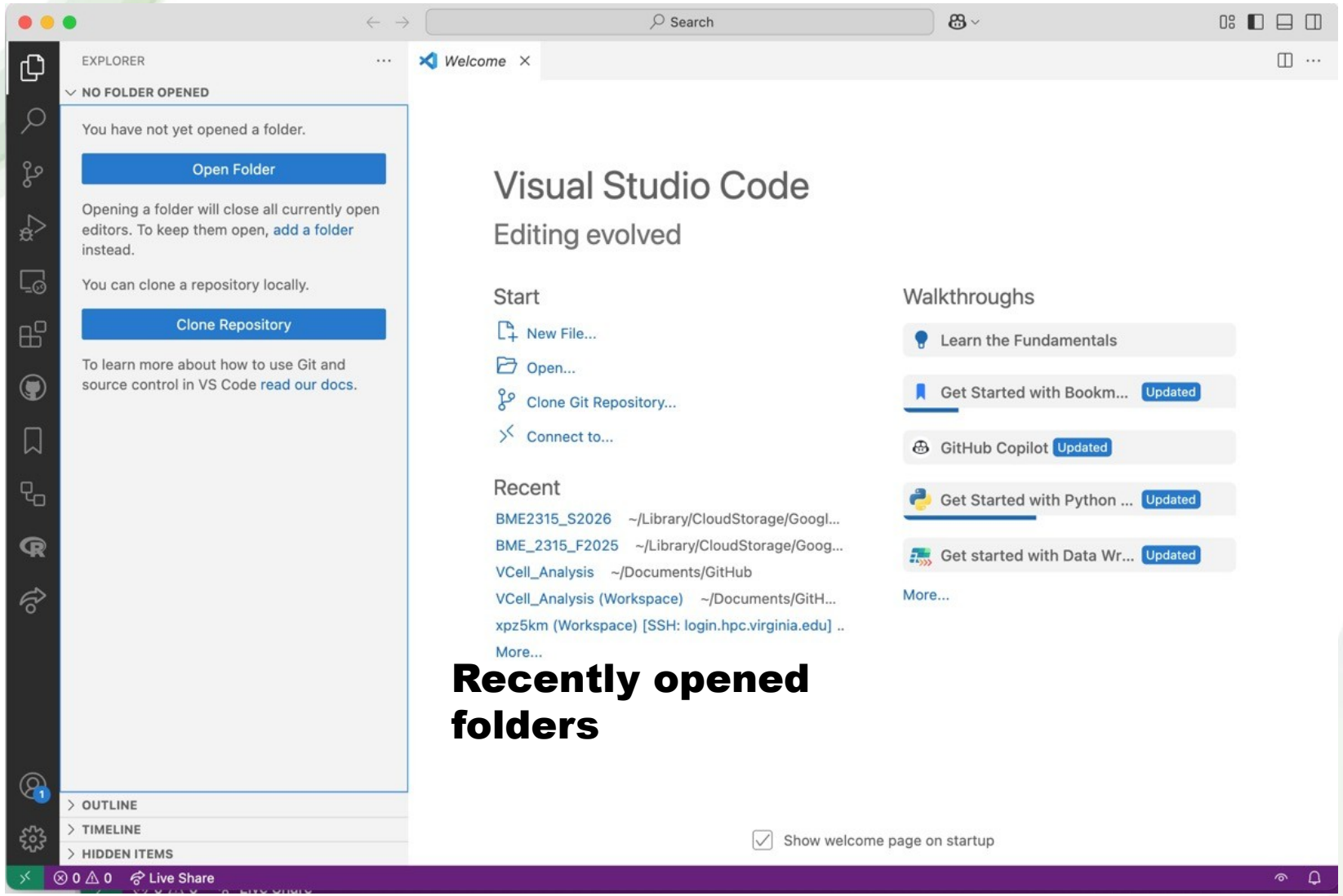
Make a **folder** for this class on your computer.

Open **VS Code** (or another editor).

Stand your name card up on its edge when you are done. Raise your hand for help!

VS Code IDE

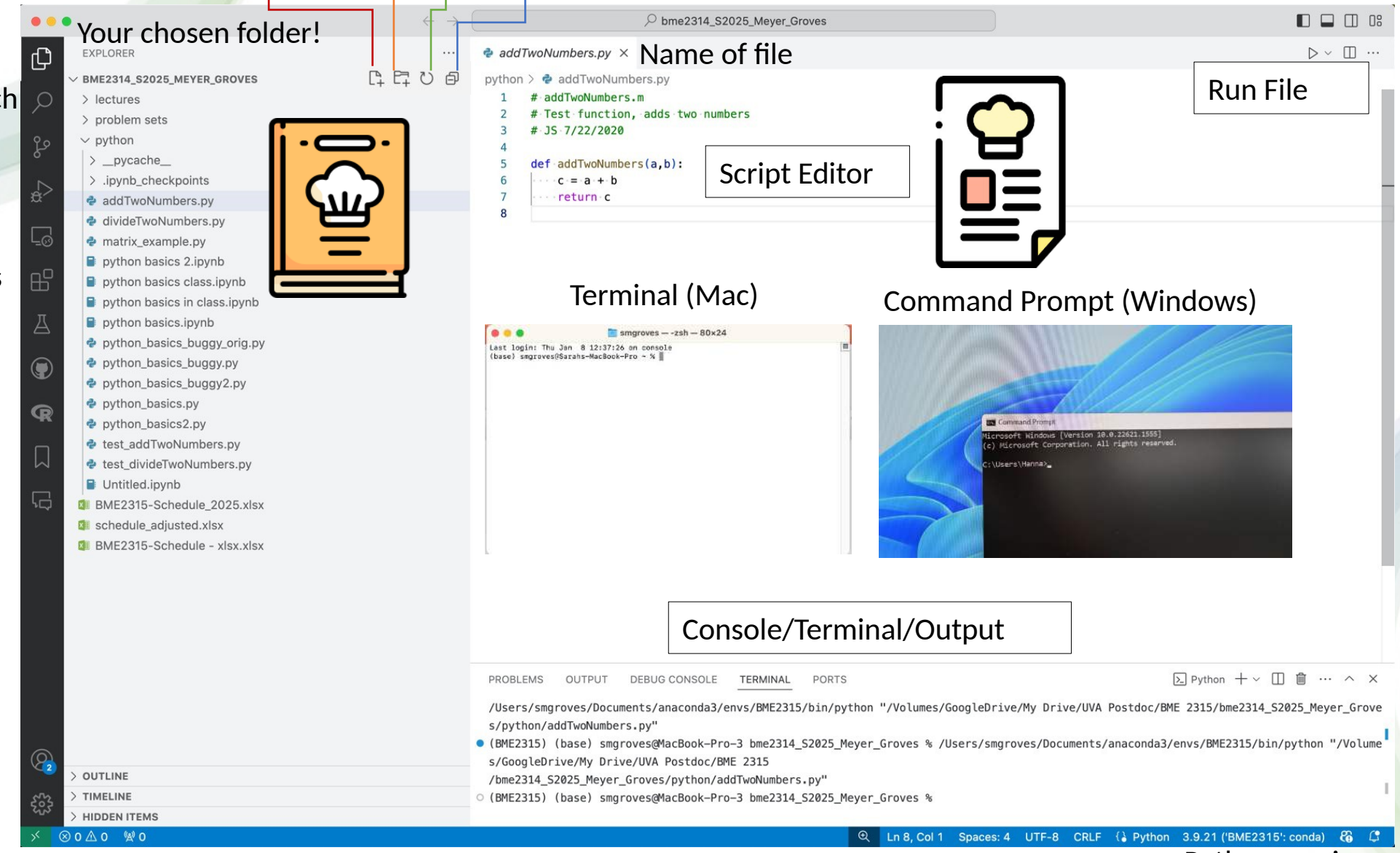
Welcome



Recently opened folders

Always start by opening your class folder from VS code!

VS Code IDE


Add a file
Add a folder
Refresh
Collapse all folders
Search or run a command
File Explorer
Search
Extensions
Your chosen folder!
Name of file
Run File
Script Editor
Terminal (Mac)
Command Prompt (Windows)
Console/Terminal/Output
Python version
EXPLORER
BME2314_S2025_MEYER_GROVES
lectures
problem sets
python
__pycache__
.ipynb_checkpoints
addTwoNumbers.py
divideTwoNumbers.py
matrix_example.py
python basics 2.ipynb
python basics class.ipynb
python basics in class.ipynb
python basics.ipynb
python_basics_buggy_orig.py
python_basics_buggy.py
python_basics_buggy2.py
python_basics.py
python_basics2.py
test_addTwoNumbers.py
test_divideTwoNumbers.py
Untitled.ipynb
BME2315-Schedule_2025.xlsx
schedule_adjusted.xlsx
BME2315-Schedule - xlsx.xlsx
OUTLINE
TIMELINE
HIDDEN ITEMS
bme2314_S2025_Meyer_Groves
addTwoNumbers.py
python > addTwoNumbers.py
# addTwoNumbers.m
# Test function, adds two numbers
# JS 7/22/2020
def addTwoNumbers(a,b):
c = a + b
return c
smgroves — -zsh — 80×24
Last login: Thu Jan 8 12:37:26 on console
(base) smgroves@Sarahs-MacBook-Pro ~ %
Command Prompt
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Hanna>
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
/Users/smgroves/Documents/anaconda3/envs/BME2315/bin/python "/Volumes/GoogleDrive/My Drive/UVA Postdoc/BME 2315/bme2314_S2025_Meyer_Groves/python/addTwoNumbers.py"
(BME2315) (base) smgroves@MacBook-Pro-3 bme2314_S2025_Meyer_Groves % /Users/smgroves/Documents/anaconda3/envs/BME2315/bin/python "/Volumes/GoogleDrive/My Drive/UVA Postdoc/BME 2315
/bme2314_S2025_Meyer_Groves/python/addTwoNumbers.py"
(BME2315) (base) smgroves@MacBook-Pro-3 bme2314_S2025_Meyer_Groves %
Ln 8, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.9.21 ('BME2315': conda)

Let’s practice!

Open the folder for this class in VS code.

Add python_practice.py from canvas into this folder and open it in VS code.

Go to the extensions marketplace and download Python & Jupyter.

Stand your name card up on its edge when you are done. Raise your hand for help!

Tip: File > New Window will open a new window with the welcome page again



Anaconda:
The supplier & pantry
Conda environments:
The kitchens

- Anaconda is the supplier that includes Python and scientific libraries
  - The package manager gives you a stocked pantry
- Conda environments = separate kitchens
  - The *base* environment is the default
  - You can make as many environments as you want. This is useful for organizing projects.
  - Each kitchen can have different chefs (Python versions) and different pantries
  - Ensures there are no “ingredient conflicts”
- VS code works in ***one kitchen at a time***

“This *special* package requires python 3.7”

`conda create …`



`conda activate …`

The Restaurant: The Big Picture



Anaconda supplies the ingredients…

…to different kitchens (environments)…

I need an analysis.py from the **blue** kitchen!

Orange kitchen, get me a statistics.py!

You write recipes (code)…



… and put them in a recipe book (folder)…



… which you can send to the order counter (terminal)…





…which tells the chef (Python) what to cook







…with different chefs (Python versions)…





The Restaurant (VS Code/IDE)

VS Code IDE



Command palette: Command/Control + Shift + P

Configures the right version of Python for the entire IDE

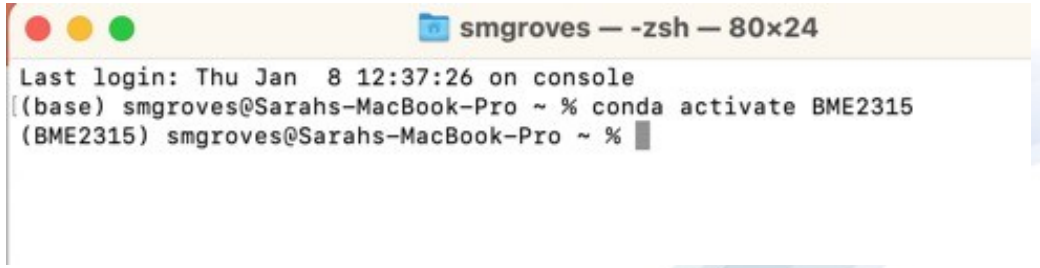
Do this once!

Terminal



The *base* env is already activated

I am activating a special env (BME2315)



“conda activate BME2315”
“source activate BME2315”

Configures the right version of Python for this **specific** terminal session

Do this when installing packages

Importing packages (adding food to the pantry)

- *conda* and *pip* are two major "warehouses"
  - Most python packages are hosted on Python Package Index (pypi) and can be installed with pip
  - Ensure the right conda environment is activated to add the package in the right location
  - If a package is not available through pip, try conda (or look at the install guide)

```
Last login: Thu Jan  8 13:59:41 on ttys002
(base) smgroves@Sarahs-MacBook-Pro ~ % conda activate BME2315
(BME2315) smgroves@Sarahs-MacBook-Pro ~ % pip install seaborn
Requirement already satisfied: seaborn in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-pack
Requirement already satisfied: pandas>=1.2 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (fro
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-p
Requirement already satisfied: contourpy>=1.0.1 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages
Requirement already satisfied: cycler>=0.10 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (fr
Requirement already satisfied: fonttools>=4.22.0 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-package
Requirement already satisfied: kiwisolver>=1.3.1 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-package
Requirement already satisfied: packaging>=20.0 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages
Requirement already satisfied: pillow>=8 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (from
Requirement already satisfied: pyparsing>=2.3.1 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages
Requirement already satisfied: python-dateutil>=2.7 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-pack
Requirement already satisfied: importlib-resources>=3.2.0 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/sit
Requirement already satisfied: pytz>=2020.1 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (fr
Requirement already satisfied: tzdata>=2022.7 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (
Requirement already satisfied: zipp>=3.1.0 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (fro
Requirement already satisfied: six>=1.5 in ./Documents/anaconda3/envs/BME2315/lib/python3.9/site-packages (from p
(BME2315) smgroves@Sarahs-MacBook-Pro ~ %
```

Let’s practice!

(1) Using the **command palette**, select the base environment in VS code

OR

(advanced, in terminal*) make a new environment for this class and select it in VS code.

(2) Using the **terminal** (in VS code or separate app), activate the conda environment and

pip install pandas numpy scipy ipykernel matplotlib

Stand your name card up on its edge when you are done.

Raise your hand for help!

*conda create --name BME2315_S26 python=3.12 numpy pandas scipy ipykernel matplotlib

Using genAI for coding



The Restaurant: The Big Picture





This dish tastes disgusting! Who made it?

I just followed the recipe that **ChatGPT** made me… I don't know where I went wrong.



















The Restaurant (VS Code/IDE)

What should we do instead?

Open in VS Code the python_practice.py* file you just added to your folder

**The completed practice is due next Thursday 1/23*

Open: python_practice.py in VS Code in your folder!

Announcements

- Python practice (submitted through GitHub) will be due **TUESDAY January 27 by 12:30pm**
- Rubric for Module 0 individual assignment:
  - GitHub Repo for Module 0
  - ReadMe updated
  - Python practice file on your own branch with all problems completed
  - Jupyter Notebook

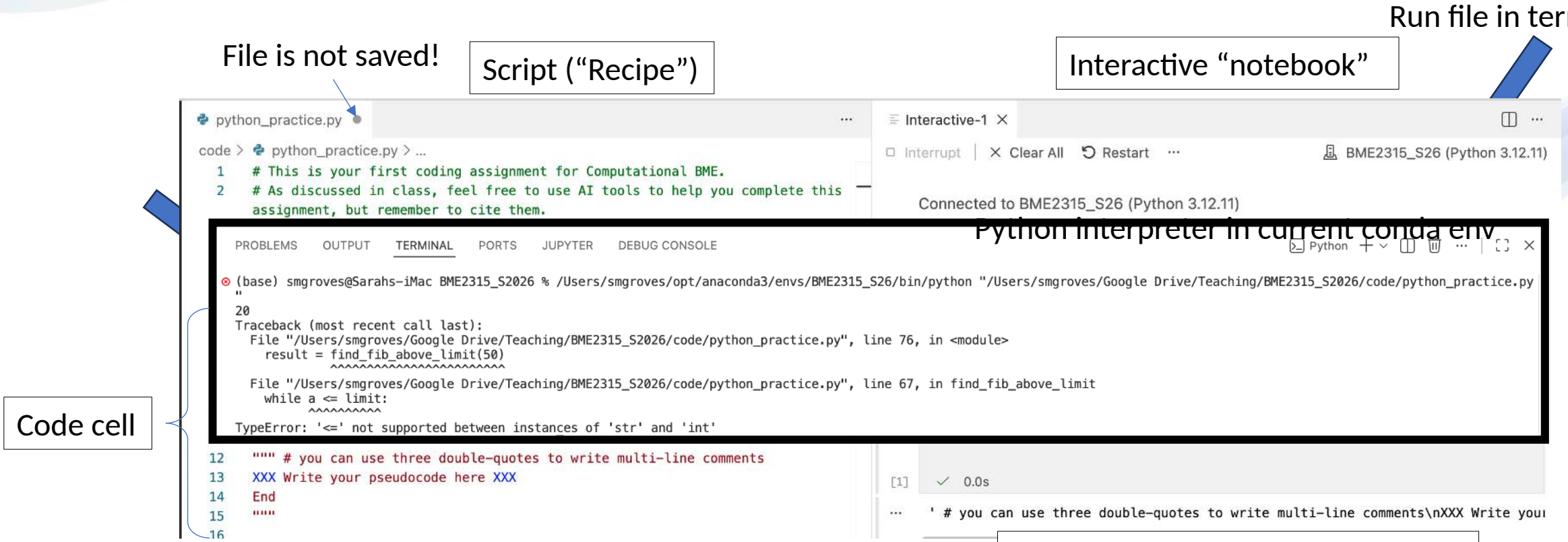
Tip # 1:
Run code in “chunks” when developing

Or, how to break code into smaller pieces that are easier to run and interpret with interactive mode or Jupyter notebooks

Try following along on your own computer!

Interactive mode in Python

- #%% will make code cells that can run in interactive mode within a **normal .py file**
- Requires Jupyter extension (running Jupyter in the background)



https://code.visualstudio.com/docs/python/jupyter-support-py

Tip # 2: Write pseudocode first

Or, how to design a project and decide what even needs to be coded before starting

In plain English, write out (line by line, as the computer should see them):

- What are the inputs?
- What are the outputs?
- What is computed or calculated?
- What variables are set or initialized?
- Any logic statements like

```
IF-THEN-ELSE
IF condition THEN
    sequence 1
ELSE
    sequence 2
ENDIF
```

- Any loop structures like

```
REPEAT-UNTIL
REPEAT
sequence
UNTIL condition
```

```
WHILE
WHILE condition
sequence
ENDWHILE
```

```
FOR
FOR iteration bounds
sequence
ENDFOR
```

Problem 1

Tip #3:
Comment your code

Or, how to remind a future you what you did and why

- You should try to be as **descriptive** as possible, based on the context of the problem you are solving
- In the practice problem, it will not be helpful to say:

```
a = 0 # set a to 0
b = 1 # set b to 1
```

- Instead, say:

```
a = 0 # set a to the first fibonacci number
b = 1 # set b to the second fibonacci number
```

Tip #4: Don't reinvent the wheel

Or, how to use common Python libraries to make things easier and more efficient

- **Modules** (ingredients) come from warehouses (Python **libraries**)
- **Anaconda**, the supplier, pulls from these warehouses and the ingredients end up in your pantry (**conda environment**)
- Any package you have installed in your environment can be used in your code with **import** statements

```
import numpy.linalg as la
```

- Means "Go to the NumPy warehouse, bring the linalg ingredients into my pantry, and label them la."

```
from matplotlib import pyplot as plt
```

- Means "From the matplotlib warehouse, grab pyplot and label it plt"

La.norm()

From numpy.linalg import *

Norm()

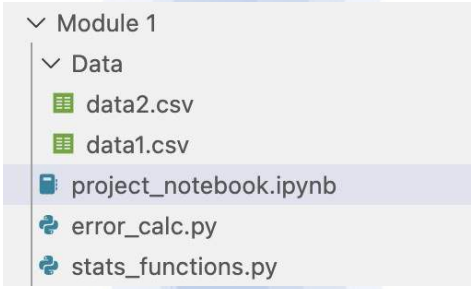
Problem 3

Tip #5: Don't repeat yourself

Or, how to recognize when you should write functions and loops

- Whenever you can think of a chunk of code as a step in a process, it is useful to write it as function
- Apply the same code to different inputs
- Import your own functions from modules (.py files) into your final Jupyter Notebook report

Example folder structure:



Example implementation in project_notebook.ipynb:

```
from error_calc import function1, function2
import stats_functions as sf
import pandas as pd

data = pd.read_csv("data/data1.csv")

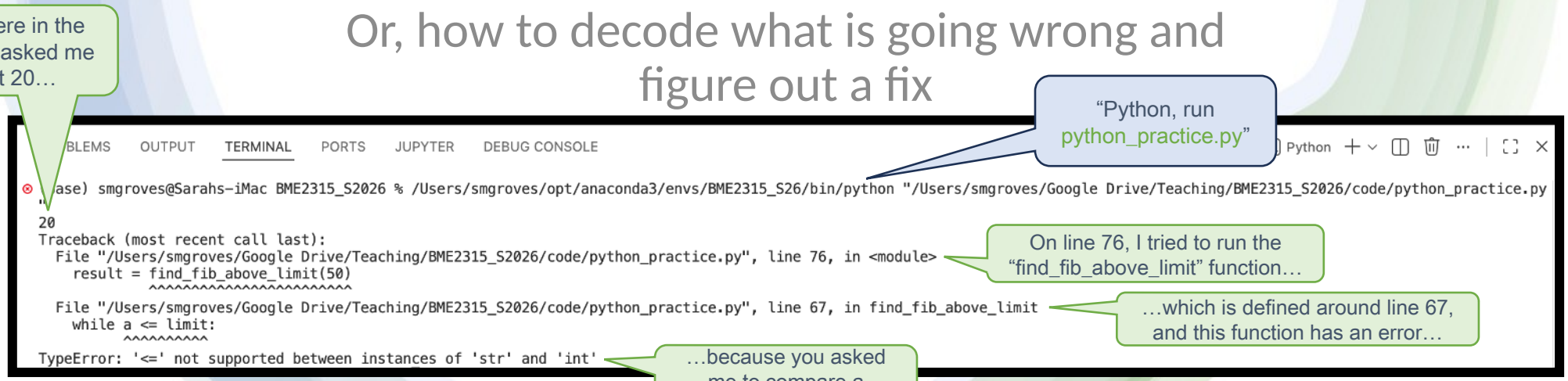
outputA = function1(data)
outputB = function1(data)

print(sf.stats_function1(outputA, outputB))
```

Problem 4

Tip #6: Read your error messages

Or, how to decode what is going wrong and figure out a fix

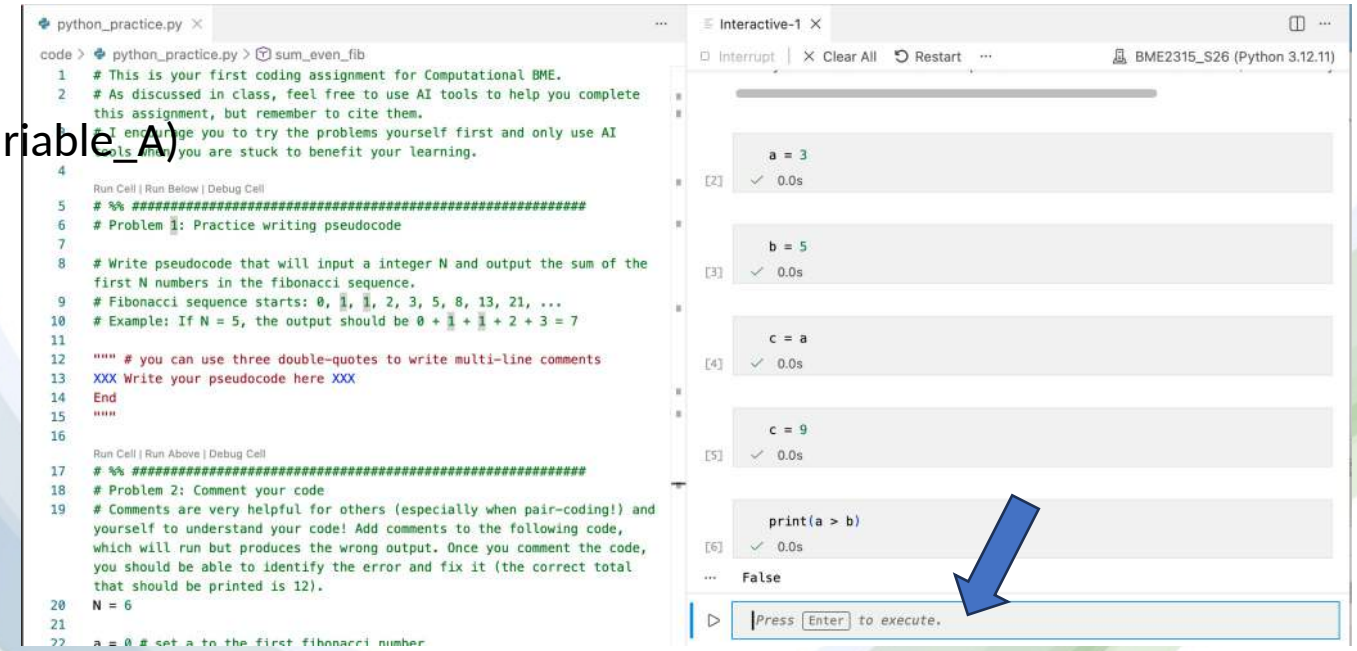


Problem 5

- Think of different **use cases** for a function you write
- Consider the **extremes** of the use cases
- Write **tests** that run the function for those extremes, and then **check** if the answer is correct
- If things are hard to decipher, add lots of **print** statements!
- In interactive mode, you can type directly into the notebook
  - The notebook will remember any variable in this session, but they are cleared once you close out of the notebook

Tip #7: Test your code

Print("1st problem", variable_A)



Problem 6

Homework

- By next class: download GitHub Desktop app if you aren't familiar with using GitHub (otherwise command line is fine!)
- Complete **video** assignment and comprehension **quiz** (remember you can retake it as many times as you'd like!)