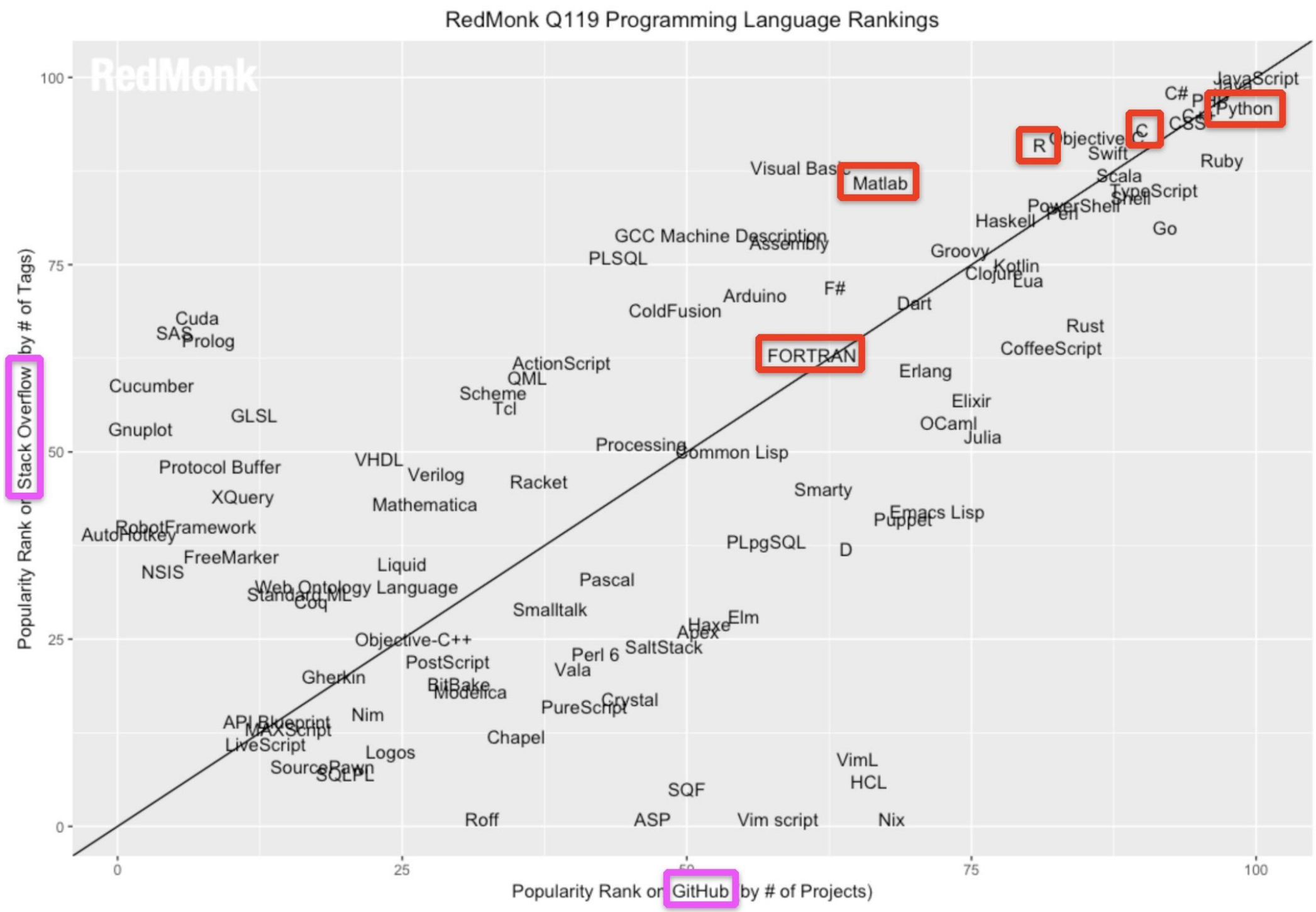
Python

Python

- popular high-level language



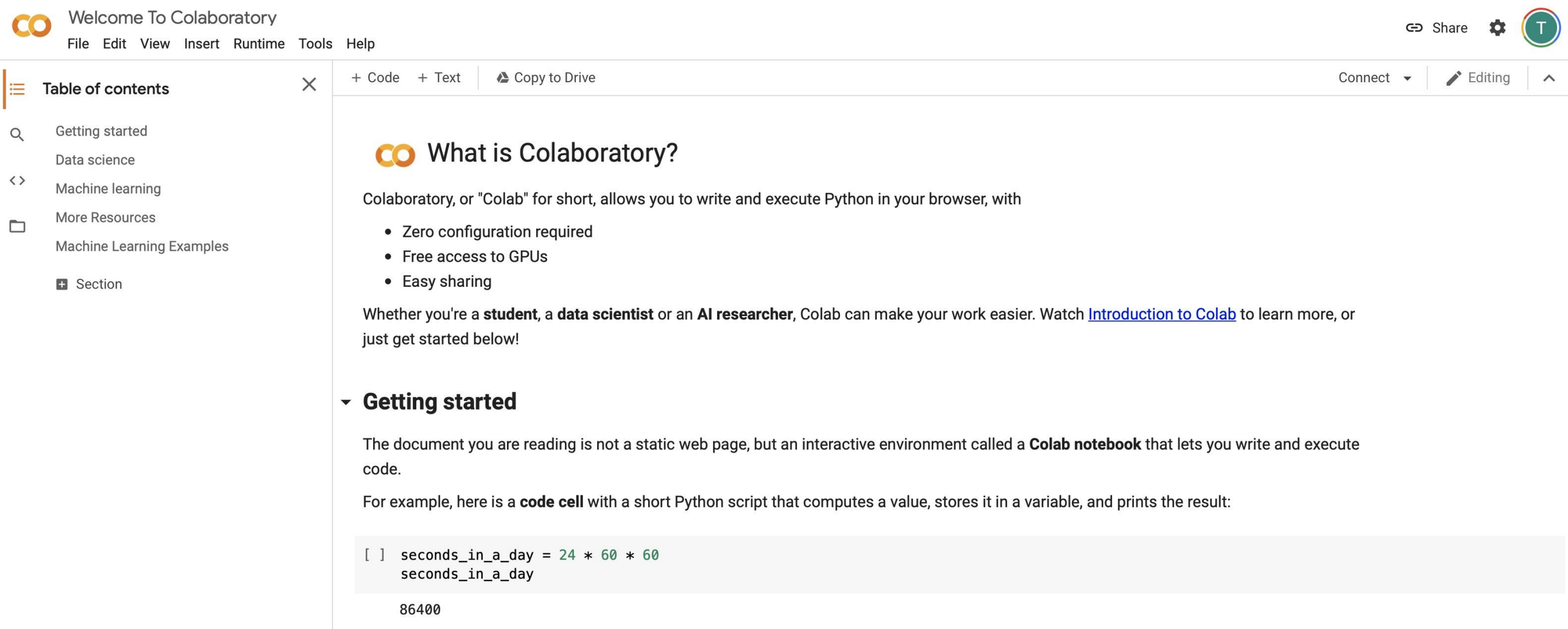
Python

- popular high-level language
- interpreted (rather than compiled) language
- allows for full programming (.py)
- as well as scripts and notebooks (Jupyter or Colab)
- free, open source
- many powerful IDEs available
- thousands of available modules and packages
- scipy, numpy, matplotlib
- keras and tensorflow for neural networks
- scikit-learn (and others) for computational analyses

Google Colab

Google Colab

https://colab.research.google.com/



Julia, Python, R

Jupyter Notebook that runs on Google's cloud services

we will talk more about how to use Jupyter Notebooks

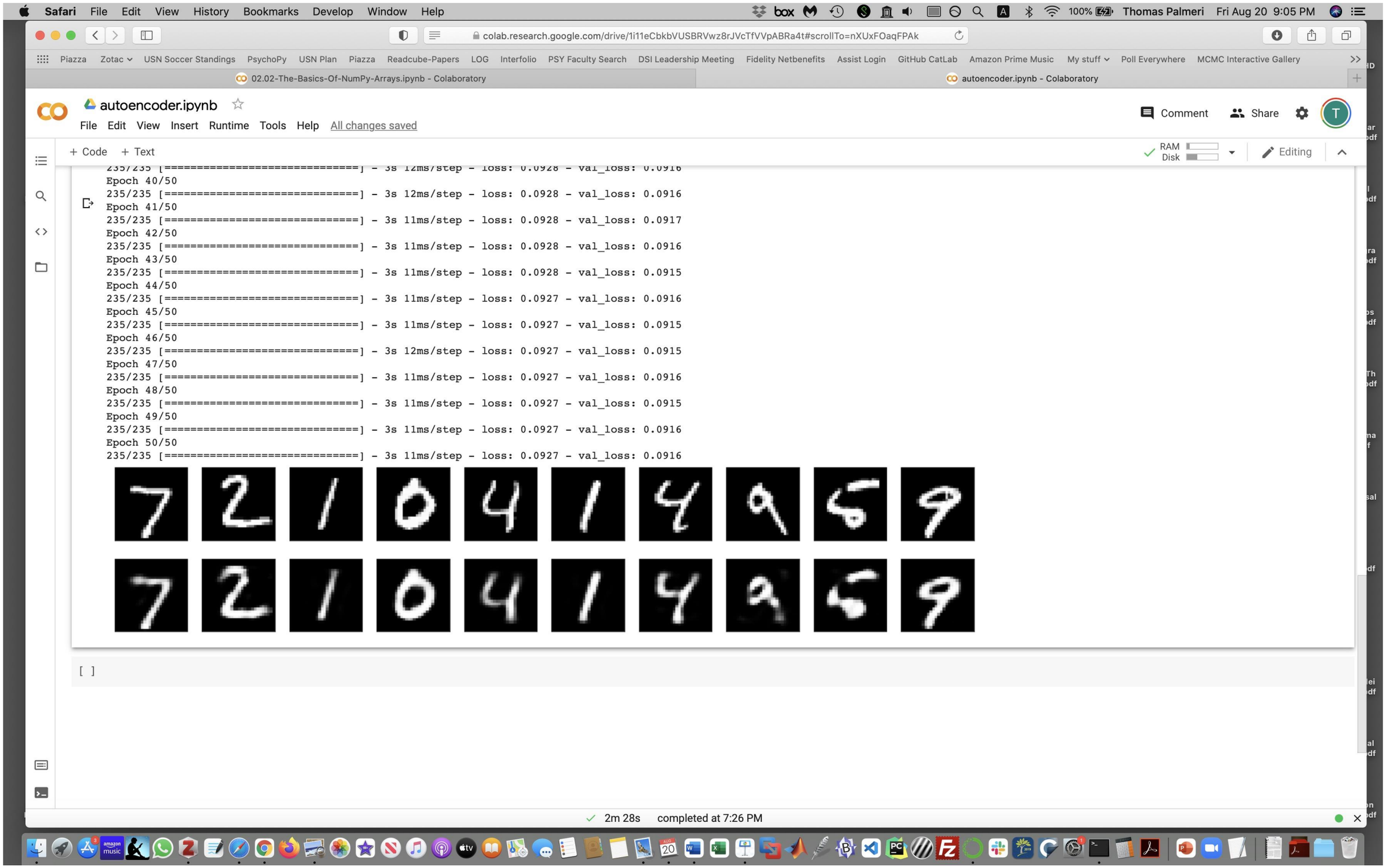
Google Colab

https://colab.research.google.com/

- **as part of Homework 1** :
  - download `autoencoder.ipynb` from Brightspace
  - File -> Upload Notebook
  - click on the Python code cell
  - hit shift return (or shift-enter) to run
  - or click
  - it will take a few minutes to train the network
  - capture a screen shot of the output and submit with other screen shots (as a ZIP file) on Brightspace

Google Colab

https://colab.research.google.com/



Google Colab

https://colab.research.google.com/

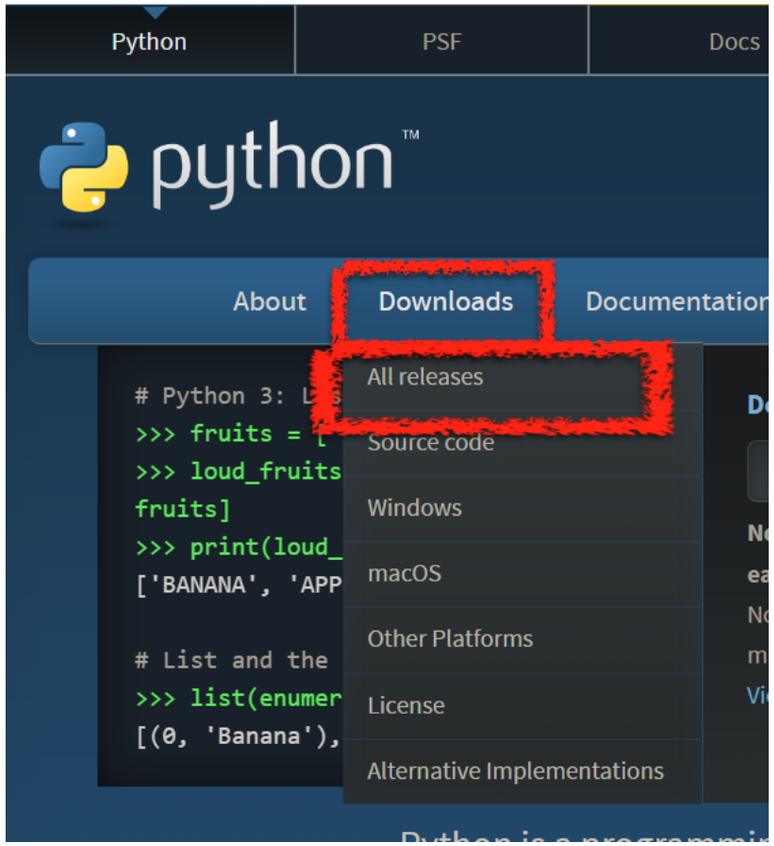
- a backup in case your local Python does not work
- or in case your computer stops working
- something to use if you need to learn (relearn) Python until you get Python installed and running on your computer
- Colab is free for some amount of usage but then you need to start paying for usage
- may be useful to use Google Colab for later assignments that might take too long to run on your computer (Colab has access to powerful GPUs & TPUs on Google cloud service)

Fresh Python installation and setup

even if you already have another version
of Python on your computer

install Python 3.9

- go to python.org
- click on Downloads -> All releases -> Python 3.9.13
- selection the version under “Files” that matches your OS

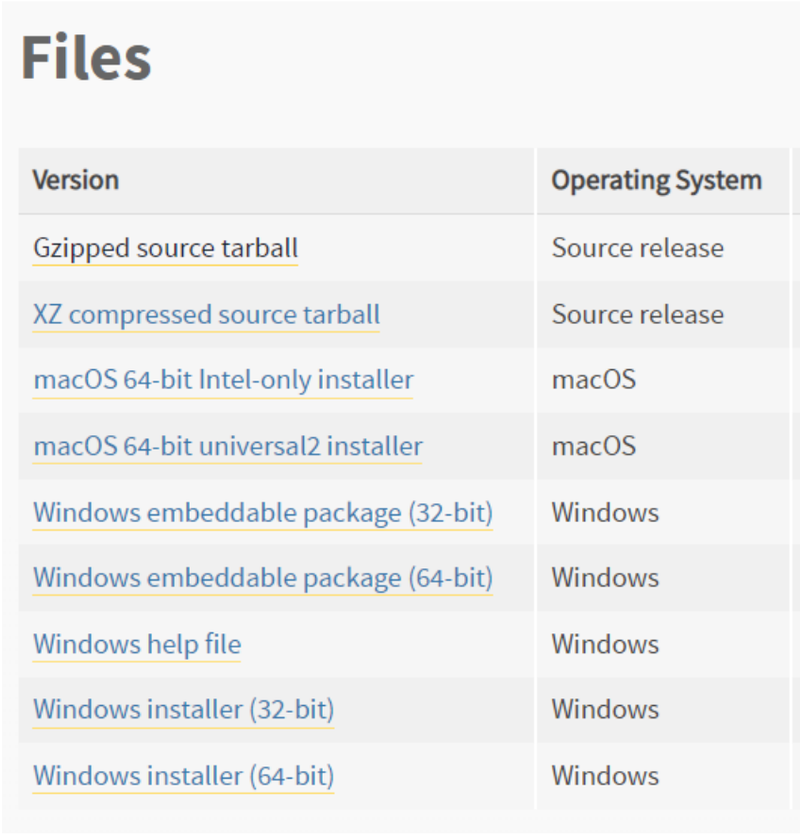


Looking for a specific release?

Python releases by version number:

Release version	Release date
Python 3.10.6	Aug. 2, 2022
Python 3.10.5	June 6, 2022
Python 3.9.13	May 17, 2022
Python 3.10.4	March 24, 2022
Python 3.9.12	March 23, 2022
Python 3.10.3	March 16, 2022
Python 3.9.11	March 16, 2022

View older releases



Files

Version	Operating System
Gzipped source tarball	Source release
XZ compressed source tarball	Source release
macOS 64-bit Intel-only installer	macOS
macOS 64-bit universal2 installer	macOS
Windows embeddable package (32-bit)	Windows
Windows embeddable package (64-bit)	Windows
Windows help file	Windows
Windows installer (32-bit)	Windows
Windows installer (64-bit)	Windows

install Python (on a PC)

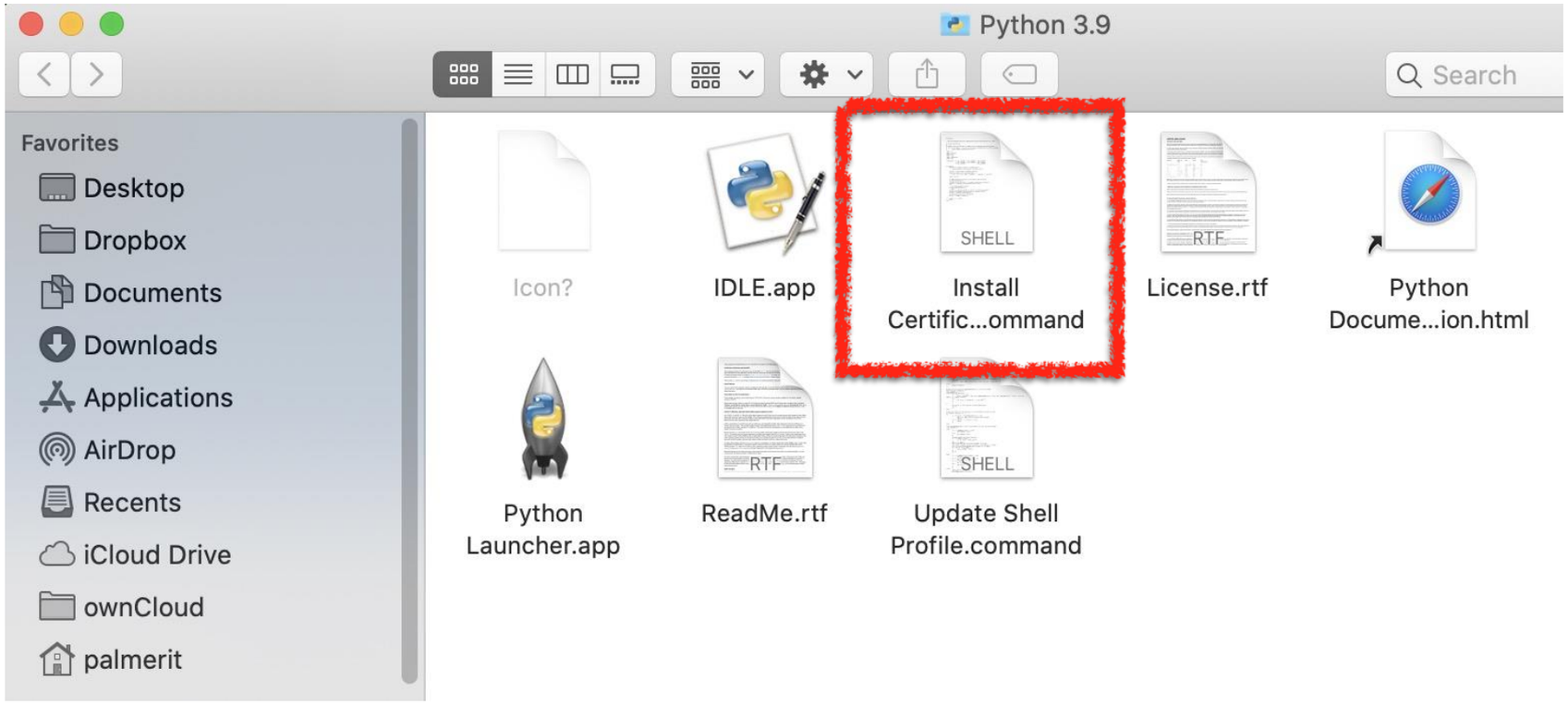
- On Windows, when installing Python, check the box at the bottom of the installer window that says “Add Python 3.9 to PATH”
- This will let you use the “python” command in your powershell or command prompt. Otherwise you’ll need to use the full path of your python.exe installation as the command instead

install Python (on a Mac)

- on a Mac, after installing Python, you will see a note like this:

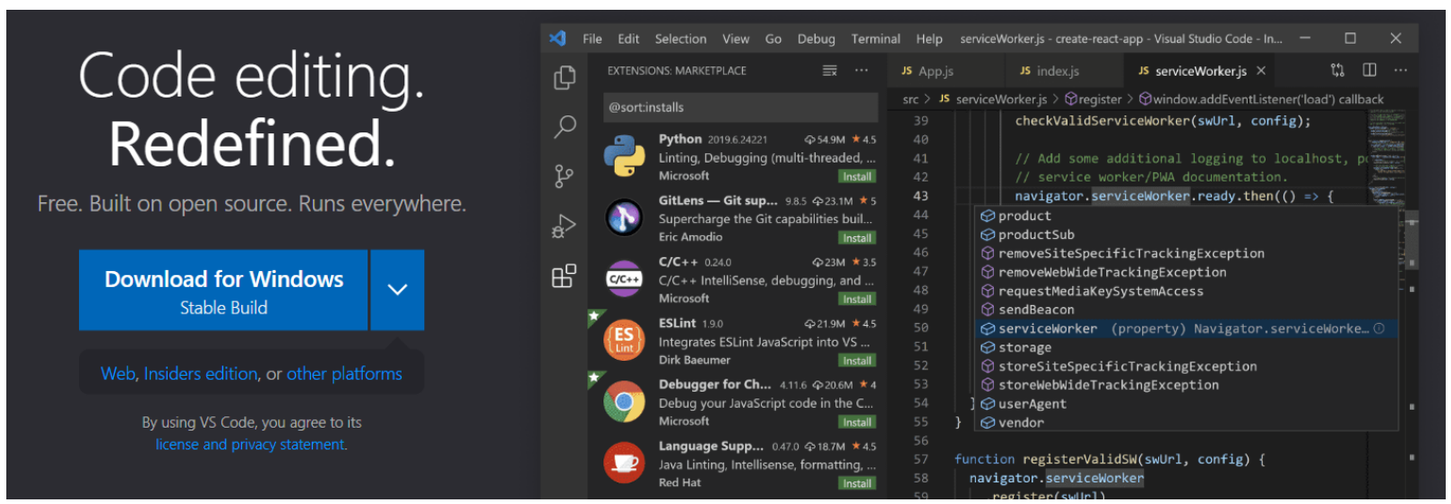
Congratulations! **Python 3.9.13 for macOS 10.9 or later** was successfully installed.

One more thing: to verify the identity of secure network connections, this Python may need a set of SSL root certificates. You can download and install a current curated set from the Certifi project by double-clicking on the Install Certificates icon in the Finder window. See the ReadMe file for more information.



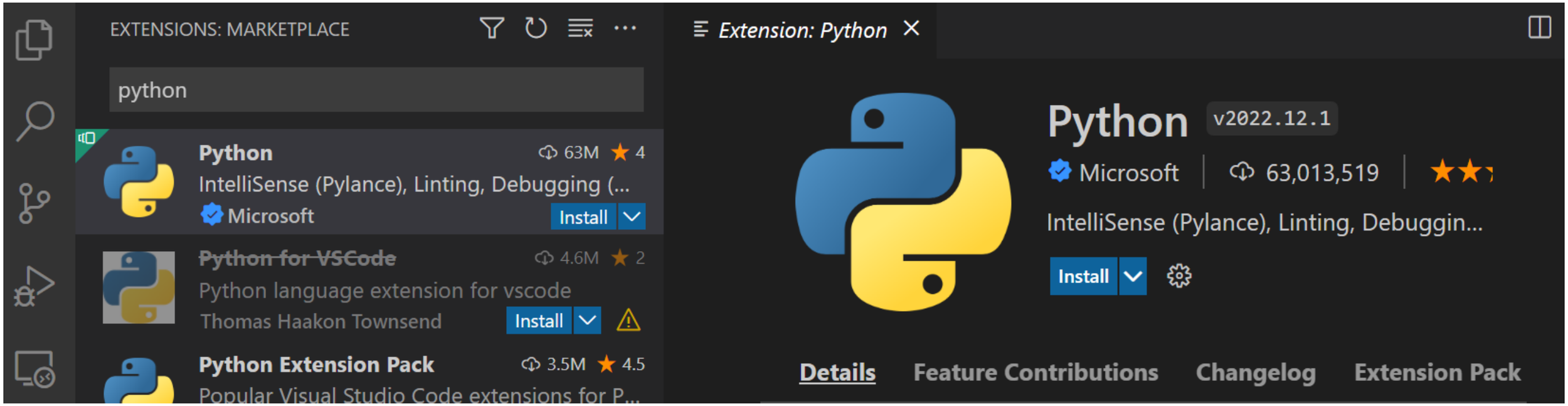
install VSCode IDE

- You can use whatever IDE or code editor you want: PyCharm, VS Code, Eclipse, Spyder, Wing IDE etc.
- I will use VSCode (and Jupyter Notebooks) in class and these notes will assume VSCode for setting up Python
- you can download VSCode from here and install: https://code.visualstudio.com/
- we will talk about how to use VSCode a bit in class, but you are free to learn more from whatever online documentation, videos, and tutorials you can find



Install Python Extension for VSCode

- For full Python language support, you will want to install the Python Extension for VSCode
- Run VSCode. On the opening screen, click **View** then select **Extensions.** Type in **Python** and install the top result (published by Microsoft).



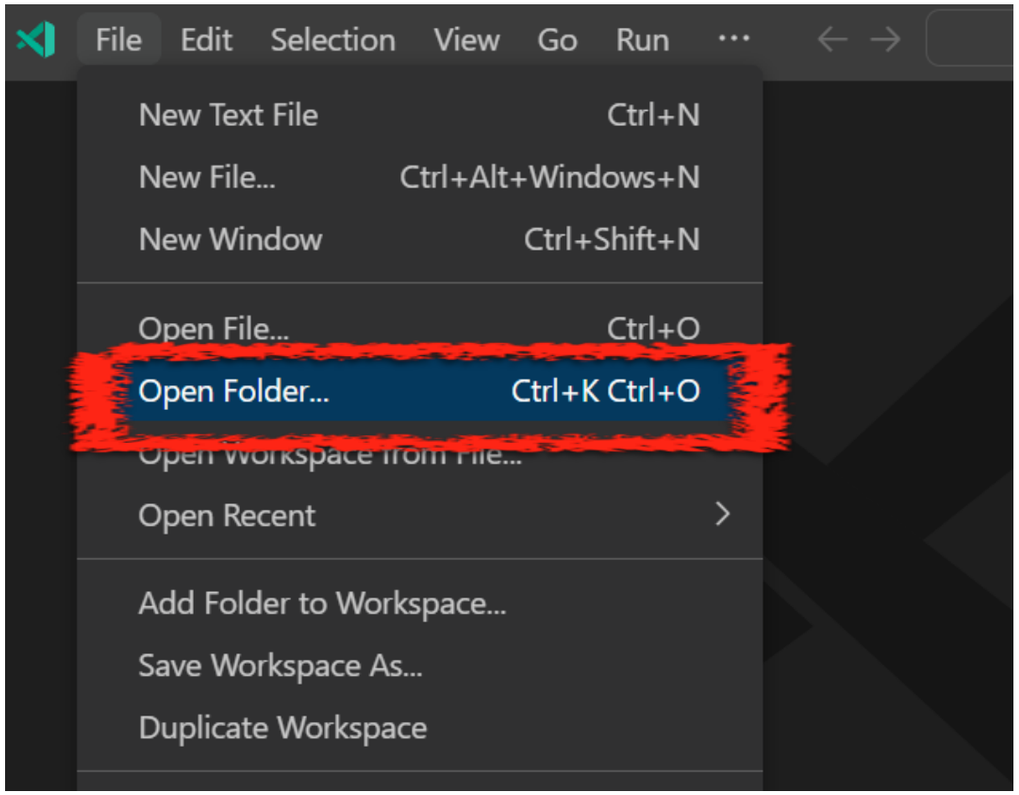
setting up Python environment within VSCode

- the easiest way to configure Python is to keep everything for this course (Python .py and .ipynb files and the environment folders) in a single folder (and subfolders) - you can organize things differently, but you will need to know how to configure things yourself

- **let's assume you create a folder** called `NSC3270` (of course, you can name it whatever you want)

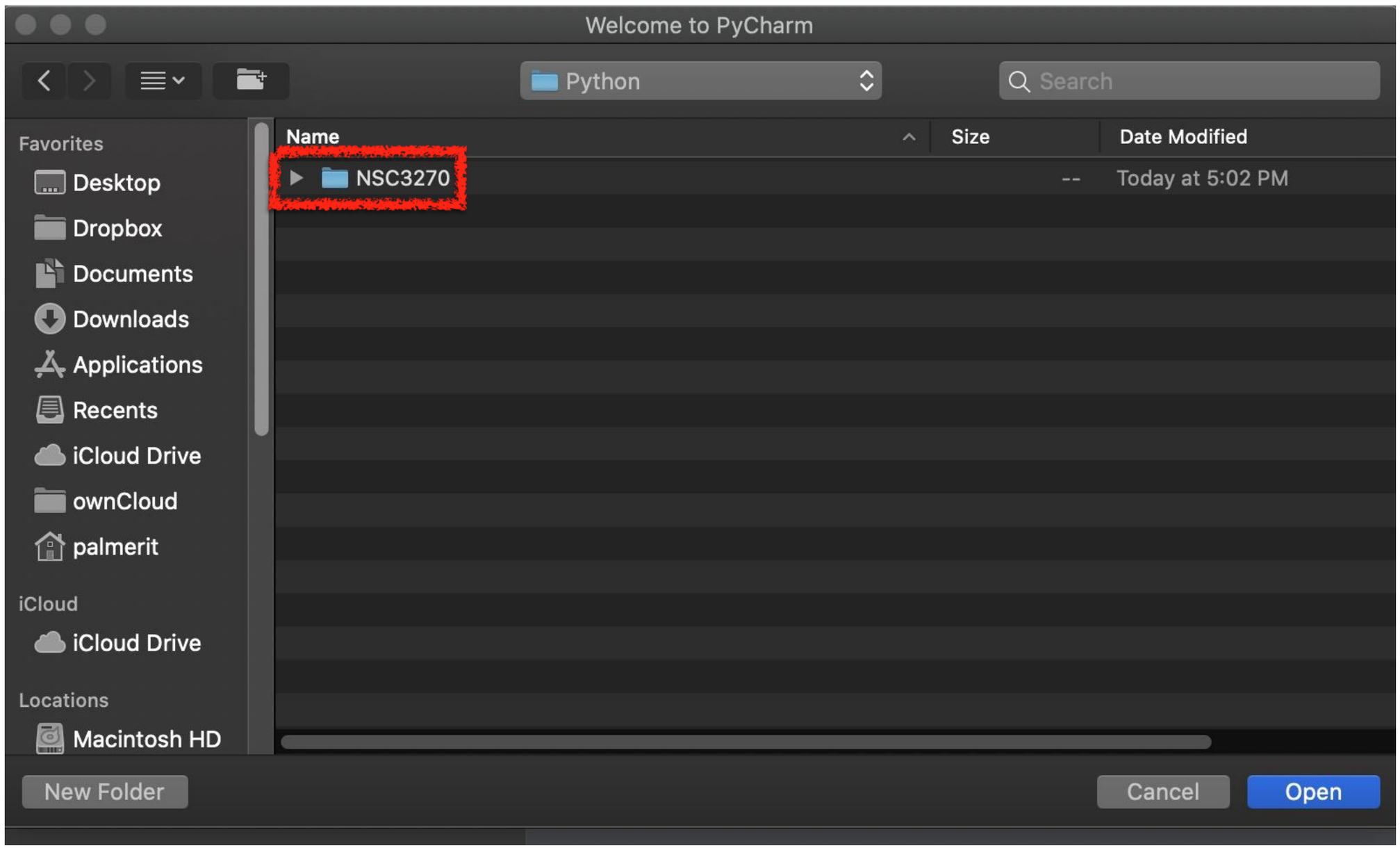
setting up Python environment within VSCode

- run VSCode
- on the opening screen, click File, select Open Folder, and then finally select `NSC3270`



setting up Python environment within PyCharm

- run PyCharm
- on the opening screen, click Open and select `NSC3270`



setting up Python environment within VSCode

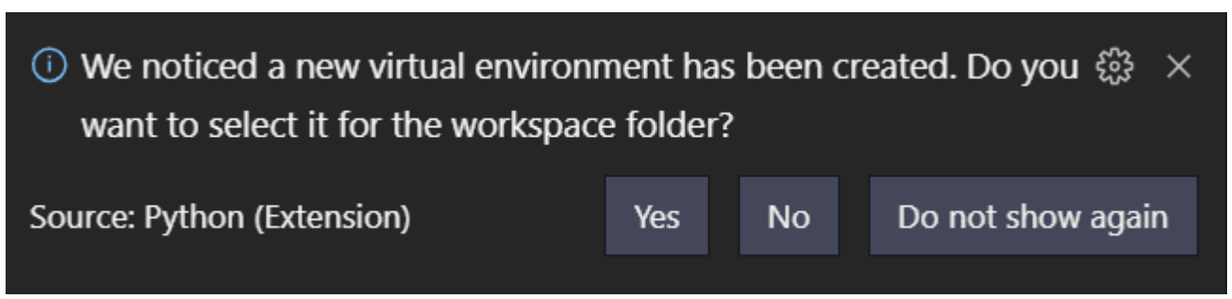
- now we create a virtual environment in VSCode
- virtual environments let you have different configurations of Python for different purposes on the same computer (e.g., a configuration needed for NSC3270, maybe a completely different configuration for a project that's part of your honors thesis), defaulting to perhaps different versions of Python and different collections of modules and packages

setting up Python environment within VSCode

- With your workspace folder opened in VSCode, to the **View** menu and select **Terminal**
- Execute one of the following commands depending on your operating system:

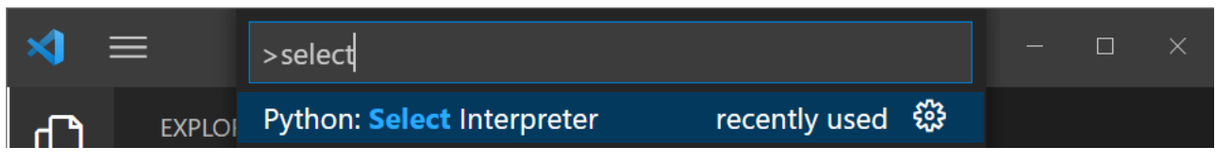
```
# macOS/Linux
python3 -m venv .venv

# Windows
python -m venv .venv
```

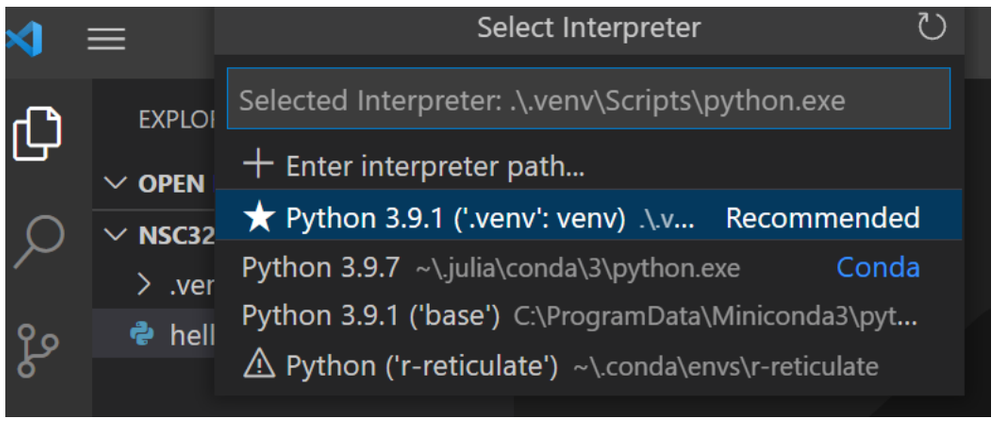
- When you create a new virtual environment, a prompt is usually displayed to allow you to select it for the workspace. Select Yes.



If Python extension does not detect an interpreter..

- VSCode may fail to detect your new virtual environment
- To do it manually, go to the **View** menu and select **Command Palette**
- type “python select” to see and select the option for **Python: Select Interpreter**.
- If things are configured properly, your virtual environment (probably labeled something like **Python 3.9.1 (.venv': venv)** should be one of the available interpreters.





Installing packages within VSCode

- You can install packages to easily use code others have written
- Reopen your **Terminal** from the **View** menu in VSCode
- Activate your virtual environment with one of these two commands depending on your OS:
- Terminal (MacOS): `source .venv/bin/activate`
- Powershell (Windows), two commands at first:
  - `set-executionpolicy RemoteSigned -Scope CurrentUser` (Only needed once! Respond “Yes” (Y) to give Powershell permission to run scripts)
  - `env\scripts\activate.bat` (to activate)
- Execute the command `pip install tensorflow`
- You can follow the same process to install each of the following packages all at once with one line:
  `pip install matplotlib seaborn pandas notebook jupyterlab scikit-learn nipy`

now test that Python installation works in VSCode

test that everything works in VSCode

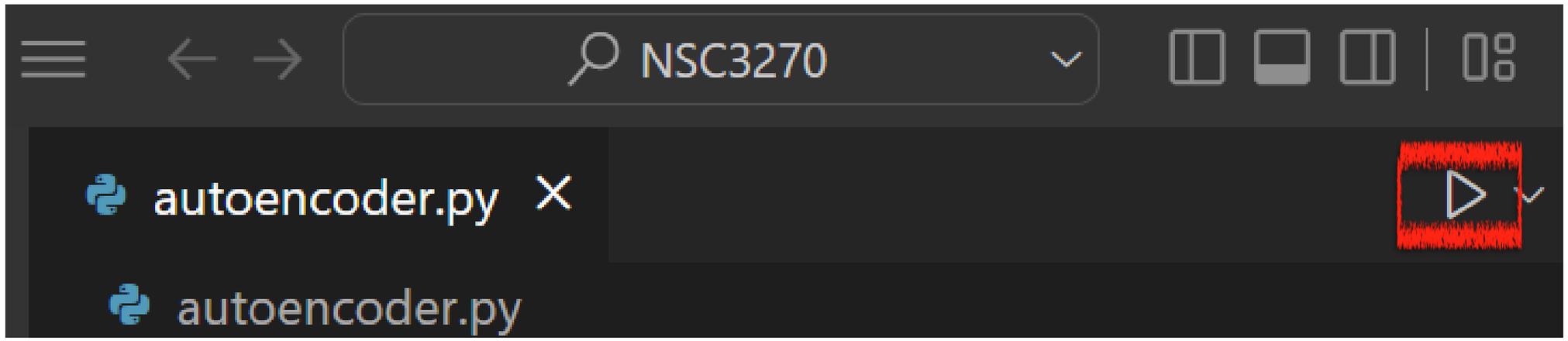
- download `autoencoder.py` from Brightspace and load into VSCode (or another IDE you are using)
- run `autoencoder.py`
- ignore any errors related to CUDA, GPU, or optimization
- you should see it produce something like this:

as part of Homework 1 :

- take a screen shot (make sure it includes the digits)
- submit with other screen shots (as a ZIP file) on Brightspace

How to Run Jupyter Notebooks in VSCode

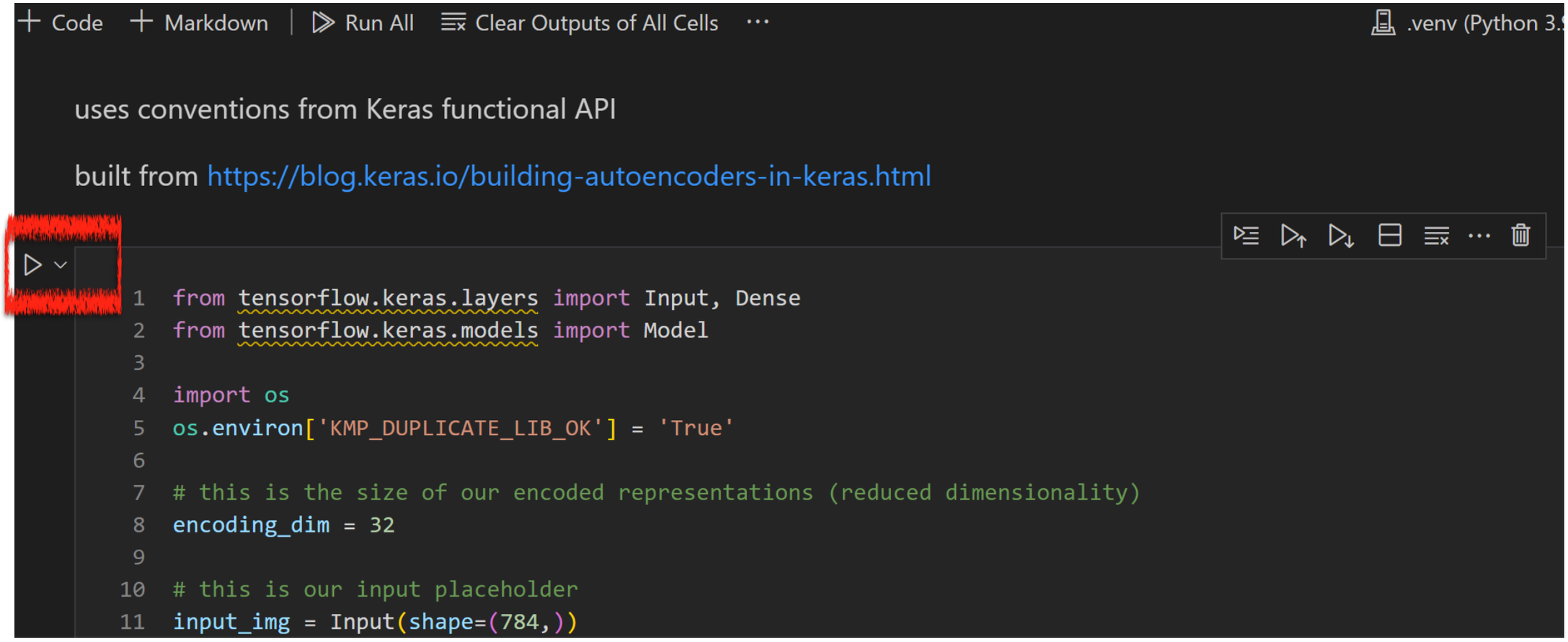
- to run a Python script in VSCode, open the file in your editor



- This should open your terminal and generate some commands and messages within the window activating your virtual environment and executing the script

How to Run Jupyter Notebooks in VSCode

- to run a Python script in VSCode, open the file in your editor
- click on Python code cells
- hit shift-return (or shift-enter) to run
- or click “Run” arrow or “Run all” at top of Editor window.



test Jupyter notebooks

- **as part of Homework 1** :
  - download `autoencoder.ipynb` from Brightspace
  - save to your `NSC3270` course folder
  - open (double-click) within Jupyter notebook
  - click on the Python code cell
  - hit shift-return (or shift-enter) to run
  - it will take a few minutes to train the network
  - capture a screen shot of the output and submit with other screen shots (as a ZIP file) on Brightspace

what if you encounter an error?

if you encounter an error in VSCode or Jupyter

- check Piazza to see if someone had a similar problem
- post a question to Piazza
- make sure you include as much detail as possible (including the entire error message) and note how you installed Python and what IDE you're using
- you can also search online using the error message as a search term - that's what Jordan or I will do if we do not recognize the source of the error
- help out by answering questions posed by others on Piazza

if you encounter an error in VSCode or Jupyter

- if you cannot get an answer on Piazza, then email Jordan jordan.gunn@vanderbilt.edu
- you will need to bring your computer in to meet with Jordan
- if anyone finds an error or omission in the slides, please email both me (sean.polyn@vanderbilt.edu) and Jordan
- remember that you can use Google Colab until you get your local Python working on your computer