Multiple python IDEs (editors)
Google colab for python notebooks is…

- Easiest
- No download needed
- Automatically saves to your google drive

Using a desktop IDE provides more flexibility

Jupyter Notebook Python (and R)

- Lab computers:
  - have Jupyter Notebook (we will use this for Python IDE)
- On your personal computer:
  - If you already have Jupyter Notebook and R that should be sufficient.
  - **If you don't, <u>what to install</u>?**
    - **There are multiple options, no one right option.**
    - **Next slide gives you instructions for Anaconda navigator, one of the option**

INSTALLATION: BASIC
this slide sufficient for Jupyter NoteBook

- Anaconda navigator INSTALLATION: Anaconda is a Python Package Manager with built-in packages useful for data analysis. What is a package manager?
  - Step by Step Tutorial For Windows: https://www.datacamp.com/community/tutorials/installing-anaconda-windows
  - Step by Step Tutorial For Mac: https://www.datacamp.com/community/tutorials/installing-anaconda-mac-os-x
- Jupyter Notebook and Sypder IDEs can be directly launched in base environment

Python CODING: basics

- Good tutorial for beginners
  - https://www.learnpython.org/

Additional Info
Package installation in conda

- General guide https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html
  - Open Anaconda prompt
  - Create new environment
  - Activate environment
  - Install package

Example – gym installation 9b_sample_gym_env1.ipynb

When installing package dependencies, to avoid conflicts with other packages, do not install in base environment; you can try to but may run into errors if there are conflicts in dependencies; first create a new env installing necessary packages activate it; and then run code in created environment

To create new environment using spec file (that has a list of all packages for that environment)
$conda create --name <myenv> --file <my spec file>
$conda activate --name <myenv>

To create new environment, install packages, and create your own spec file
$conda create --name <myenv>
$conda install -c conda-forge gym-box2d #install all packages as usual
To create this spec list as a file in the current working directory, run:
$ conda list --explicit > spec-file.txt

To delete myenv after done
$ conda env remove --name <myenv>
To simply deactivate and return to base use (this will allow you to come back to this environment the next time you need this)
$ conda deactivate

Useful links for gym
#https://www.gymlibrary.dev/environments/box2d/
#https://www.gymlibrary.dev/environments/classic_control/
#https://www.gymlibrary.dev/api/utils/
#making own custom environment :
https://www.gymlibrary.dev/content/environment_creation/

Goto Anaconda command window
Some tools to keep updated generally
$ conda update anaconda
$ conda update setuptools

T0 go back to environment after deactivating it, first activate environment and then run the py code
$conda info –envs #lists all environments you have previously created
$conda activate <myenv> #activates <myenv>
$python sample_gym_env.py

Conda environment create and activate
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Additional Info
Package installation in conda

- General guide https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html
  - Open Anaconda prompt
  - Create new environment
  - Activate environment
  - Install package
- Example: R studio
  - https://anaconda.org/r/rstudio
  - Using with Jupyter Notebook for Rstudio
  - https://docs.anaconda.com/anaconda/navigator/tutorials/r-lang/
- Example: Mesa package for agent-based simulation (equivalent of Netlogo for Python):
  - Open Anaconda prompt
  - Make sure using updated conda version “conda update conda”
  - Create environment “conda create --name myenv” myenv is user given name
  - Activate environment “conda create --name myenv“
  - Install mesa “install -c conda-forge mesa”

Additional Info
Package installation in conda
(you can visit this part if we get to this stage)

- Example: Mesa package for agent-based simulation (equivalent of Netlogo for Python):
  - Open Anaconda prompt
  - Make sure using updated conda version “conda update conda”
  - Create environment “conda create --name myenv” myenv is user given name
  - Activate environment “conda create --name myenv“
  - Install mesa “install -c conda-forge mesa”
- 4.Python + Netlogo interfacing
  - Python Netlogo https://pynetlogo.readthedocs.io/en/latest/
  - Python Netlogo https://opensourcelibs.com/lib/nl4py
  - Netlogo Python https://ccl.northwestern.edu/netlogo/docs/py.html