

**Author:** Eric Gibbons

**Subject:** Using Jupyter Notebooks

**Date:** September 2, 2025



## How to Use Jupyter Notebooks: A Student Guide

Jupyter Notebooks are a powerful tool for data analysis, visualization, and sharing your work. They are a great way to combine code (in this case, Python), text, and visualizations in a single document. This section will provide a brief overview of how to use Jupyter Notebooks to complete your assignments.

### Completing your assignment

1. Navigate to the directory where the assignments notebooks are located.
2. Open Jupyter Notebook on your local machine or server.
  - (a) If you are using a local installation, open a terminal and run the command: `jupyter notebook`
  - (b) You can also use Jupyter Lab by running the command: `jupyter lab`
  - (c) Alternatively, you can use VS Code to edit and run Jupyter Notebooks.
3. Open the notebook you need to work on.
4. Edit the notebook as needed.
  - (a) Read the instructions carefully.
  - (b) Add code where directed. Delete the `# YOUR CODE HERE` and `raise NotImplementedError()` lines and replace it with your code.
  - (c) DO NOT ADD ADDITIONAL CELLS OR REMOVE EXISTING CELLS. THIS WILL CAUSE THE AUTOGRADER TO FAIL, AND YOU WILL RECEIVE A ZERO FOR THE ASSIGNMENT.
  - (d) You may modify test cells, but do not delete them. When I run the autograder, contents of the test cells will be replaced with the correct tests. For many test cells I have added visualizations to help you understand the problem and gain insight into your solution.
  - (e) Many answers you will provide will be numerical. Make sure to round to the appropriate number of decimal places. I generally set an appropriate tolerance for the tests based on rounding errors.
  - (f) Oftentimes the answer will be represented in a functional form. E.g., if you solve a problem and the answer is

$$y(t) = \begin{cases} t & 0 \leq t \leq 1 \\ -t + 2 & 1 < t \leq 2 \\ 0 & \text{otherwise} \end{cases}.$$

You can represent this with a Python function in your notebook:

```
def y(t):
    if 0 <= t <= 1:
        return t
    elif 1 < t <= 2:
```

```
        return -t + 2
    else:
        return 0
```

The upside to this approach is that you can use this function to generate a plot to verify it matches your intuition!

5. Make sure to run all the code cells in the notebook to generate the output and there are no errors.
6. Save your work frequently to avoid data loss.
7. Once you have completed the assignment, make sure to save and close the notebook.
8. Commit and push your changes to the repository to submit your work.

### Common issues

- Due dates are firm. After the deadline passes, you will lose the ability to submit your work. There are no extensions.
- Do not modify the notebook in any way that is not explicitly allowed. This includes adding or removing cells, changing the order of cells, or changing the contents of the test cells.
- Do not change the cell type. In Jupyter, each cell has a type (code, markdown, etc.). Do not change the type of any cell. This will cause the autograder to fail.
- Do not expect me to debug why your notebook is failing the autograder.