

**Download** (right-click, save target as ...) this page as a jupyterlab notebook from:

[Laboratory 1](#)

---

# Laboratory 1: A Notebook Like No Other!

**Medrano, Giovanni**

**R11521018**

ENGR 1330 Laboratory 1 - In-Lab

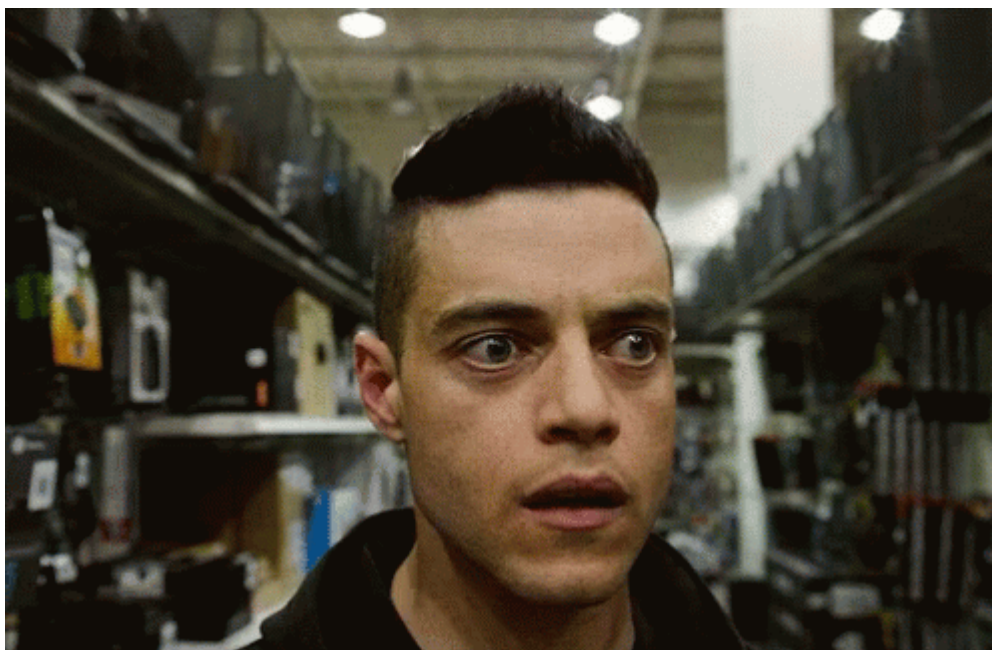
Welcome to your first (or second) Jupyter Notebook. This is the medium that we will be using throughout the semester.

---

**Why is this called a notebook?** Because you can write stuff in it!

**Is that it?** Nope! you can **write** and **run** CODE in this notebook! Plus a bunch of other cool stuff such as making graphs, running tests and simulations, adding images, and building documents (such as this one!).

## The Environment - Let's have a look around this window!



\*Rami Malek in Mr. Robot\*

- The tabs:
  - File
  - Edit

- View
  - Insert
  - Cell
  - Kernel
  - The Icons:
    - Save
    - Insert Cell Below
    - Cut
    - Copy
    - Paste Cells Below
    - Move Up
    - Move Down
    - Run
    - Interrupt Kernel
    - Restart Kernel
    - Cell Type Selector (Dropdown list)
- 

The notebook consists of a sequence of cells. A cell is a multiline text input field, and its contents can be executed by using Shift-Enter, or by clicking Run in the menu bar. The execution behavior of a cell is determined by the cell's type.

There are three types of cells: code cells, markdown cells, and raw cells. Every cell starts off being a code cell, but its type can be changed by using a drop-down on the toolbar (which will be "Code", initially).

## Code Cells:

A code cell allows you to edit and write new code, with full syntax highlighting and tab completion. The programming language you use depends on the kernel. What we will use for this course and the default kernel IPython runs, is Python code.

When a code cell is executed, code that it contains is sent to the kernel associated with the notebook. The results that are returned from this computation are then displayed in the notebook as the cell's output. The output is not limited to text, with many other possible forms of output are also possible, including matplotlib figures and HTML tables. This is known as IPython's rich display capability.

## Markdown Cells:

You can document the computational process in a literate way, alternating descriptive text with code, using rich text. In IPython this is accomplished by marking up text with the Markdown language. The corresponding cells are called Markdown cells. The Markdown language provides a simple way to perform this text markup, that is, to specify which parts of the text should be

emphasized (italics), bold, form lists, etc. In fact, markdown cells allow a variety of cool modifications to be applied:

If you want to provide structure for your document, you can use markdown headings. Markdown headings consist of 1 to 5 hash # signs followed by a space and the title of your section. (The markdown heading will be converted to a clickable link for a section of the notebook. It is also used as a hint when exporting to other document formats, like PDF.) Here is how it looks:

# # title

## ## major headings

### ### subheadings

#### #### 4th level subheadings

#### ##### 5th level subheadings

These codes are also quite useful:

- Use triple " " *before and after a word (without spacing) to make the word bold and italic*  
B&I: ***string***
- or **\*\* before and after a word (without spacing) to make the word bold**  
**Bold:** **string** or **string**
- *\_ or before and after a word (without spacing to make the word italic*  
*Italic:* *string* or *string*
- Double ~ before and after a word (without spacing to make the word scratched  
Scratched: ~~string~~
- For line breaks use "br" in the middle of <>
- For colors use this code:

change this to a merkdwn cell and run it!

Text

Text

Text

- For indented quoting, use a greater than sign (>) and then a space, then type the text. The text is indented and has a gray horizontal line to the left of it until the next carriage return.

here is an example of how it works!

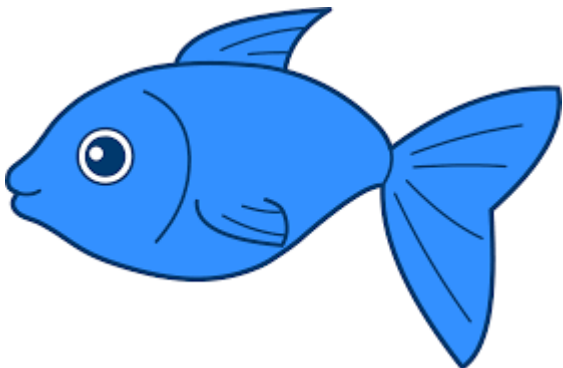
- For bullets, use the dash sign ( - ) with a space after it, or a space, a dash, and a space ( - ), to create a circular bullet. To create a sub bullet, use a tab followed a dash and a space. You can also use an asterisk instead of a dash, and it works the same.
- For numbered lists, start with 1. followed by a space, then it starts numbering for you. Start each line with some number and a period, then a space. Tab to indent to get subnumbering.

1. first
2. second
3. third
4. ...

- For horizontal lines: Use three asterisks: \*\*\*



- For graphics, you can attach image files directly to a notebook only in Markdown cells. Drag and drop your images to the Markdown cell to attach it to the notebook.



Raw Cells:

Raw cells provide a place in which you can write output directly. Raw cells are not evaluated by the notebook.

This is a raw cell

**Let's meet world's most popular python!**



## What is python?

"Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace." - Wikipedia @

[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

## How to have access to it?

There are plenty of ways, from online compilers to our beloved Jupyter Notebook on your local machines. Here are a few examples of online compilers:

- <https://www.programiz.com/python-programming/online-compiler/>
- [https://www.onlinegdb.com/online\\_python\\_compiler](https://www.onlinegdb.com/online_python_compiler)
- [https://www.w3schools.com/python/python\\_compiler.asp](https://www.w3schools.com/python/python_compiler.asp)

d. <https://repl.it/languages/python3>

We can do the exact same thing in this notebook. But we need a CODE cell.

```
In [26]: x = 3  
         print(x)
```

3

This is the classic "first program" of many languages! The script input is quite simple, we instruct the computer to print the literal string "hello world" to standard input/output device which is the console. Let's change it and see what happens:

```
In [27]: print("This is my first notebook!")
```

This is my first notebook!

## How to save a notebook?

- **As a notebook file (.ipynb):**

Go to File > Download As > Notebook (.ipynb)

- **As an HTML file (.html):**

Go to File > Download As > HTML (.html)

- **As a Pdf (.pdf):**

Go to File > Download As > PDF via LaTeX (.pdf) or Save it as an HTML file and then convert that to a pdf via a website such as <https://html2pdf.com/>

*Unless stated otherwise, we want you to submit your lab assignments in PDF and your exam and project deliverables in both PDF and .ipynb formats.*

---

## Readings

*This notebook was inspired by several blogposts including:*

- **"Markdown for Jupyter notebooks cheatsheet"** by **Inge Halilovic** available at [\\*https://medium.com/@ingeh/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed](https://medium.com/@ingeh/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed)
- **"Jupyter Notebook: An Introduction"** by **Mike Driscoll** available at [\\*https://realpython.com/jupyter-notebook-introduction/](https://realpython.com/jupyter-notebook-introduction/)

*Here are some great reads on this topic:*

- **"Jupyter Notebook Tutorial: The Definitive Guide"** by **Karlijn Willems** available at [\\*https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook](https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook)
- **"Introduction to Jupyter Notebooks"** by **Quinn Dombrowski, Tassie Gniady, and David Kloster** available at [\\*https://programminghistorian.org/en/lessons/jupyter-notebooks](https://programminghistorian.org/en/lessons/jupyter-notebooks)
- **"12 Things to know about Jupyter Notebook Markdown"** by **Dayal Chand Aichara** available at [\\*https://medium.com/game-of-data/12-things-to-know-about-jupyter-notebook-markdown-3f6cef811707](https://medium.com/game-of-data/12-things-to-know-about-jupyter-notebook-markdown-3f6cef811707)



Here are some great videos on these topics:

- **"Jupyter Notebook Tutorial: Introduction, Setup, and Walkthrough"** by **Corey Schafer** available at [\\*https://www.youtube.com/watch?v=HW29067qVWk](https://www.youtube.com/watch?v=HW29067qVWk)
  - **"Quick introduction to Jupyter Notebook"** by **Michael Fudge** available at [\\*https://www.youtube.com/watch?v=jZ952vChhul](https://www.youtube.com/watch?v=jZ952vChhul)
  - **"What is Jupyter Notebook?"** by **codebasics** available at [\\*https://www.youtube.com/watch?v=q\\_BzsPwxLOE](https://www.youtube.com/watch?v=q_BzsPwxLOE)
- 

## Exercise: Let's see who you are!

Similar to the hello world example, use a code cell and print a paragraph about you. You can introduce yourselves and write about interesting things to and about you! A few lines below can get you started, replace the ... parts and the other parts to make your paragraph.

```
In [28]: print('my name is Giovanni Medrano')
         print('my favorite food is vietnamese pho')
         print('I am currently studying to be an electrical engineer')
         print('I speak 3 languages, they are: english, spanish, and of course profanity')
```

```
my name is Giovanni Medrano
my favorite food is vietnamese pho
I am currently studying to be an electrical engineer
I speak 3 languages, they are: english, spanish, and of course profanity
```

Do we leave these two cells blank?

```
In [ ]:
```