Just a Few Days Left!

When: Saturday, April 18th from 10 am - 12 am (MIDNIGHT)

Where: Sitterson and Fred Brooks 1st floor (Floor 0)

- Please confirm your attendance by responding to the official Hack110 email.
- Be on the lookout for an email to join the Hack110 GroupMe
- **Don't want to make a project or stay the full time?** We have plenty of workshops throughout the day to benefit you in multiple ways!
- Still time to sign up. Don't Wait!

Website!

Questions? Email **hack110@office.unc.edu**

Perks

Lunch catered by **Panera**

Dinner catered by **Kolapasi**

Free Hack 110 Shirt

Free Snacks and **Drinks** Throughout

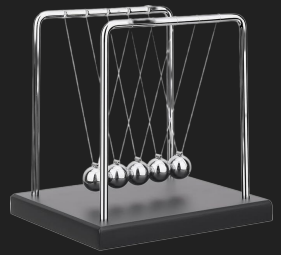
Earn **CLE credit**

Win Vouchers for UNC's best cafe: **Tea Hill**

Compete to **win** many amazing **prizes!**

Select a Prize from the Pool

Laptop Stand

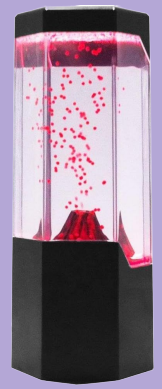


Newton’s Cradle

Portable Fan

Survival Kit

Tile Tracker



Volcano Lamp

Wireless Keyboard

Galaxy Projector

Power Banks

Phone Mount

Analyzing Data from a CSV in a Jupyter Notebook

Get ready to code with me in VSCode,
or take notes on what the code I’m writing does!

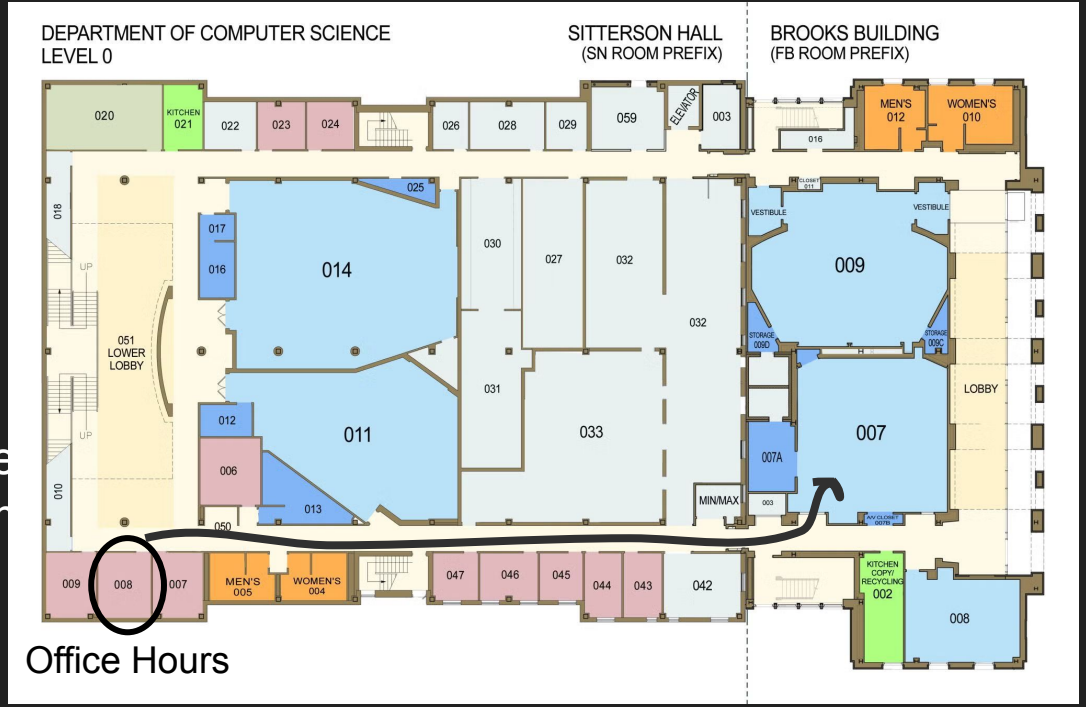
Announcements

Re: Assignments:

- **EX08: Linked List Utility Functions** due at 11:59pm tonight

Quiz 03 this Friday! To prep:

- Review your previous quiz(zes)
- Practice problems on the website
- Hybrid review session today from 6:30-7:30pm in Fred Brooks, room 007 + online



We've covered a LOT in this course thus far…

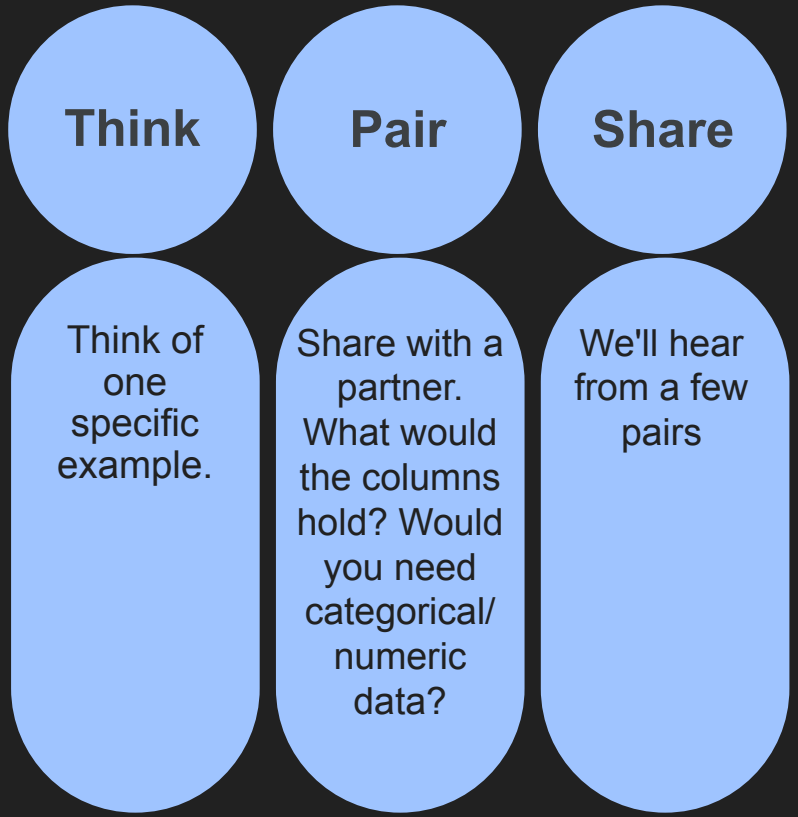
- Objects
- Data types
- Expressions
- Functions
- Memory diagrams
- Boolean expressions
- Conditionals
- Scope
- User input
- While loops
- For loops
- Importing modules
- Lists
- Unit tests
- Dictionaries
- Object-oriented prog.
- Magic methods + operator overloading
- Recursive structures
- Recursive functions/ methods

(and more…)

Now, it's time to take these ideas and put them to use by analyzing real-world data!

Think about your major(s) or a potential career path…

What data might you need to gather, track, or analyze?



EX09: Data Analysis for Continuous Improvement

For your last exercise, you will be analyzing the anonymized data from the class survey (LS16) to make evidence-based recommendations regarding how we can improve the course in the future!

This assignment will require you to:

we'll introduce these today!

- Read a CSV of data in Python
- Use "helper" functions to wrangle the data
- Analyze the data
- Generate figures summarizing the results of your analysis
  - (with the help of a library called **`seaborn`**!)
- Summarize your work and findings on your own website

New skills!

Let’s analyze Raleigh weather data and Walter Wally’s predictions to see how accurate his winter weather forecasts are!

(Walter Wally is Raleigh’s version of Punxsutawney Phil!)

Today's agenda:

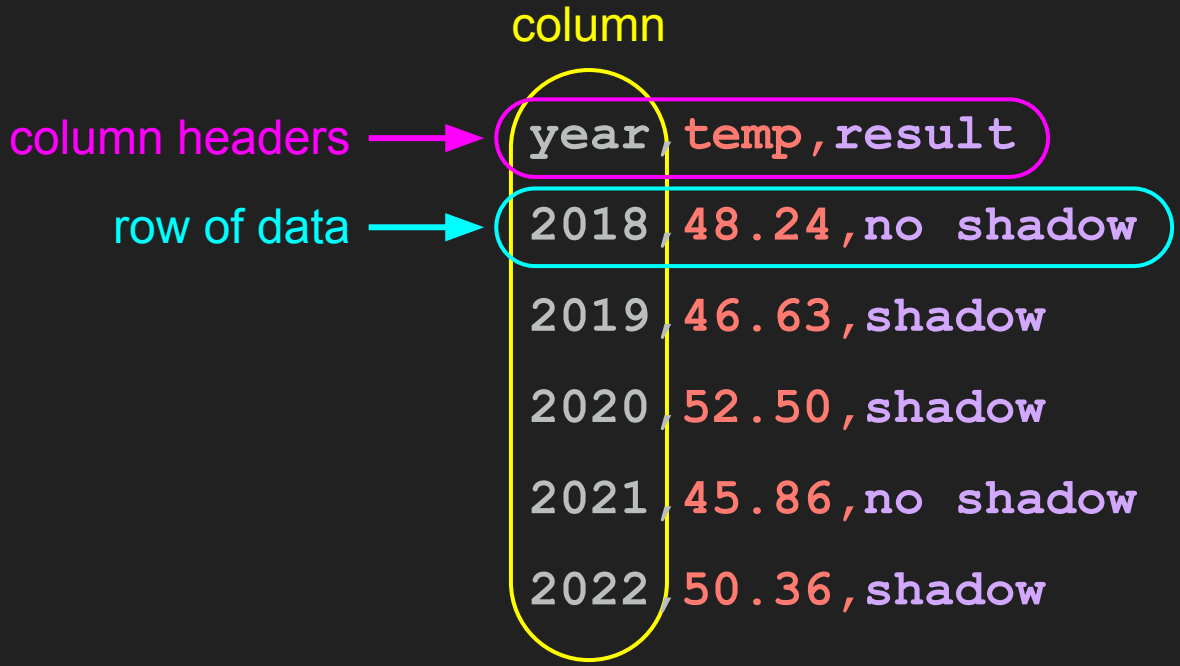
1. Review CSV structure
2. Introduce Jupyter Notebooks
    a. Markdown cells (formatted text)
    b. Python cells (code)
3. Read CSV data
4. Wrangle data

Next week:

1. More data-wrangling
2. Analyze data
3. Read documentation/use Python packages and libraries

What is a CSV?

A .csv file stores tabular data as plain text. Each line is a row, and values in that row are separated by commas.



What is a CSV?

A .csv file stores tabular data as plain text. Each line is a row, and values in that row are separated by commas.

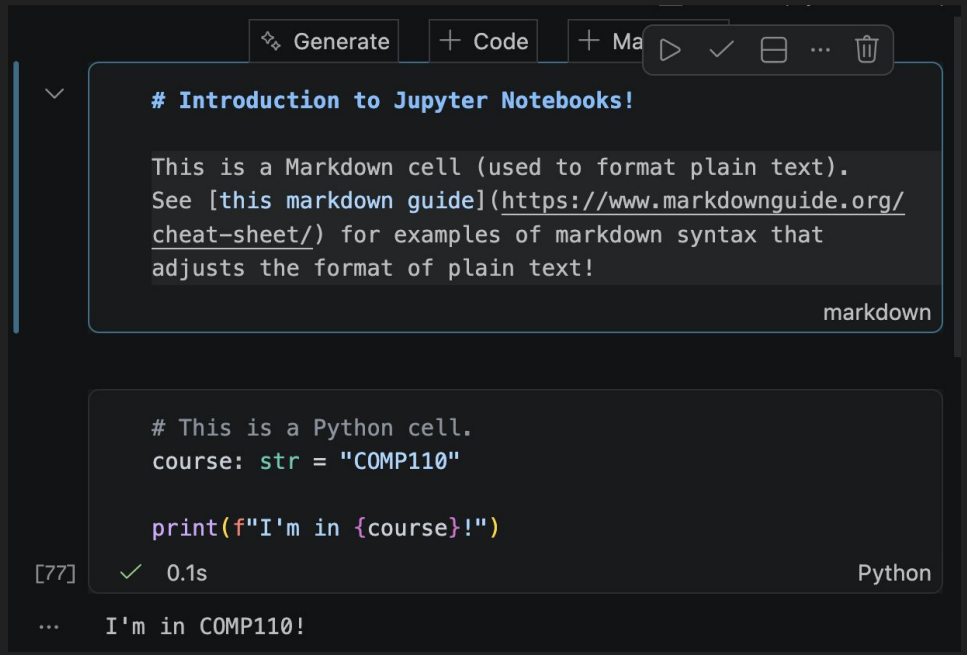
In your VSCode workspace:

1. Create a directory (folder) called **`data_analysis`**
2. In **`data_analysis`**, create a directory called **`data`**
3. In your **`data`** directory, create a file called **`walter_wally.csv`**
4. Copy the data to the right, and paste it into **`walter_wally.csv`**
5. Save the .csv file

```
year,temp,result
2018,48.24,no shadow
2019,46.63,shadow
2020,52.50,shadow
2021,45.86,no shadow
2022,50.36,shadow
```

What is a Jupyter Notebook?

An .ipynb file mixes live Python code, rich text, and output, all in one document. Instead of running an entire script at once, you execute small chunks (“cells”) and see output immediately below each cell.



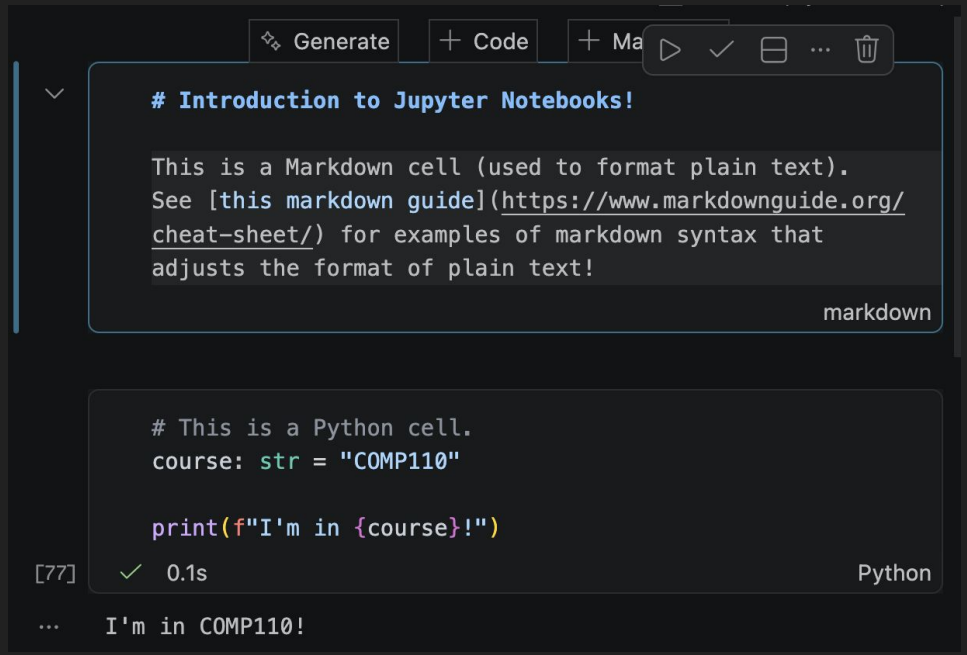
In your VSCode workspace:

1. In **`data_analysis`**, create a new file called **`analysis.ipynb`**

What is a Jupyter Notebook?

Let's try it!

An .ipynb file mixes live Python code, rich text, and output, all in one document. Instead of running an entire script at once, you execute small chunks ("cells") and see output immediately below each cell.



Using Notebooks in VS Code

1. **Open or create an .ipynb file**
   New file... → name it <anything>.ipynb
2. **Select your Python kernel**
   Click the kernel picker (top right) → select a Python version
3. **Add a cell**
   Click "+ Code" or "+ Markdown" in the toolbar
4. **Run a cell**
   Click ▶ or press Shift + Enter
5. **Run all cells**
   Kernel → Restart & Run All (clears old outputs first)

Markdown in Jupyter Notebooks (Cheat Sheet here!)

Syntax	Renders as
`# Heading 1`	**Large section title**
`## Heading 2`	**Subsection title**
`**bold text**`	**bold text**
`*italic text*`	*italic text*
`- item\n- item`	• Bulleted list
`1. first\n2. sec`	1. Numbered list
`` `inline code` ``	`monospace snippet`

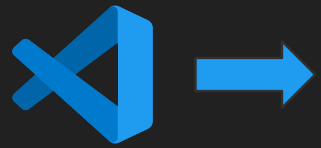
Let’s write a function to help us read and wrangle the data!

`read_csv_rows`

`read_csv_rows` should read an entire CSV of data and return it as a list of rows, where each row represents a `dict[str, str]`.

- Function Name: `read_csv_rows`
- Parameter:
  - `str` path to CSV file
- Return Type: `list[dict[str, str]]`

Let’s call it to read our CSV data and convert them to a `list[dict[str, str]]`



read_csv_rows

Before calling read_csv_rows (data in our .csv):

```
year,temp,result
2018,48.24,no shadow
2019,46.63,shadow
2020,52.50,shadow
2021,45.86,no shadow
2022,50.36,shadow
```

After calling read_csv_rows (Python objects in our .ipynb):

```
[{'year': '2018', 'temp': '48.24', 'result': 'no shadow'}
{'year': '2019', 'temp': '46.63', 'result': 'shadow'},
{'year': '2020', 'temp': '52.50', 'result': 'shadow'},
{'year': '2021', 'temp': '45.86', 'result': 'no shadow'},
{'year': '2022', 'temp': '50.36', 'result': 'shadow'}]
```

`read_csv_rows` (solution)

```
def read_csv_rows(filename: str) -> list[dict[str, str]]:
    """Read the rows of a CSV into a 'table'."""
    result: list[dict[str, str]] = []

    # Open a handle to the data file
    file_handle = open(filename, "r", encoding="utf8")

    # Prepare to read the data file as a CSV rather than just strings.
    csv_reader = DictReader(file_handle)

    # Read each row of the CSV line-by-line
    for row in csv_reader:
        result.append(row)

    # Close the file when done, to free its resources.
    file_handle.close()

    return result
```