

COMPOUTION

For Data Science

+ Diane Woodbridge, PH.D



ABOUT DIANE

Education

Ph.D. (and M.S.) in Computer Science, UCLA. 2012, 2010



Experience

Senior Member of Technical Staff, Sandia National Laboratories.

2012 – 2016

Professor, 2016-Current, MS in Data Science, USF

Co-Founder, StudyStudio.ai. 2024 - Current



Research Interest: Database management and machine learning in health IoT, remote sensing, oral history, etc.

And I also love working with students!



“



About Computation for DS (MSDS 501)

Learning Outcomes

- Through this course, the students will learn computer organization and Python programming for data science. We will practice what a data scientist will use on a daily basis.
 - We will practice shell commands using Terminal.
 - We will use version control via Git and GitHub.
 - We will review core concepts in Python, including data types, conditional statements, loop, file I/O, data aliasing, function, and object oriented programming (OOP)
 - We will also learn concepts of packages/libraries and a few useful ones, including pytest, numpy, etc.
 - Most importantly, we will learn about how we can approach programming as problem-solving.
 - We will practice writing clean code using PEP (Python Enhancement Proposals).



About Computation for DS (MSDS 501)

Class Schedule - Tuesday and Friday

- Section 1 : 10 am - 12 PM
- Section 2 : 1 PM - 3 PM
 - Make sure to attend the session that you are assigned to for better engagement.

Office Hour

- Friday : 3 - 4pm (Room #606)

All student laptops must be closed during class unless we are doing a lab or I specifically ask you to follow along as I type into my computer. All materials for the course are available on Canvas and Github.

No cellphones, social media, slack, or texting during the class.

About Computation for DS (MSDS501)

Class Overview

Day 1

Class Intro, Basic Computer Architecture,
Terminal and Shell Commands

Day 2

Version Control & Git, Programming & Python,
Debugging, Error Handling

Day 3

Python Data Types, Loop

Day 4

Loop , Conditional Statements

Day 5

Function, pytest

Day 6

Python File I/O, Data Aliasing

Day 7

Programming Patterns

Day 8

Packages, Libraries and Modules

Day 9

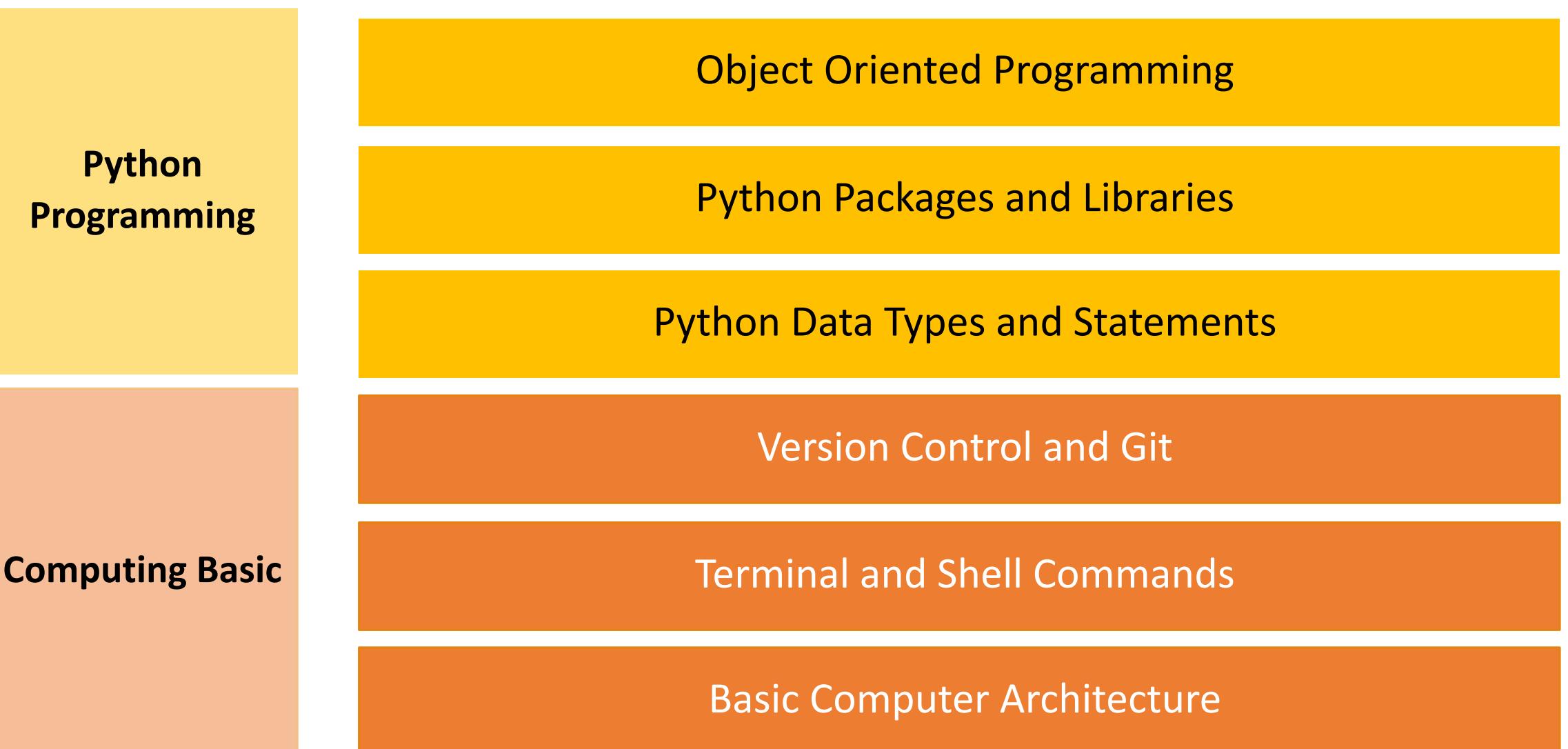
Object Oriented Programming

Day 10

PEP8 and PEP20

Class Summary

Class Overview



Are you a programming expert?

We are lifelong learners 😊

This is an introductory course to make everyone on the same page, and I expect you to master all the materials.

- Please review the materials thoroughly, as doing well in the boot camp courses are essential for proceeding to the program and being placed in your dream practicum.

If you need more advanced materials, please get a Premium [Leetcode](#) account and start practicing.

- Start with [the Learn section](#) and move forward.

About Computation for DS (MSDS501)

Evaluation Criteria

- Attendance and Professionalism - 5 %
- Weekly Assignments - 12 % (Due: 9pm on Thursdays)
- Quiz - 83 %
 - Quiz 1 : 18 %
 - Quiz 2 : 20 %
 - Quiz 3 : 20 %
 - Quiz 4 : 25 %

Grader : Helen Lin (hlin65@dons.usfca.edu)



Weekly Assignment

All assignment for this course will be “Group Assignment” and you will have a group for HW_1_2 and HW_3_4.

- I will assign up to 3 ppl for group and you will submit one assignment per group.
(Not every member needs to submit)
 - You can consider it as your study group.
 - For building networks, I will randomly assign groups.
- If you do not work as a group, please let me know today.
 - However, I found that working individually didn't help my student who has a computer science degree - especially for building networks in the program.
- I am going to ask participation rate and your grade will be weighed based on it.
 - Tip : Make sure to do your homework as much as possible prior to your group meeting.

Weekly Assignment

It will be created on Canvas - Do not create your own group (it is not visible to me)

≡ [MSDS-501](#) > [People](#) > [Groups](#)

Summer 2022

Home

Syllabus

Modules

Assignments

Quizzes

Piazza

People

Grades

Echo360 AID

Everyone

HW_1&2_Group

+ Group Set

+ Import

+ Group

:

UNASSIGNED STUDENTS (0)

Search users

There are currently no students in this group. Add a student to get started.

GROUPS (24)

▶ [HW 1&2 Group 1](#)

4 students

:

▶ [HW 1&2 Group 2](#)

4 students

:

About Computation for DS (MSDS501)

This class is a standard, graded course with letter grades A - F.

- I consider an A grade to be above and beyond what most students have achieved.
- A B grade is an average grade for a student or what you could call "competence" in a business setting.
- A C grade means that you either did not or could not put forth the effort to achieve competence.
- Below C (F) implies you did very little work or had great difficulty with the class compared to other students.

The expected final score for this course is 85 ± 3 (and close to normal distributions).

The following grades will be given if the class grade distribution falls within the expectation.

However, if the grade distribution does not meet the aforementioned criteria, grades will be curved.

Score	Letter Grades
90 - 100	A+, A and A-
80 - 90	B+, B and B-
70 - 80	C+, C and C-
Below 70	F

About Computation for DS (MSDS501)

Please no plagiarism! – Zero tolerant.

```
(64%)  
19-24  
8-16  
24-27  
27-31  
  
data = sc.textFile(input_file1)\n    .map(lambda x: x.split(","))\n    .filter(lambda x: len(x) == 5)  
  
sensor_type = sc.textFile(input_file2)\n    .map(lambda x: x.split(","))\n    .map(lambda x: (int(x[0]), x[1]))  
  
sensor_readings_types = data.groupBy(lambda x: int(x[1]))  
  
sensor_readings = data.map(lambda x: (x[1]+", "+x[0],\n                                [float(x[2]), float(x[3]), float(x[4])]))\\  
  
    .mapValues(lambda x: (1, x))\n    .reduceByKey(lambda x, y: (x[0]+y[0],\n                                [x[1][0]+y[1][0],\n                                 x[1][1]+y[1][1],\n                                 x[1][2]+y[1][2]]))\\  
    .mapValues(lambda x: (round(x[1][0]/x[0], 4),\n                                round(x[1][1]/x[0], 4),\n                                round(x[1][2]/x[0], 4)))\\  
    .map(lambda x: (int(x[0].split(",")[0]),\n                                [float(x[0].split(",")[1]), x[1]]))  
  
f = open(output_file, 'w')\nfor item in sensor_readings_types.leftOuterJoin(sensor_type)\\  
  
    .sortByKey().collect():  
    f.write(str(item[0])+" : "+str(item[1][1])+"\n")  
    for prs in sensor_readings.groupByKey().collect():  
        if item[0] == prs[0]:  
  
data = sc.textFile(input_file1)\n    .map(lambda x : x.split(","))\n    .filter(lambda x : len(x) == 5)  
  
sensor_type = sc.textFile(input_file2)\n    .map(lambda x : x.split(","))\n    .map(lambda x : (int(x[0]), x[1]))  
  
sensor_type_in_readings = data.keyBy(lambda x : int(x[1]))\n    .groupByKey()  
  
#Transform data : timestamp, sensortype, x axis, y axis, z axis ==> sensortype:timestamp,  
  
sensor_readings = data.map(lambda x : ((x[1]+ ":" + x[0]), [float(x[2]), float(x[3]), float(x[4])]))  
  
#Calculate mean if there are multiple data with the same timestamp for each sensor.  
  
preprocessed_sensor_readings = sensor_readings.mapValues(lambda x: (1,x))\n    .reduceByKey(lambda x, y: (x[0]+y[0],[x[1]+y[1],\n                                x[2]+y[2],\n                                x[3]+y[3]]))\\  
    .mapValues(lambda x : [round(x[1][0]/x[0], 4),\n                                round(x[1][1]/x[0], 4),\n                                round(x[1][2]/x[0], 4)])  
  
#Print sensor information and first and last n_element\nf = open(output_file,"w")\nfor type in sensor_type_in_readings.leftOuterJoin(sensor_type).sortByKey().collect():\n    #print readings\n    for preprocessed_reading in preprocessed_sensor_readings.groupByKey().collect():  
  
        if(type[0] == preprocessed_reading[0]):\n            f.write(str(type[0]) + " : " + str(type[1][1]) + "\n")\n            for i in sorted(list(preprocessed_reading[1]))[:n_element]:\n                f.write(str(i) + "\n")
```

About Computation for DS (MSDS501)

AI Policy

- Assignments
 - You are not allowed to use Generative AI (e.g., ChatGPT, Copilot, etc.) to complete your assignments.
 - However, you may use GenAI to help you understand error messages or debug issues when you're stuck.
 - If you receive help from GenAI, you must include proper attribution as a comment in your code (e.g., `# Helped by ChatGPT to understand error message`).
- Quiz
 - No external tools or materials are allowed during quizzes.
 - This includes, but is not limited to: IDEs, GenAI tools, course notes, textbooks, websites, or any other outside resources.

Student Engagement

Example Data : https://github.com/dianewoodbridge/msds501_computation_2025

Piazza : <https://piazza.com/usfca/summer2025/msds501>

Poll : <https://pollev.com/msds>

Canvas

- Under each module (week), there are learning outcome, slides, homework, example tests, tests, etc.
- Lectures will be recorded and uploaded on Panopto.

Student Engagement

Canvas

• Panopto

Spring 2023

Search in folder "2023-MSDS697"...

+ Create

Powered by Panopto

Home

Syllabus

Piazza

Modules

Echo360 ALP

Assignments

Quizzes

Grades

People

Honorlock

Simple Syllabus

Panopto Video

Announcements

Rubrics

Pages

Outcomes

Discussions

Files

BigBlueButton

Collaborations

2023-MSDS697

Sort by: Name Duration Date ▾

+ Add folder

MSDS697-Task3



12:49

MSDS697-Sharding and Replication



22:21

Student Engagement

Canvas

- Piazza - For discussing lectures and assignments. (Preferable)

The screenshot shows the Piazza platform interface for a class named 'MSDS-691-01'. The top navigation bar includes links for 'LIVE Q&A', 'Drafts', and various course sections like 'lecture_-_week1' through 'lecture_-_week6'. A search bar and a 'New Post' button are also present. The main content area displays the 'Class at a Glance' summary, which indicates 'no unread posts', 'no unanswered questions', and 'no unresolved followups'. It shows 6 total posts, 6 total contributions, 0 instructors' responses, 0 students' responses, and n/a avg. response time. Below this, a 'Student Enrollment' section shows 103 enrolled students. A 'Share Your Class' section provides a demo link: https://piazza.com/demo_login?nid=kfzsup0ak1p6rn&auth=deb6dd8. A product update message at the bottom right mentions 'Product Updates: October 2, 2020'.

MSDS-691-01 > Relational Databases - 01 (Fall 2020)

Fall 2020

Home

Announcements

Syllabus

Zoom

Modules

Piazza

Quizzes

Honorlock

Grades

People

Collaborations

Rubrics

Discussions

Files

Pages

Outcomes

Conferences

Assignments

Settings

PIAZZA

MSDS-691 2

Setup Q & A Resources Statistics Manage Class

Diane Woodbridge

LIVE Q&A Drafts lecture_-_week1 hw1 lecture_-_week2 hw2 lecture_-_week3 hw3 lecture_-_week4 hw4 lecture_-_week5 hw5 lecture_-_week6 hw6

Unread Updated Unresolved Following

New Post Search or add a post...

Filtering on: lecture_-_week1

no results found

Class at a Glance Updated 14 seconds ago. Reload Go to Live Q&A

no unread posts 6 total posts

no unanswered questions 6 total contributions

no unresolved followups 0 instructors' responses

n/a avg. response time 0 students' responses

Student Enrollment ..out of 90 (estimated) Edit

103 enrolled

Download us in the app store: [App Store](#) [Google play](#)

Share Your Class

Professors appreciate Piazza best when they see how it is being used.

Allow colleagues to view your class through a demo link - a restricted, read only version of your class where all students' names are anonymized and all student information hidden.

https://piazza.com/demo_login?nid=kfzsup0ak1p6rn&auth=deb6dd8

Opening this link in the same browser will log you out as dwoodbridge@usfca.edu

Product Updates: October 2, 2020

We've created this space in your home screen to inform you of product updates that our team is working on.

- [Released] Uploading groups as a CSV file to easily assign students in a section to groups
- [Released] Ability to shut down a class entirely by hiding all posts in class
- [Released] Option for instructors to only allow private posts in their class
- [Released] Ability for professors to disable the Resolved/Unresolved toggle for their class

After this class...

I found that many of the students become comfortable to solve Easy - Medium level Leetcode questions.

The screenshot shows the LeetCode homepage with a search bar containing "Google". Below the search bar, there are navigation links: Explore, Problems, Interview, Contest, Discuss, and Store. A "Notice" box informs users about improved company tag calculations and weekly updates on Saturday. It also mentions filtering by time periods. A checkbox for "Show problem tags" is checked. A dropdown menu "Select time period: 6 months" is open. A table lists several problems with columns for ID, Title, Acceptance rate, Difficulty, and Frequency. The problems listed are: Two Sum, Add Two Numbers, Median of Two Sorted Arrays, Longest Palindromic Substring, Reverse Integer, Container With Most Water, and Roman to Integer.

#	Title	Acceptance	Difficulty	Frequency
✓ 1	Two Sum	50.0%	Easy	Medium
✓ 2	Add Two Numbers	40.6%	Medium	Medium
4	Median of Two Sorted Arrays	36.5%	Hard	Medium
✓ 5	Longest Palindromic Substring	32.4%	Medium	Medium
✓ 7	Reverse Integer	27.5%	Medium	Medium
11	Container With Most Water	54.0%	Medium	Medium
✓ 13	Roman to Integer	58.6%	Easy	Medium

Any Questions?

Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

Content

Basic Computer Architecture

Files and Folders

Terminal, Shell Commands/Scripts

Homebrew/Anaconda/Python Installation

Why did you purchase your computer??

Computer Programming Languages and Tools

Laptops

You'll be using your laptop every day, either in class or at Practicum. Your laptop must be in perfect working order and have the necessary computing power and storage. If you have a broken or slow laptop, it will directly affect your success in the program.

We strongly recommend you buy a Mac laptop with the new M1 processor, rather than the Intel processor. That means either the MacBook Air or MacBook Pro 13" which will have very similar performance. Get 16G not 8G of RAM, but the minimum 256G SSD storage is sufficient (minimum). The MacBook Air M1 with those specs is \$1,199 without the educational discount (which is usually small). Student discounts on Apple products can be accessed at

<https://www.apple.com/us-hed/shop?aid=AIC-AOS-Edu-EdMktng-HP> .

If you would like a bigger screen, you can look at the MacBook Pro 16" but you will pay a premium (\$2,399). It has an Intel Core i7 processor that boosts up to 4.5 GHz and 16G RAM at that price. (December 23, 2020)

Content

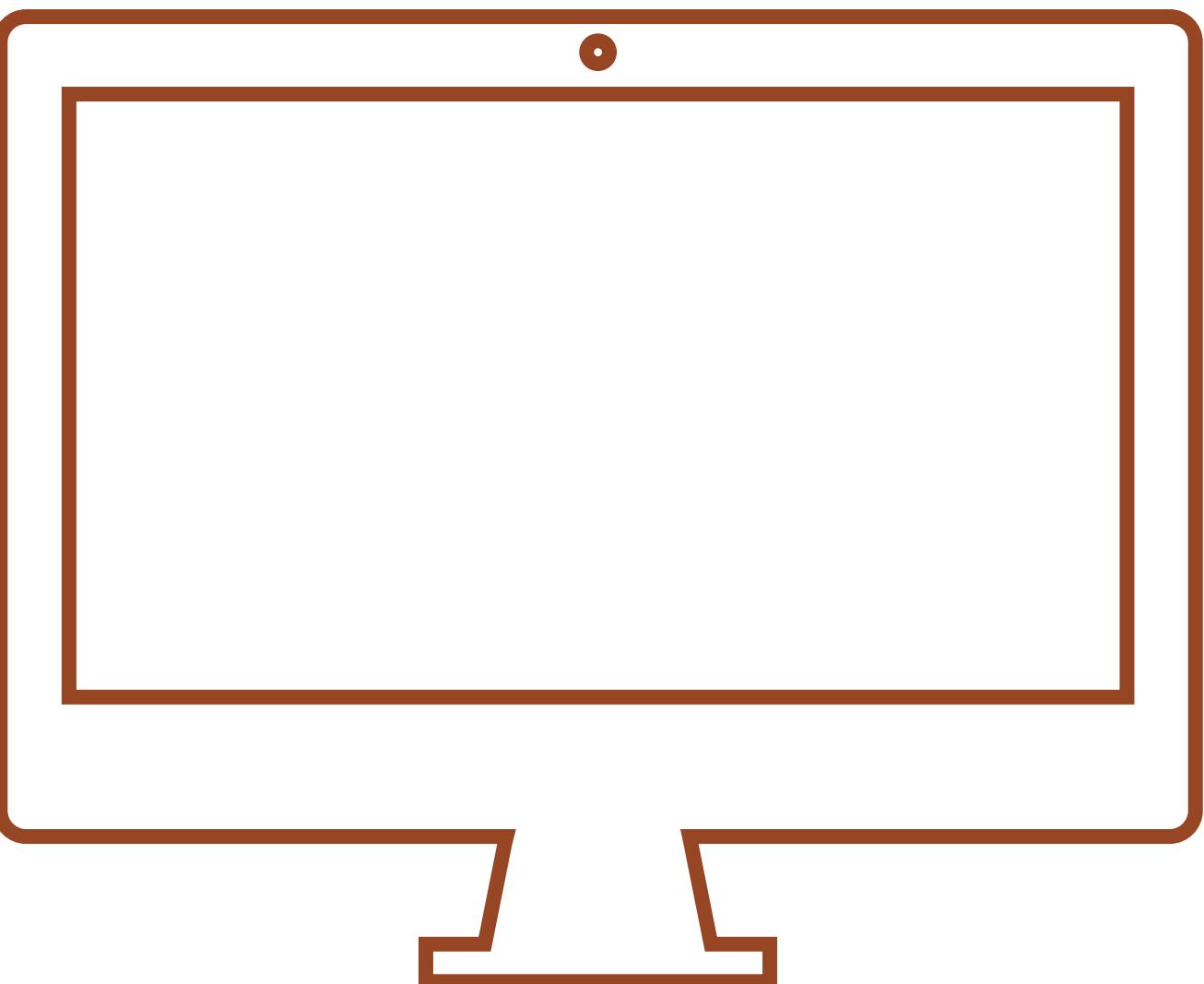
Basic Computer Architecture

Files and Folders

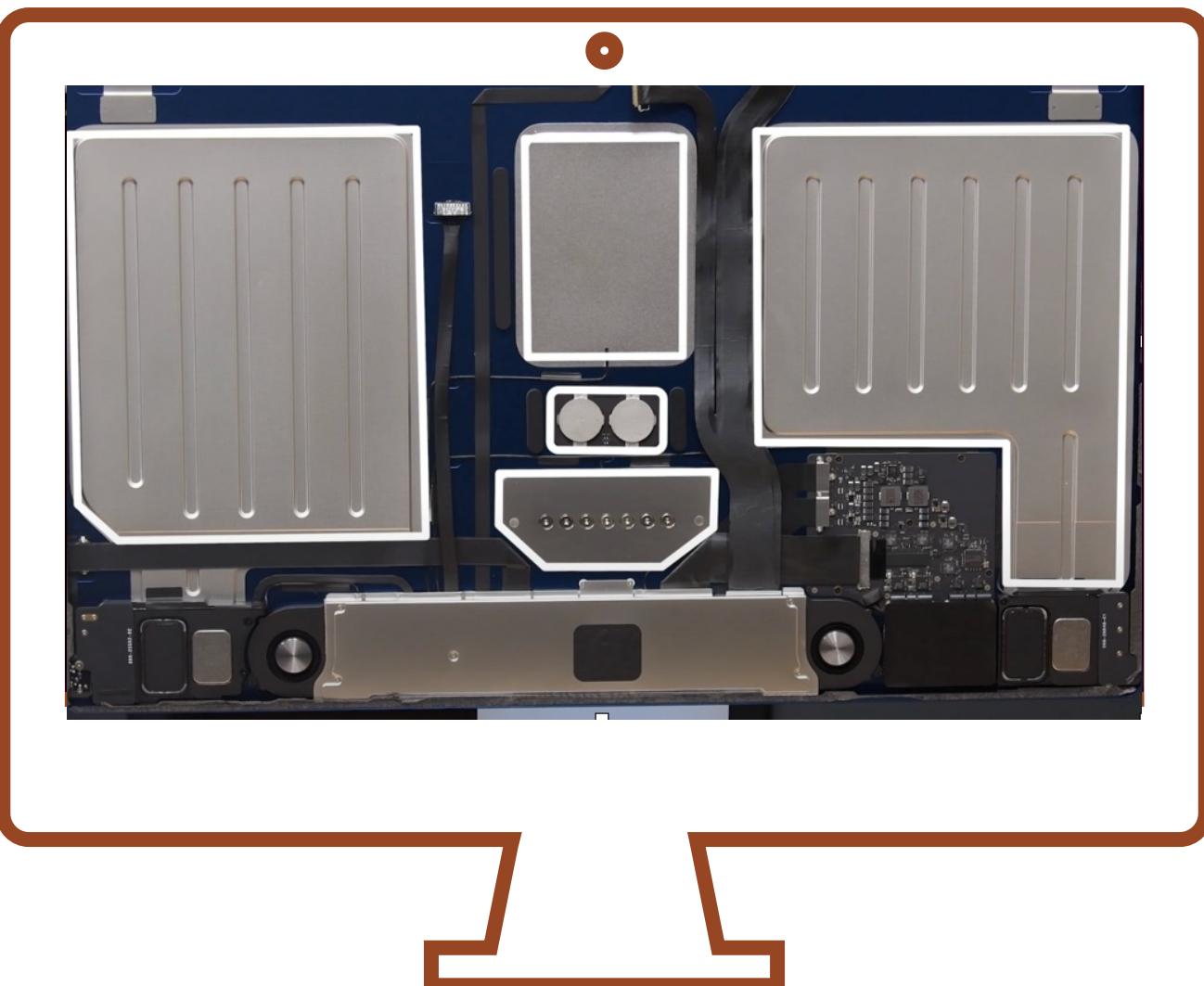
Terminal, Shell Commands/Scripts

Homebrew/Anaconda/Python Installation

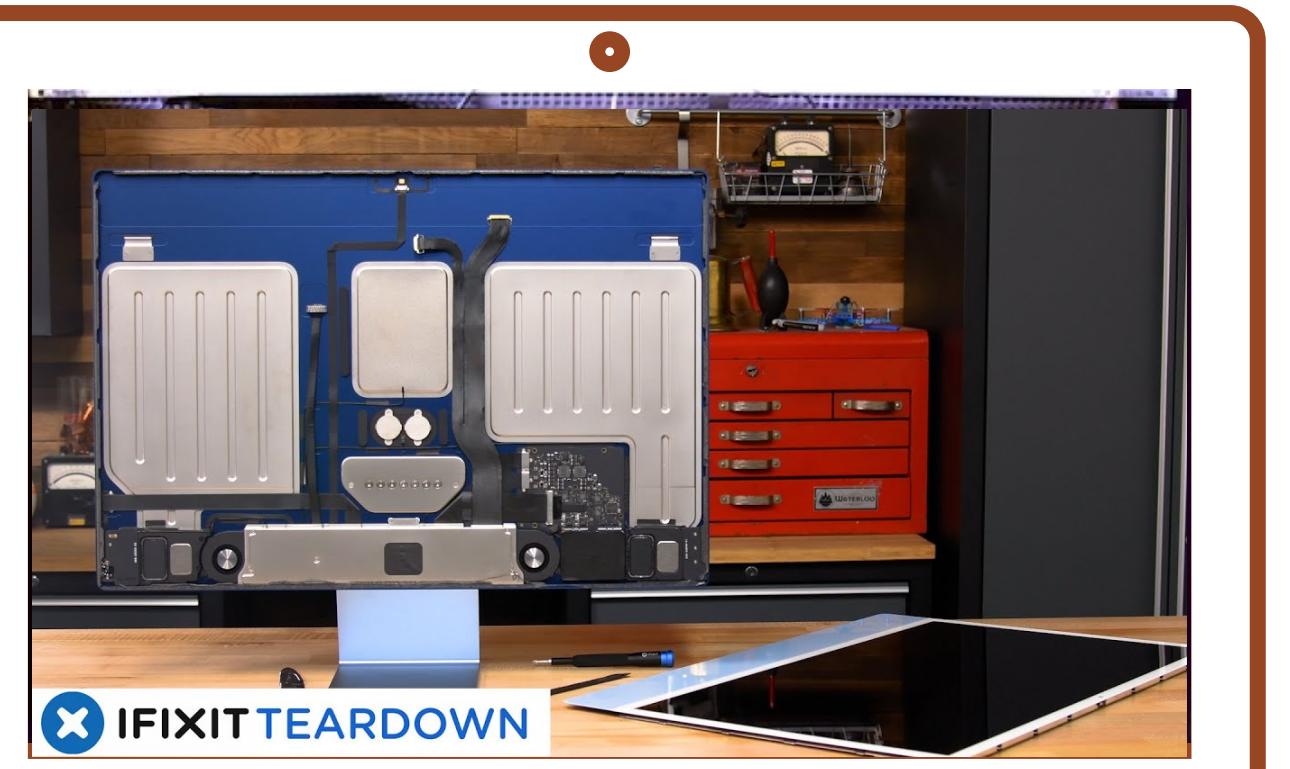
Your Computer



Your Computer



Your Computer



[youtube.com/watch?v=-WtGjkgIXGM&t=1s](https://www.youtube.com/watch?v=-WtGjkgIXGM&t=1s)

Your Computer



Your Computer

CPU

- Core
- Thread



Memory (RAM)



Nonvolatile Storage (Disk)



Your Computer

Central Processing Unit (CPU)



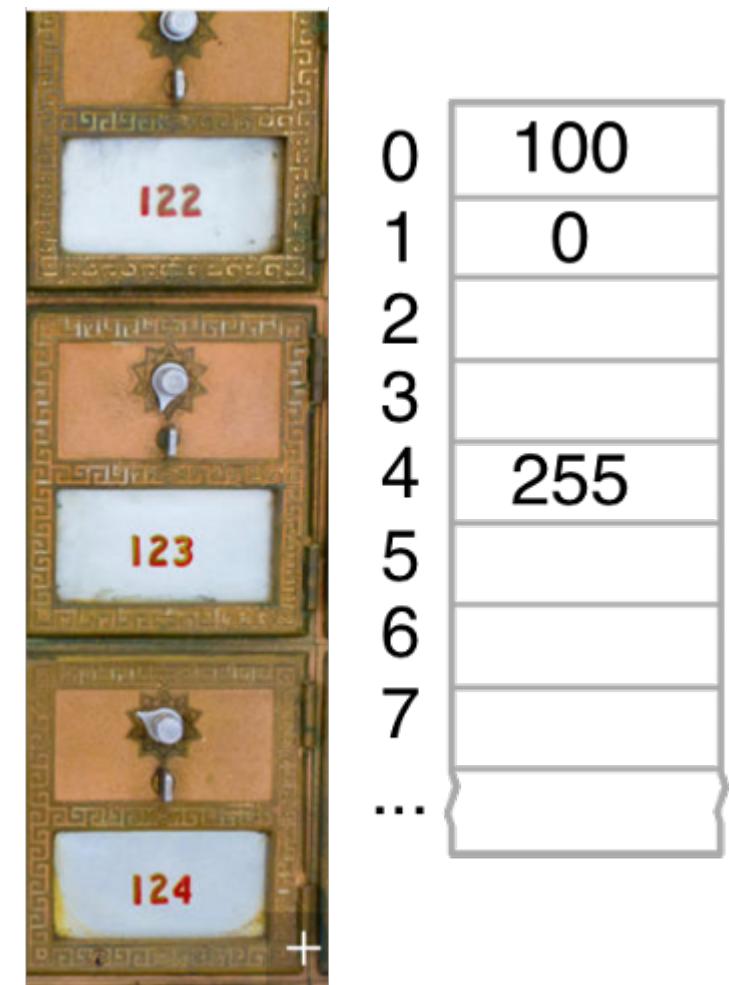
- Executes operations and computations
- 5 Key Task
 1. **Load**: Fetch small chunks of data from memory into the CPU.
 2. **Compute**: Perform arithmetic or logical operations on data.
 3. **Store**: Write small chunks of data from the CPU back to memory.
 4. **Jump**: Move execution to a new location in the program (enables loops).
 5. **Conditional Jump**: Move to a new location only if a condition is true (enables if/else logic)
- Key CPU Terminologies
 - **Core**: Think of a core as a worker - it processes tasks. More cores mean more workers handling jobs in parallel.
 - **Thread**: Each core can handle multiple tasks at once using threads. You can think of threads as the hands of each worker - more hands allow a worker to juggle more tasks concurrently.

Your Computer



Memory (RAM)

- RAM (Random Access Memory) is where the CPU reads and writes data during program execution.
 - It allows fast access to data at specific memory locations.
- How Memory is Structured
 - Memory is divided into cells called bytes.
 - 1 byte = 8 bits, and can store a number from 0 to 255.
 - Larger units:
 - 1 KB = 2^{10} bytes = 1024 bytes
 - 1 MB = 1024 KB, and so on.

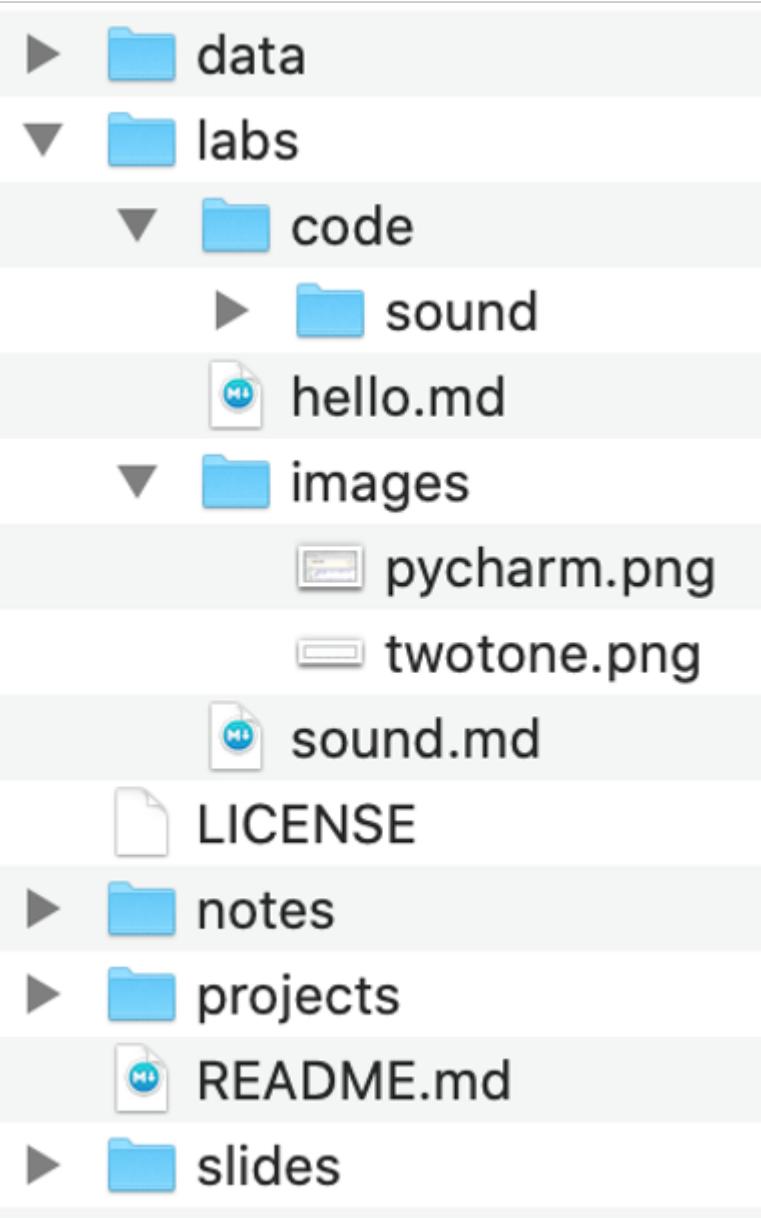
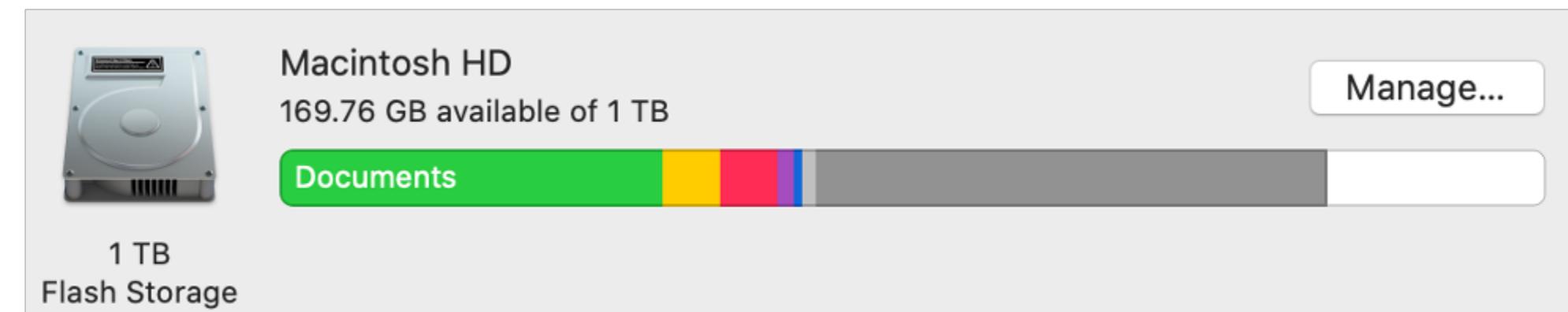


Your Computer

Nonvolatile Storage (Disk)



- Disk storage is nonvolatile, meaning data is retained even when the power is off.
 - Used for storing files and folders permanently
- Files and Directories
 - A file is a unit of data — like bts.mp3 or sales.csv.
 - A directory (also called a folder) groups:
 - Files, and/or
 - Other subdirectories.



Your Computer

Nonvolatile Storage (Disk)

- Disk Structure = Tree
 - The disk is organized like a tree:
 - The root is the base directory (e.g., / on Unix or Mac).
 - Folders branch out from the root.
 - Files are usually the leaves of the tree.

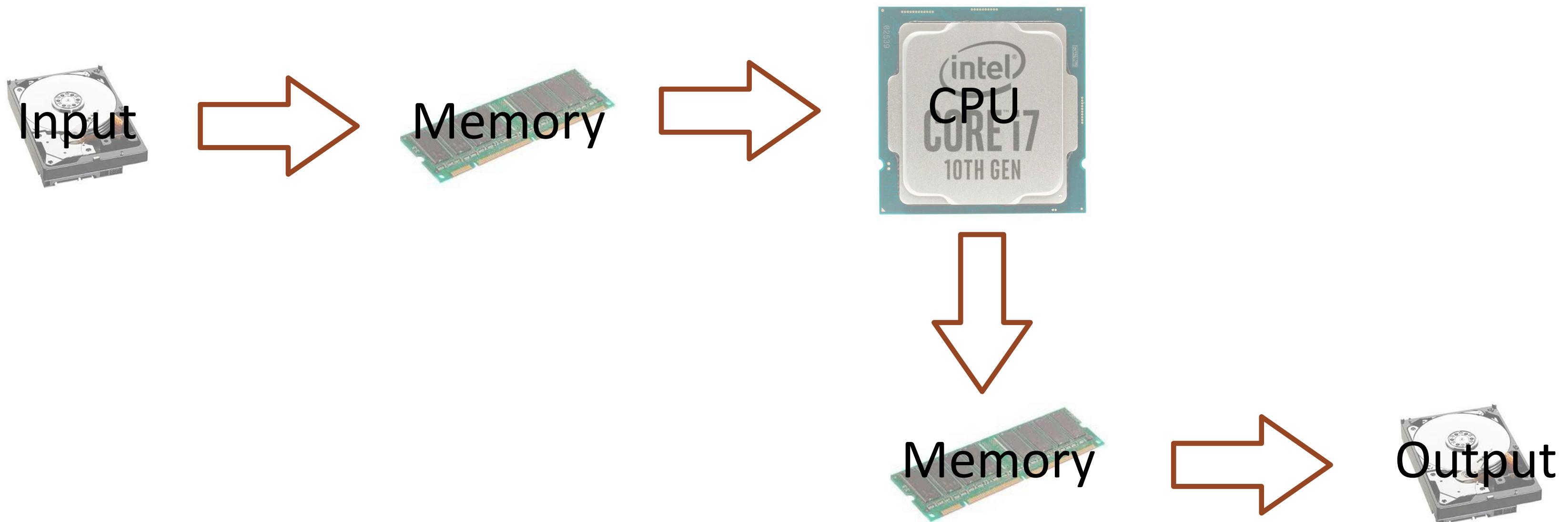


Your Computer

Memory vs Disk

	Memory	Disk
Speed	Faster	Slower
Size	Smaller	Larger
Persistence	Temporary (cleared on power-off)	Permanently

Read/process data from a file and write the results to a file.



Where are unopened files and folders primarily located?

Disc

Memory

Where are unopened files and folders primarily located?

Disc

0%

Memory

0%

Where are unopened files and folders primarily located?

Disc

0%

Memory

0%

Content

Basic Computer Architecture

Files and Folders

Terminal, Shell Commands/Scripts

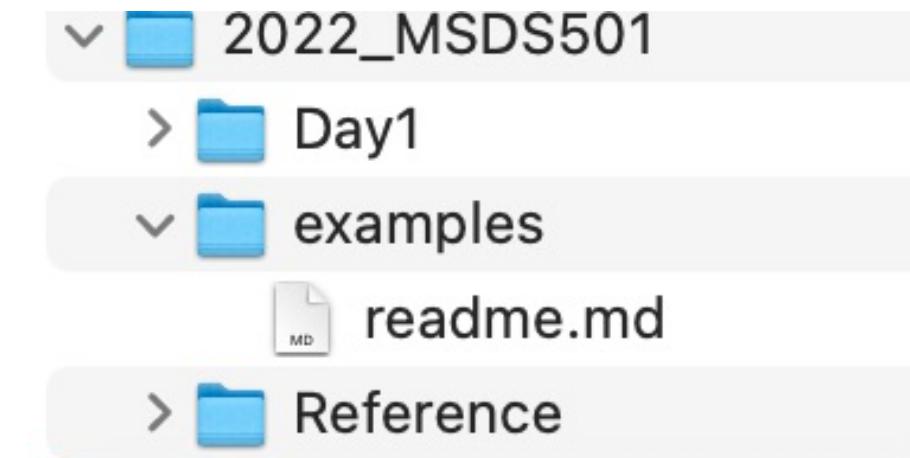
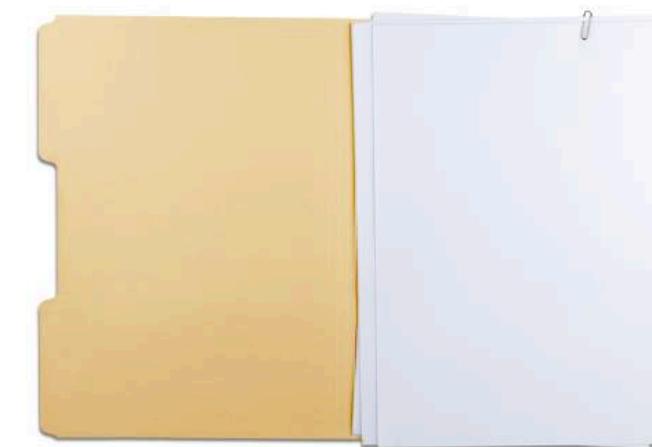
Homebrew/Anaconda/Python Installation

Files and Folders

When a program runs, it has a Current Working Directory (CWD)

- the folder it's “currently in.”

- File paths can be 1) Relative paths, or 2) absolute paths.



Files and Folders

Absolute Path and Relative Path

- **Absolute Path** : File or folder location (address) from the highest location (root directory, /)
 - Ex. /Users/dwoodbridge/Class/2025_MSDS501/examples
- **Relative Path** : File or folder location that is relative to a current directory
 - Current directory - where you are currently working
 - You can represent the current directory as .
 - Parent directory - A directory that contains the current directory
 - Ex. Parent directory of /Users/dwoodbridge/Class/2025_MSDS501/examples is /Users/dwoodbridge/Class/2025_MSDS501
 - You can represent parent directory as ../

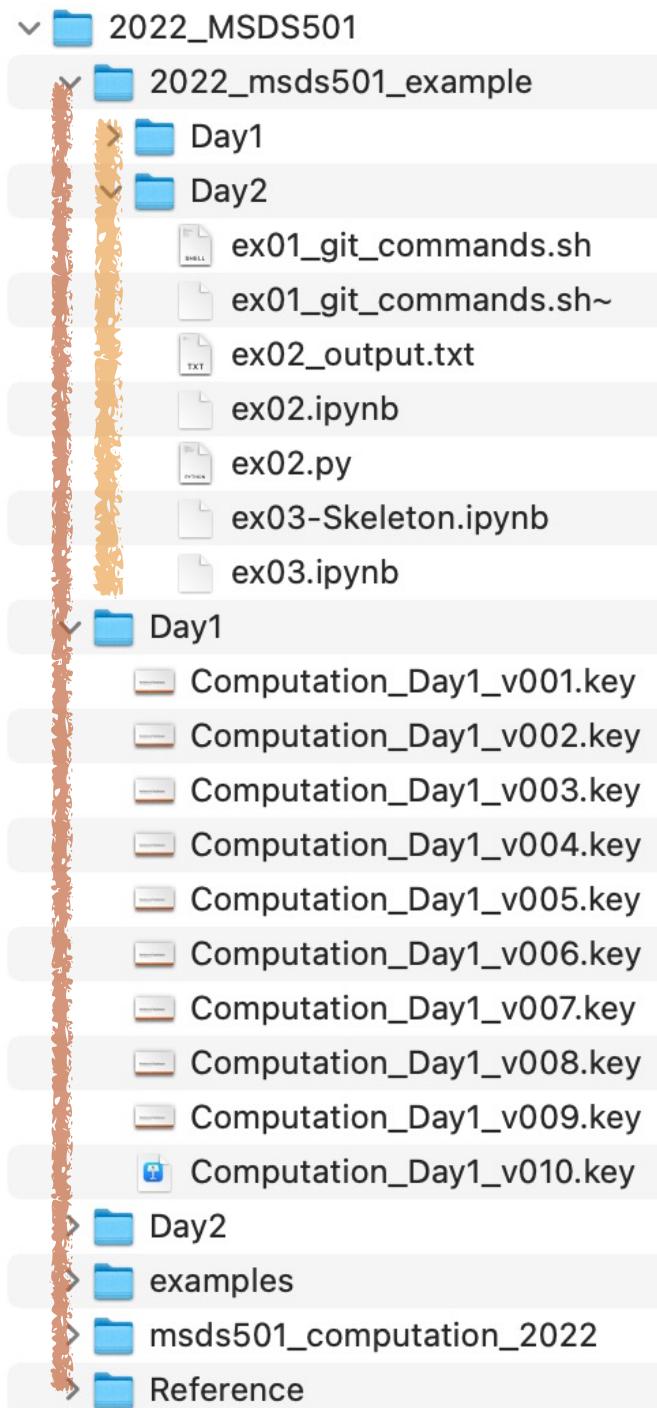
Files and Folders

Absolute Path and Relative Path



Example 1

For Day1/Computation_Day1_v001.key, what is the relative path for Day2/ex02.py?



Content

Basic Computer Architecture

Files and Folders

Terminal, Shell Commands/Scripts

Homebrew/Anaconda/Python Installation

Terminal



Terminal provides a command line-interface (CLI) to MacOS.

- The terminal is running a "shell", a command interpreter designed to run programs, manage and interact with computers.
- In my class, I use *commands* to distinguish terminal commands

A screenshot of the Mac OS Spotlight search interface. The search bar at the top contains the text "terminal". Below the search bar, the results are displayed under the heading "Communication". A single result, "Terminal.app", is shown with its icon and name. At the bottom of the screen, a terminal window is open with the following text displayed:

```
(DistributedComputing) ML-ITS-901885:2022_MSDS501 dwoodbridge$ pwd  
/Users/dwoodbridge/Class/2022_MSDS501  
(DistributedComputing) ML-ITS-901885:2022_MSDS501 dwoodbridge$ ls  
Day1 Reference  
(DistributedComputing) ML-ITS-901885:2022_MSDS501 dwoodbridge$
```

Terminal



Basic Terminal Commands

Type	Command	Explanation
Directory	<code>pwd</code>	Print current working directory
	<code>cd path</code>	Change current working directory
	<code>mkdir name</code>	Make a new directory
	<code>ls (-la)</code>	List files and subdirectories (with long format including hidden files)
	<code>rm -r name</code>	Remove/Delete the directory (-r means recursively). Carefully use it, as you cannot undo especially with -f option (force).
File	<code>cp from to</code>	Copy “from” to “to”
	<code>mv from to</code>	Rename/Move “from” to “to”
	<code>rm name</code>	Remove/Delete the file
	<code>cat file</code>	Print the content of the file.
	<code>touch file</code>	Create <i>file</i> without content.
	<code>diff file_1 file_2</code>	Return difference between <i>file_1</i> and <i>file_2</i>
Permission	<code>chmod option filename</code>	Change the read, write, and execute permissions option - 3 digit for user, group, and others. Each digit is a sum of 4, 2, 1, or 0. (4: read, 2: write, 1: execute, and 0 : no permission)
Process	<code>ps aux</code>	Print all process of all users
	<code>kill pid</code>	kills the processes with pid

Terminal



Basic Terminal Commands

- chmod
 - User class
 - User (u) — the owner of the file
 - Group (g) — a set of users who share the same group
 - Others (o) — everyone else (anyone who's not the owner or in the group)
 - Permission for each class
 - r - Read (4)
 - w - Write/Modify (2)
 - x - Execute/Run (1)
 - Change permissions for each class of users using symbols (+,-,= with r,w,x) or numbers
 - Ex. If file.txt is a read-only for the user, the followings will give read, write, and execute access to the user.
 - chmod u+wx file.txt
 - chmod u=rwx file.txt
 - chmod 700 file.txt

Terminal



Terminal Tips

- Tab does “auto-complete”
- If you are not specifying the directory, the commands work in your current working directory.
- If you get permission related errors, add sudo at the beginning of your commands.
- You can also pipeline a sequence of multiple commands by adding '| ' or ' |& '
- Parameter -f is for “forcing”, -r is for “recursively”.
Ex. If you need to delete all the files in a folder recursively, `$rm -r folder_name`

Example 2

- Create a folder called 2025_MSDS501_Day1 on the home (~) directory.
- In the current working directory, create *empty.txt*.
- Change your directory to 2025_MSDS501_Day1 and move *empty.txt* to here.
- Create an *empty_2.txt* with some contents.
- Compare *empty.txt* and *empty_2.txt*
- Change *empty.txt* to be readable/writeable and executable to everyone.
- Delete 2025_MSDS501_Day1 and all of its contents.

Example 2

- Create a folder called 2025_MSDS501_Day1 on the home (~) directory.
- In the current working directory, create *empty.txt*.
- Change your directory to 2025_MSDS501_Day1 and move *empty.txt* to here.
- Create an *empty_2.txt* with some contents.
- Compare *empty.txt* and *empty_2.txt*
- Change *empty.txt* to be readable/writeable and executable to everyone.
- Delete 2025_MSDS501_Day1 and all of its contents.

```
#Create a folder called 2025_MSDS501_Day1 on the home (~)
directory.
mkdir ~/2025_MSDS501_Day1
# In the current working directory, create empty.txt.
touch empty.txt
# Change your directory to 2025_MSDS501_Day1 and move empty.txt
# to here.
cd 2025_MSDS501_Day1
##### The following can be done by $mv ../empty.txt empty.txt
cp ../empty.txt .
rm ../empty.txt
#####
# Create an empty_2.txt with some contents.
vi empty_2.txt

# Compare empty.txt and empty_2.txt
diff empty.txt empty_2.txt

# Change empty.txt to be readable/writeable and executable to
everyone.
chmod 777 empty.txt
# Delete 2025_MSDS501_Day1 and all of its contents.
rm -rf ~/2025_MSDS501_Day1
```

Terminal



Terminal Commands



**Not familiar with shell commands and
GUI seems waaaaay easy at this point?**

Make no mouse/trackpad hours.
Within those hours, you are only allowed to
use keyboards and shell commands.



Shell Scripts

A script including shell commands for directory/ file manipulation, program execution, etc.

- File extension : .sh
- You can execute it by \$sh xxx.sh
 - OR if you add a Shebang line #!/bin/sh at the beginning of the file, you can execute it by \$./xxx.sh
- You can combine with other control sequences including for variable assignments, loop, conditional statements, etc. (Note : Those syntax are different from Python syntax)

Shell Scripts

Variable Assignment

- Use `=` (equal sign) without any space before/after. Ex. `a=13`
- To refer to the variable after assignment, you need to add `$` before the variable name.
Ex. `echo $a`

Loops

- Create a block of

```
for arg in list_of_items_separated_by_space
do
    command(s)...
done
```

Conditional Statement

- Create a block of

```
if [ condition ]
then
    command(s)
fi
```

Array Declaration (List)

```
ARRAY=(value1 value2 ... valueN)
```

Using an Array in for Loop

```
for value in ${ARRAY[@]}
```

Comparison

- Integer : `-gt`, `-gte`, `-lt`, `-lte`, `-eq`, `-ne`
- String : `==`, `!=`, `<`, `>`

Shell Script

Extra for Homework 1

- **\$1, \$2, ...** : 1st, 2nd, command line argument
 - Ex. sh hw1.sh `url` `folder`
- **\$#** : Number of arguments passed to a script or function
- **\$?** : Exit status of the last command
 - Returns 0 if the last command succeeded
 - Returns non-zero if failed
- **exit [N]**
 - N is an optional integer representing the exit status code.
 - If N is omitted, the exit status is that of the last executed command.
 - N being 0 (Zero): Conventionally indicates successful execution.

Example 3

For numbers in *nums* defined in ex03.sh, print if the number is greater than *threshold*.

Array Variables : https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_10_02.html

Comparison Operator : <https://tldp.org/LDP/abs/html/comparison-ops.html>

Example 3

For numbers in *nums* defined in *ex03.sh*, print if the number is greater than *threshold*.

```
nums=(10 11 12 13)
threshold=12

for num in ${nums[@]}
do
if [ $num -gt $threshold ]
then
echo $num
fi
done
```

Array Variables : https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_10_02.html

Comparison Operator : <https://tldp.org/LDP/abs/html/comparison-ops.html>

Example 3

For numbers in *nums* defined in `ex03.sh`, print if the number is greater than *threshold*.

Array Declaration (List)

`ARRAY=(value1 value2 ... valueN)`

Using an Array in for Loop

`for value in ${ARRAY[@]}`

Comparison

- Integer : `-gt`, `-gte`, `-lt`, `-lte`, `-eq`, `-ne`
- String : `==`, `!=`, `<`, `>`

Array Variables : https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_10_02.html

Comparison Operator : <https://tldp.org/LDP/abs/html/comparison-ops.html>

Environment Variables

An environment variable is a user-defined variable that stores information used by running processes on your computer.

- As a data scientist or developer, you'll often use environment variables to:
 - Avoid hardcoding sensitive information (e.g., passwords, API keys)
 - Configure cloud or container environments
 - Pass parameters to programs/scripts without changing code

Environment Variables

Setting Environment Variables

- To set an environment variable temporarily in your terminal session: `$export ENV_NAME=VALUE`
 - Make sure not to use any space before/after =
 - To verify it's set: `$echo $ENV_NAME`
- Recommendation : If you want the variable to be available every time you open a terminal, add to `~/.bash_profile.sh` or `~/.zshrc` (mac's default)
 - After updating it either open a new terminal or do `$source ~/.bash_profile` or `$source ~/.zshrc`

Content

Basic Computer Architecture

Files and Folders

Terminal, Shell Commands/Scripts

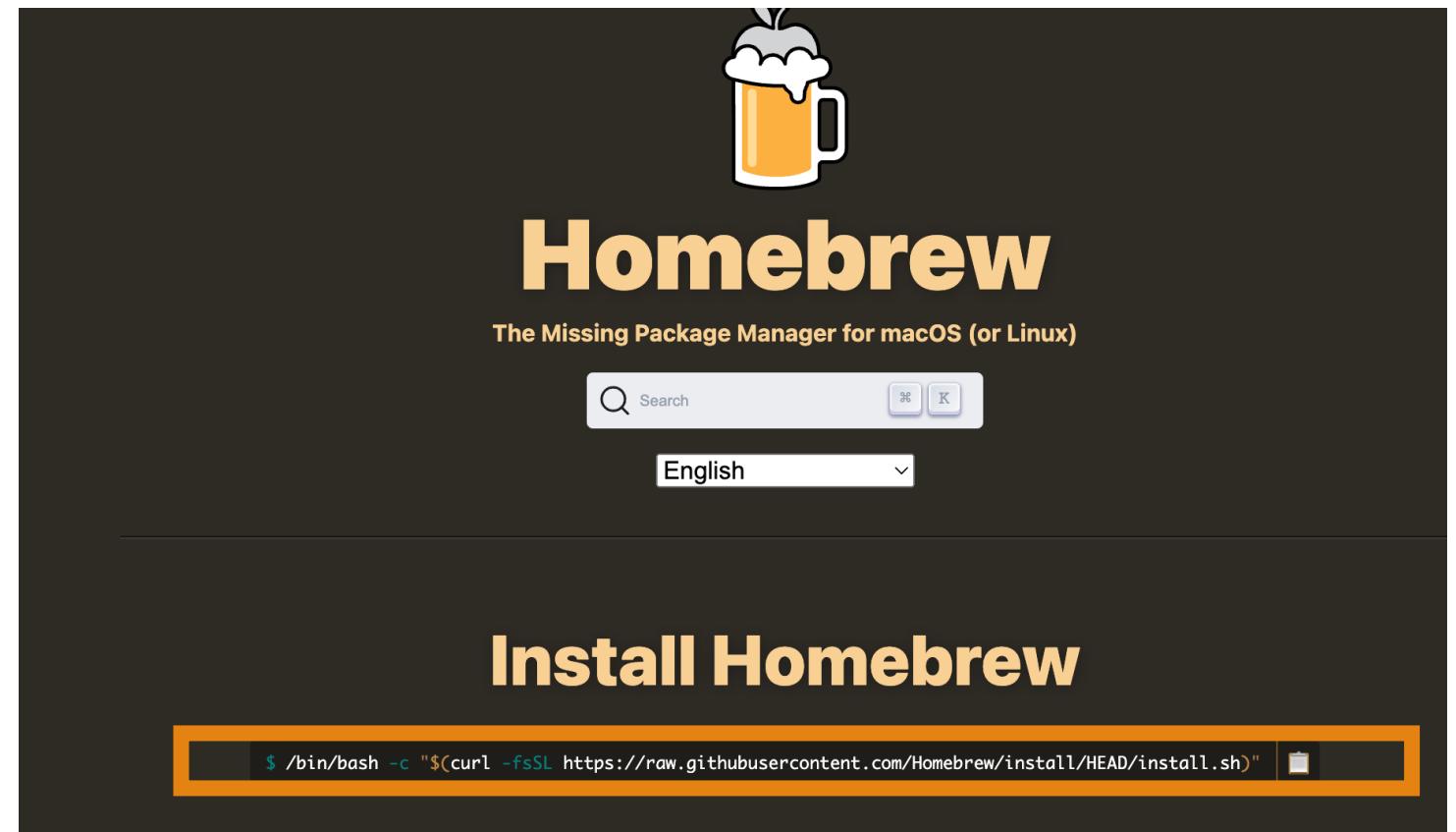
Homebrew/Anaconda/Python Installation

Let's Get Ready

Installation

- Homebrew
 - For installing missing packages for MacOS
 - Download and Install

```
$/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```



<https://brew.sh/>

Let's Get Ready

Installation

- Anaconda
 - Package management tool - It supports more than Python distributions
 - We will install packages using Anconda using
`$ conda install package_name`
 - Download and Install
 - `$ brew install --cask anaconda`

<https://formulae.brew.sh/cask/anaconda>

<https://docs.brew.sh/Formula-Cookbook#homebrew-terminology>

Let's Get Ready

Installation

- Python - Install version 3.13
 - Using conda, it will install other packages including numpy, scipy, pandas, jupyter. etc.
 - To install a certain version of python

```
$conda install python=3.x
```

```
dwoodbridge@ML-ITS-210588 anaconda3 % conda install python=3.13
Retrieving notices: done
Channels:
  - defaults
Platform: osx-arm64
Collecting package metadata (repodata.json): done
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: /opt/homebrew/anaconda3
```

```
added / updated specs:
```

Example 4

Check Python being installed, by type `python` on your terminal

- If it doesn't give the proper output (especially >>>), you might need to configure your path on the environment file.

```
[dwoodbridge@ML-ITS-210588 ~ % python
Python 3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 11:23:37) [Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

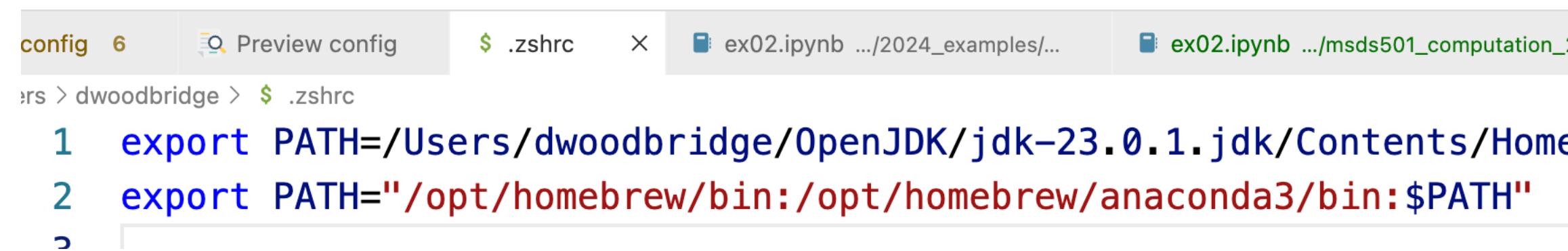
- To exit from the interactive mode, type in `exit`

```
[dwoodbridge@ML-ITS-210588 ~ % python
Python 3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 11:23:37) [Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> exit
dwoodbridge@ML-ITS-210588 ~ % █
```

Example 4

Check Python being installed, by type `python` on your terminal

- If it doesn't give the proper output (especially >>>), you might need to configure your path on the environment file.
 - To configure add the following lines in .zshrc (or corresponding file)
`export PATH="/opt/homebrew/bin:/opt/homebrew/anaconda3/bin:$PATH"`
 - Once saved make sure it is updated by either restarting a terminal or type \$ `source ~/.zshrc`



```
config 6 Preview config $ .zshrc x ex02.ipynb .../2024_examples/... ex02.ipynb .../msds501_computation_1  
ers > dwoodbridge > $ .zshrc  
1 export PATH=/Users/dwoodbridge/OpenJDK/jdk-23.0.1.jdk/Contents/Home  
2 export PATH="/opt/homebrew/bin:/opt/homebrew/anaconda3/bin:$PATH"  
3
```

If you had an issue to install...

First post on Piazza

- Many encounter similar issues.
- Sharing your problems and getting help from me or other students will help others.
- If it is something that I need to configure in person, I will ask you to meet.
 - If it is urgent, please email/slack me (dwoodbridge@usfca.edu)

Content

Basic Computer Architecture

Files and Folders

Terminal, Shell Commands/Scripts

Homebrew/Anaconda/Python Installation

Checklist

- Understand the scope and exceptions of the class**
- Understand how files and folders are stored in a computer**
- Using shell commands, be able to navigate and manipulate files, folders, and processes.**
- Be able to create a shell script to combine multiple commands and automate the process**
- Install Homebrew**
- Install Python**

Day 1 - Comments (What you liked/disliked so far? What should I do for you?)

References

- M1 iMac Teardown : youtube.com/watch?v=-WtGjkgIXGM&t=1s
- Advanced Bash-Scripting Guide : <https://tldp.org/LDP/abs/html/>
- Homebrew : <https://brew.sh/>
- Anaconda : <https://docs.anaconda.com/>
- Shell Commands : <https://mally.stanford.edu/~sr/computing/basic-unix.html>

Class Summary

Class Overview

**Python
Programming**

**Computing
Basic**

Object Oriented Programming

Python Packages and Libraries

Python Data Types and Statements

Version Control and Git

Terminal and Shell Commands

Basic Computer Architecture
Terminal and Shell Commands

Appendix

If `\\$ brew` or `\\$ conda`, returns command not found,

- Edit your `~/.zshrc` (or `~/.bash_profile`) to include the PATH variable.
 - `export PATH=/opt/homebrew/bin:/opt/homebrew/anaconda3/bin:$PATH`
 - Make sure there is no extra space in the above line!