

---

---

# CS106A Code in Place

— Section 6: Data Structures —

Week 6: 2<sup>nd</sup> June, 2023

---

---

# Today's plan



- Check-in
  - Concept review
    - Lists
    - Dictionaries
    - Files
  - Section Problem
    - Index game
    - List Practice
    - Heads Up
- 
- Q&A

# Final Catch-Up

- Last section 😞
- But there is still an extra week and the forum will be still be active!
- Question time (answer question then name someone else):
  - What was your favourite problem, section, or memory from your time here?

# Concept Review

- Data structures
- Lists
- Dictionaries

# Data Structures

## List



- An ordered collection of values.
- Usually used to store a “list” of alike data.
- Example: a grocery list with a whole bunch of foods! The only difference is you might not number your grocery list.  
:)

## Dictionary



- An unordered collection of key/value pairs.
- Usually used to store relational data.
- One example is a phone book or a contact list in a phone. Each name has a number associated with it!

# Lists: Group of Boxes

- Creating an empty list: `list_name = [ ]`
- Creating a list with values: `names = ['Julie', 'Mehran', 'Simba', 'Ayesha']`

Instead of a variable name and value, you have a list name, an index or position, and a value

Value:	'Julie'	'Mehran'	'Simba'	'Ayesha'
Index:	0	1	2	3

names

# Lists: Going through all elements in a list

value	'red'	'green'	'blue'
index	0	1	2

- 2 methods:
  - Manually print by accessing element X
  - For-loops

```
colors_list=['red', 'green', 'blue']

# manually print all colors
print(colors_list[0])
print(colors_list[1])
print(colors_list[2])

# dynamically print all colors
for i in range(len(colors_list)):
    print(colors_list[i])
```

# List: Operations

```
colors_list=['red', 'green', 'blue']
```

value	'red'	'green'	'blue'
index	0	1	2

## Accessing

```
print(colors_list[3])
```

# IndexError: list index out of range

## Appending

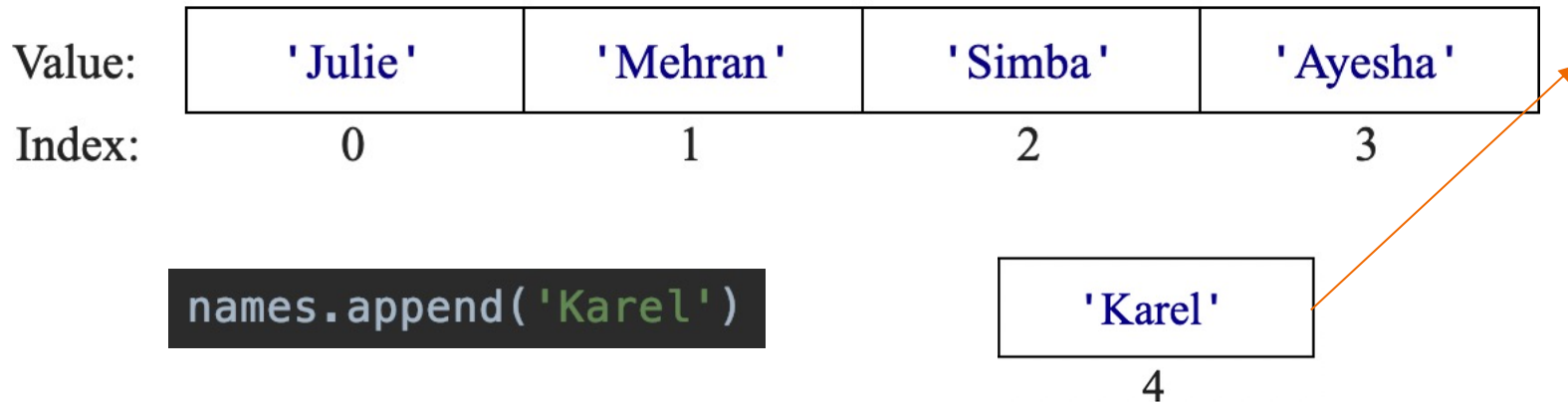
```
colors_list.append('alpha')  
print(colors_list[3])
```

# alpha

value	'red'	'green'	'blue'	'alpha'
index	0	1	2	3



# Lists: Appending example



Value:	'Julie'	'Mehran'	'Simba'	'Ayesha'	'Karel'
Index:	0	1	2	3	4

# Lists: Operations (cont'd)

```
colors_list=['red', 'green', 'blue', 'alpha']
```

value	'red'	'green'	'blue'	'alpha'
index	0	1	2	3

## Changing values

```
colors_list[1]='red'
```

value	'red'	'red'	'blue'	'alpha'
index	0	1	2	3

## Summary

```
for i in range(len(colors_list)):
    print("color ", str(i + 1), "is ", colors_list[i])
    # color 1 is red
    # color 2 is ...
    # ...
```

# Lists: Documentation & Facts

## Some useful documentation:

- New empty list: `list_name = [ ]`
- New non-empty list: `list_name = [value_1, value_2, ..., value_n]`
- Append to end: `list_name.append(elem)`
- Insert specific location:  
`list_name.insert(index, elem)`
- Remove first instance of an item:  
`list_name.remove(elem)`
- Get specific and remove specific item:  
`popped_item = list_name.pop(index)`
- Get specific item: `accessed_item = list_name[index]`
- Get length of list: `len(list_name)`

## List Facts:

- Lists are ordered (accessible via index, which start at 0 in Python)
- Lists are mutable (changing list in helper function will change the original list)
- Lists can contain data of different types, but same type usually preferred

# Dictionaries: Intro

`capitals = {'France': 'Paris', 'Germany': 'Berlin'}`

key      value      item

# Dictionaries: Documentation & Facts

## Some useful documentation:

- New empty dict: `dict_name = { }`
- New non-empty dict: `dict_name = {key: value, key2: value_2, ...}`
- Add new item to dict: `dict_name[key] = value`
- Deleting key/value pair: `del dict_name[key]`
- Key a value and deleting key/value pair: `value = dict_name.pop(key, None)`
- Unsafe key access and value return: `value = dict_name[key]`
- Safe key access and value return: `value = dict_name.get(key)`

## Dict Facts:

- Mutable, list lists
- Keys must be unique
- Keys must be hashable (usually strings)
- Before accessing/deleting data, make sure key is in dictionary (especially if key is user input)

# Files

- We have a file called *"covid-survival-list.txt"*
- We want to read the file and add each line to a list:

*covid-survival-list.txt*

```
food  
water  
toilet paper
```



```
FILENAME="covid19-survival-list.txt"  
items_list=[]  
  
with open(FILENAME) as file:  
    for line in file:  
        items_list.append(line.strip())  
print(items_list)  
  
# ['food', 'water', 'toilet paper']
```



## Section Problem



Index Game

List Practice

Heads Up

# Heads Up

- Heads Up is a game where one person takes the role of a “guesser”.
- A word will be generated at random, and everyone who isn't the guesser is providing hints as to what the word is without saying the actual word.
- The guessers job is to guess what the word is as fast as possible! An example of a round of this game is shown on the right.





# Heads Up

- Example Input (list of words):

Apple  
Banana  
Pear  
Blueberry  
Strawberry  
Orange

- Example Output:

Word to guess: Orange  
Hit enter to get next word.

Word to guess: Pear  
Hit enter to get next word.

...

# Heads Up

## Problem details

- Words: We provide a list of words for you to use in a file named "cswords.txt".
- Files: We are given a function that will read words from a file for us called `get_words_from_file(...)`.
- Game Loop: We want to repeat the game infinitely (or until we hit the stop button at the top of the IDE)



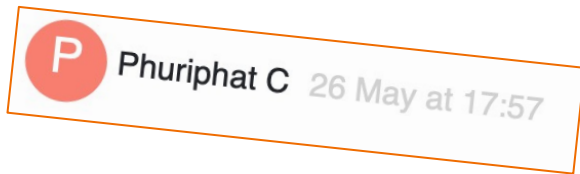
# Q&A



Answering any questions you might have about the course, Python, coding in general, working in industry, future projects, next steps etc.

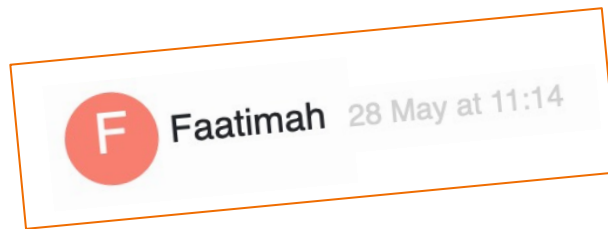
- First, the questions that were asked on the forum, then any questions you might have now

# Q&A



- Can I use import canvas outside of Code in Place ?
  - Canvas is created for Code In Place, but it's based on another package that you can install and use in a very similar way: <https://cs106a.stanford.edu/graphics>
- I learned that after this we're going to have to use something like PyCharm if we are to continue coding in python.  
I downloaded pycharm, but was kinda confused about the files and library things. It'd be great if you can elaborate on that next week.
  - Use text editors (Atom, Sublime Text, Notepad++) or a fully-fledged IDE (PyCharm, Visual Studio Code)
  - Difference is you run the code outside the text editor (a terminal), but can run/test everything inside IDE like PyCharm

# Q&A



- What IDE do you use?
  - Different IDEs for different languages. Will stick to Python only:
    - PyCharm for large Python projects (apps, websites, etc.)
    - JupyterLab for data science projects (building models)
    - Visual Studio Code for smaller projects where there are many individual files
- Any sites/newsletters/communities that you'd recommend for learning python and for keeping updated with new developments in this field?
  - Follow the stars of AI on LinkedIn and Twitter (most influential companies and people)
  - Subscribe to newsletters (Medium Daily Digest, TLDR, TLDR AI, Tech Brew, The Batch @DeepLearning.AI, PyCoder's Weekly)
  - Follow and check out code repositories on GitHub

\* the subscript refers to an offset from the starting position of an array, so the first element has an offset of zero.

# Q&A



Faatimah 28 May at 11:14

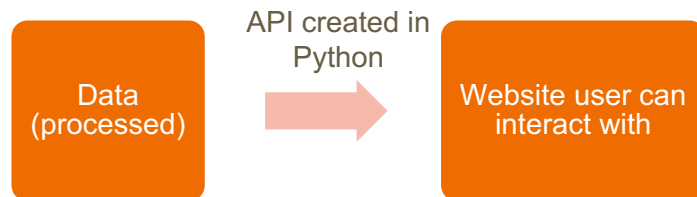
- Are there any plug-ins/apps that you use often for general productivity and for coding?
  - To stay off your phone: Forest App and Opal
  - To be productive and measure your productivity: Pomodoro Technique
  - To keep track of tasks and key info: Notion and Trello (any Kanban board will do)
- What's the reason for python indexing starting at 0 instead of 1 ?
  - Comes down to a **design decision** by the language creators and the main applications of that language.
  - Many languages are 0-indexed (C, C++, Go, Java, JavaScript, PHP, Python, Ruby, Swift, et) and many are 1-indexed (Lua, MATLAB, R, Fortran, Julia, Sass, etc.)
  - One reason is **C** is 0-indexed and became very widespread in 1980s. Because many languages are derived from C, they adopted 0-index as well. Other languages chose 1 as it's more natural to humans.
  - C is 0-indexed as it's less math-heavy and relies heavily on pointer arithmetic\*, which is more optimal using 0-index.

# Q&A



Faatimah 28 May at 11:14

- When we share our programs that we've created, what's going on behind the scenes? Basically, how does the functions become the screen that users can see and interact with?
  - Code that creates a visual output that users can interact with usually use libraries (such as graphics which gives us Canvas).
  - There are many libraries that create different types of outputs such as (Python only):
    - To create an interface that the user can interact with: Tkinter or Pygame
    - To plot data: matplotlib or seaborn
    - To create websites: redirect the data from Python to a front-end



# Q&A



Faatimah 1 June at 07:30

- In machine learning hackathons, a leader board is used to judge the quality of the model performance. In 'real life' how is a model's performance assessed?
  - Hackathons and competitions: performance is based on accuracy (# of correct predictions) and sometimes speed.
  - Real-life: also performance/speed + other metrics:
    - Ease of implementing with current systems and speed in the live production environment
    - Stability and robustness: will it continue performing well over time if the data changes?
- In your experience with data science, is it better to be a generalist ( i.e. knowing a bit of everything) or a specialist (being really good in a single domain)?
  - A good data scientist will know all types of models and when to use which. Will also know best practices
  - If you plan on working in 1 industry (e.g., finance or marketing), then you will have the upper-hand as in you will know common problems that are faced in this industry and will know which tools work better. But a good data scientist can adapt to any industry / problem.



# Q&A



Leslie L 1 June at 16:27

- Do you have some advice on the roadmap that I could follow to learn data analysis and machine learning? Not sure if that is possible without a solid background in computer science...
  - There are many routes to take as a data scientist has many skills (but not all) such as:
    - Computer science (manipulating databases, beginner coding in Python and SQL minimum)
    - Machine learning (understanding different ML models and best practices techniques)
    - Storytelling via the numbers you output (Python outputs, Excel, presentations, dealing with stakeholders who are not tech-oriented)
    - Optional: more advanced coding to deploy models into production
  - Some of these routes include taking online courses like DataCamp, or doing a 1-year Master's in data science. The objective is to get the skills to do the job of a data scientist in the domain you want.

# Q&A

Any other questions?

# Thank you for everything

- Connect with me on LinkedIn:
  - <https://www.linkedin.com/in/adamjaamour/>
- Drop me an email:
  - [adam@jaamour.com](mailto:adam@jaamour.com)
- Discord:
  - <https://discord.gg/h3pgTENC>
- If you're ever at a tech event in London, hit me up 😊



---

---

# Thank you so much!

It was a pleasure to transfer some of my  
knowledge to you guys, I hope you all learned a  
lot and enjoyed it and will continue growing this  
new skill you have developed!

---

---