Introduction to Data Science Programming
Live Session

Week 3

Taylor Martin

Section 8

Remind me to start recording! ☺

& Sync your forks

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Today

- Big Ideas Presentation
- DEIB Agreement
- Part I: Sequences Week 3 activity & discussion
- Part II: Immutability Week 3 activity & discussion
- Interview Question Practice
- Survey

Different Levels

- This course is aimed at getting new programmers up to speed
- If you need more of a challenge, projects are the best way to go
  - Weekly challenge projects on solid-garbanzo
  - https://github.com/rfordatascience/tidytuesday
- Data science interview questions - mock interviews
  - Set up for yourself
  - Get together with another classmate to make it more realistic.
- If you just love lectures, we have things for you too ☺:
  - week_03_summary.ipynb
  - Resources (Optional) folder on bcourses
  - In several of the weeks' folders on bcourses, there are folders named things like "Week_3_OPTIONAL" with more detailed/advanced topics

Big Ideas Presentation

- Take it away Geon & William…

DEIB Agreement

- On slack

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Week 3 Activity: Part I Breakout (15 min)

- See the file week_03_activity.ipynb
- Stop at “3. The mutability exercise”

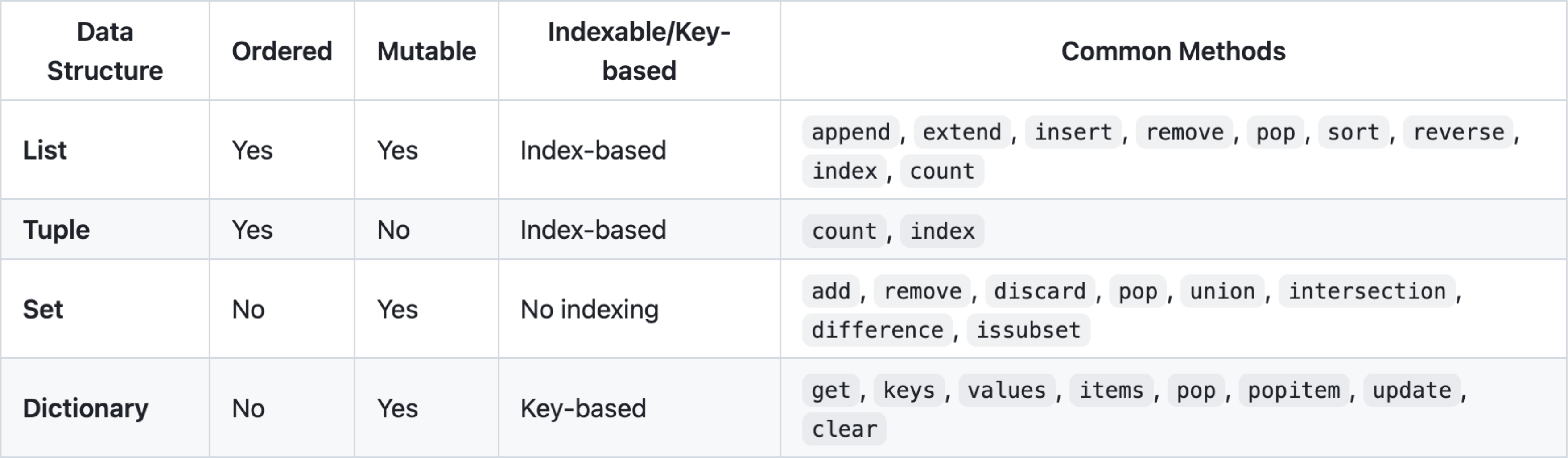
Sequences

- How did you handle the range for:
  [2, 4, 6, 8, 10, 12, 13, 14, 15, 17, 19, 21]
- What does pop do here?

```
while len(string_list) > 0:
    word= string_list.pop()
```

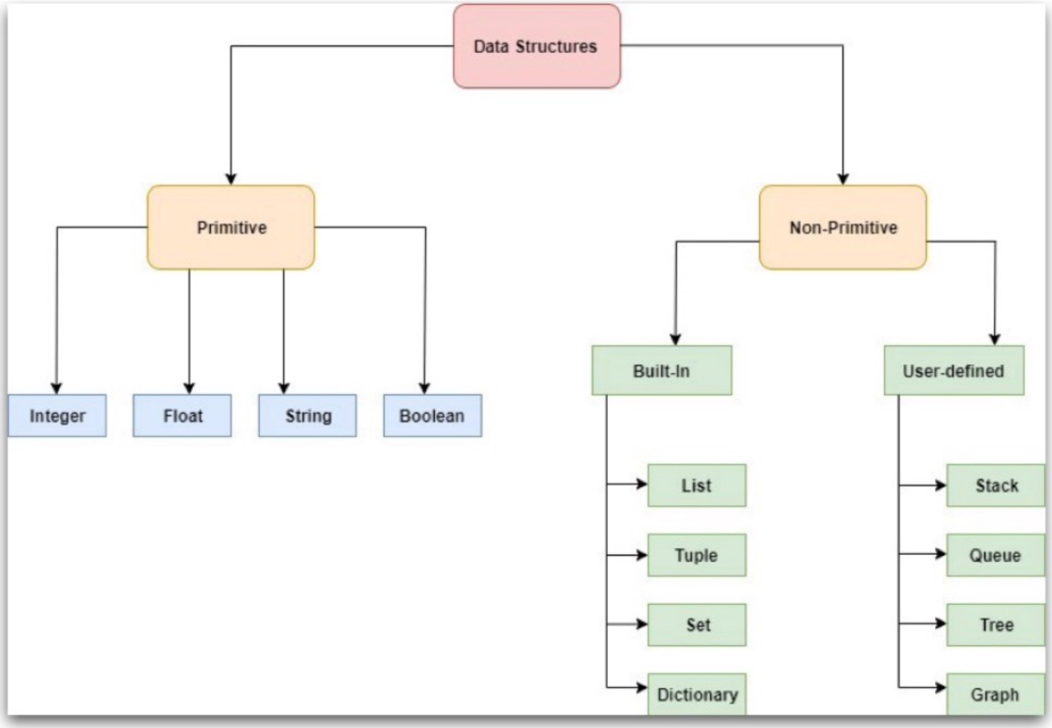
- In an OOP language, objects have properties and behaviors(or methods).
- What are some of the the properties and methods of sets, tuples, lists & dictionaries? *[Whiteboard]*

Sequences as Objects



Data Structure	Ordered	Mutable	Indexable/Key-based	Common Methods
List	Yes	Yes	Index-based	`append`, `extend`, `insert`, `remove`, `pop`, `sort`, `reverse`, `index`, `count`
Tuple	Yes	No	Index-based	`count`, `index`
Set	No	Yes	No indexing	`add`, `remove`, `discard`, `pop`, `union`, `intersection`, `difference`, `issubset`
Dictionary	No	Yes	Key-based	`get`, `keys`, `values`, `items`, `pop`, `popitem`, `update`, `clear`

Data Structures



Immutability

- In Python, immutability:
  - Property of an object that prevents its internal state from being changed after it is created.
- Key points of immutability:
  - The object's contents remain constant after creation.
  - If you need to "modify" it, you must create a new object with the desired changes.
- 2. Examples of Immutable Objects:
  - Tuples
  - Strings
  - Numbers: Integers, floats, and other numeric types
- See immutability.ipynb

Week 3 Activity: Part II Breakout (15 min)

- See the file week_03_activity.ipynb
- “3. The mutability exercise”, aka, Mutability can lead to issues when values are changed unexpectedly.

Modify in place, copy & deepcopy?

- See the file mod-in-place-copy-deepcopy.ipynb

Data Science Interview Practice

- week-3-interview-qs.ipynb

W200.8 Martin Week 3 Survey

- https://forms.gle/5htvVFhUBzf37S5K8

Berkeley
SCHOOL OF INFORMATION
MIDS • DATA SCIENCE PROGRAM

Challenge Yourself

- Challenge project
- Interview q&a
- Week 3 optional material on bcourses