Collection Types

- List
  - Ordered, changeable, duplicates allowed
  - Can store mixed data types
    List = [1,2,3,4,5]
- Dictionary
  - Ordered changeable, duplicates not allowed
  - Can store mixed data types
    Dictionary = {1:[1, 2], 2:[3, 4, 5]}
- Set
  - Unordered, unchangeable, duplicates not allowed
  - Values can be added or removed but values themselves cannot be changed
  - Cannot have lists or dictionaries as values
  - If you add a duplicate doesn't actually change anything in the data
    Can store mixed data types
    Set = set([1,2,3,4,5])
- Tuple
  - Unordered, unchangeable, duplicates allowed
  - Used if you want to ensure that data remains constant and is not accidentally modified
  - Can store mixed data types
    Tuple = (1, 2, 3, 4, 5)

(ChatGPT Revision of Notes)

Lists:

- Ordered: Elements are stored in a specific order and can be accessed by index.
- Mutable: You can modify their contents (add, remove, or change elements).
- Commonly used for sequences of items.
- ```
  my_list = [1, 2, 3, 4, 5]
  ```
- ```
  # Access the first element (index 0) first_element = my_list[0] # Returns 1
  ```

Dictionaries:

- Ordered (Python 3.7+): Elements maintain insertion order.
- Key-Value Pairs: Consist of key-value pairs, where keys are unique and used for quick data retrieval.
- Commonly used for mapping or associating values.
- ```
  my_dict = {"name": "Alice", "age": 30, "city": "New York"}
  my_dict = dict(name="Bob", age=25, city="Los Angeles")
  ```
- ```
  # Access the "name" key name = my_dict["name"] # Returns "Alice"
  ```

Sets:

- Mutable: You can add and remove elements, but the elements within the set must be immutable.
- Store Only Unique Elements: Sets store only unique elements, making them useful for removing duplicates.
- ```
  my_set = set([1, 2, 3, 4, 5])
  ```
- ```
  # Check if 3 is in the set contains_3 = 3 in my_set # Returns True
  ```
- Sets you can only check if a value is contained in the set

Tuples:

- Immutable: Elements cannot be changed once a tuple is created.
- Often used when data should remain constant and not be modified.
- ```
  my_tuple = (1, 2, 3, 4, 5)
  ```
- ```
  second_element = my_tuple[1] # Returns 2
  ```

Facebook_Friends

Name	City	Age
Matt	Los Angeles	27
Dave	San Francisco	30
Tim	Oakland	33

Column-Oriented:

- Grouping by features
- Top to bottom

Row-Oriented

- Grouping by a single observation
- Left to right

Loops:

- While Loop
  - `While condition:`
- For Loops
  - `For thing in collection:`

Useful Methods

- Dictionaries
  - https://www.w3schools.com/python/python_ref_dictionary.asp
  - .values()
  - .items()
  - .keys()
- Lists
  - https://www.w3schools.com/python/python_ref_list.asp
  - .len()
  - .append()
  - .sort()
- Miscellaneous
  - .range()
  - .print()
  - .split()
  - .type()
  - .int()
  - .str()