Week 5 - Wednesday

COMP 1800

Last time

- What did we talk about last time?
- Data structures
- Lists

Questions?

Assignment 4

Statistics

Mean

- A common operation we do with numerical data is to find the **mean** (average)
- A simple arithmetic mean adds up all the data in a list and then divides by the total number in the list

A mean function

- Let's implement a mean function that:
  - Adds up everything (assuming it's numerical data)
  - Divides by the total number of items
  - Returns the value

```
def mean(values):
```

The maximum value

- Finding the maximum (or minimum) value in a list can also be useful
- Algorithm:
  - Assume the first thing in the list is the biggest value and put it in a variable
  - Loop through every item in the list
    - If an item is bigger than the value we currently think is biggest, store it as the biggest

```
def findMax(values):
```

Built-in functions for max and min

- Fortunately, we don't have to write those functions again because Python has built-in versions:
  - **`max()`** Finds the largest element in a list
  - **`min()`** Finds the smallest element in a list

```
numbers = [3, 8, 14, -2, 6]
print(max(numbers)) # prints 14
words = ['if', 'you', 'take', 'the', 'A', 'train']
print(min(words)) # prints 'A'
```

Median

- The **median** of a list of numbers is the value that's in the middle, if the list is sorted
- Or, if the length of the list is even, it's the average of the two values nearest the middle

A median function

- Let's implement a median function that:
  - Sorts the list (using the **`sort()`** method)
  - If the number of elements in the list is odd, return the middle value
  - Otherwise, return the average of the two middle values

```
def median(values):
```

Dictionaries

Dictionaries

- A dictionary goes by many names:
  - Map
  - Lookup table
  - Symbol table
- The idea is a table that has a two columns, a key and a value
- You can store, lookup, and change the value based on the key

Examples

- A dictionary can be applied to almost anything:

Key	Value
Spiderman	Climbing and webs
Wolverine	Super healing
Professor X	Telepathy
Human Torch	Flames and flying
Deadpool	Super healing
Mr. Fantastic	Stretchiness

- The key doesn't have to be a string
- But it should be unique

Key	Value
1500	Introduction to Computer Science
1600	Introduction to Programming
2000	Object-Oriented Design
2100	Data Structures
3100	Software Engineering

Dictionaries in Python

- You can create a dictionary in Python
  - Enclosed in curly braces (**{ }**)
  - With a colon (**:**) between each key-value pair

```
superheroes = {'Spiderman' : 'Climbing and webs',
 'Wolverine' : 'Super healing', 'Professor X' :
 'Telepathy', 'Human Torch' :
 'Flames and flying', 'Deadpool' :
 'Super healing', 'Mr. Fantastic' : 'Stretchiness'}
```

Accessing values by key

- Like lists, you can index into a dictionary with square brackets
- **Unlike** lists, you put the key into the square brackets, not a number

```
print(superheroes['Spiderman'])
# prints 'Climbing and webs'
```

- You can also change the value for a given key with square brackets

```
superheroes['Spiderman'] = 'Science stuff'
```

Keys and values

- Dictionaries allow you to get a data structure that contains all the keys using the **keys()** method

```
print(superheroes.keys())
# 'Spiderman', 'Wolverine', etc.
```

- You can also get all the values using the `values()` method

```
print(superheroes.values())
# 'Climbing and webs', 'Super healing', etc.
```

- These structures aren't lists, but you can iterate over them with a `for` loop

```
for key in superheroes.keys():
  print(key)
```

Other dictionary operations

- The **in** operator lets us see if a key is in a dictionary

```
if 'Spiderman' in superheroes:
  print('We have a webslinger!')
```

- You can also remove a key from a dictionary with the **del** operator

```
del superheroes['Spiderman'] # no more Spiderman!
```

Mode

- The mode of a list of values is the value that occurs most frequently
- If there is more than one value that appears most frequently:
  - One option is to make a list of all of such elements
  - Another option is to say that there is no mode

Finding the mode

- Algorithm:
  - Make an empty dictionary
  - Loop through every item in the list
    - If it's in the dictionary, add 1 to the value stored for that key
    - Otherwise, put 1 as the value for that key
  - Get all the values from the dictionary
  - Find the maximum of those values
  - Make an empty list of modes
  - Loop through all the keys in the dictionary
    - If its value is the maximum value
      - Put it in the list
  - Return the list of modes

```
def findMode(values):
```

Upcoming

Next time…

- Review for Exam 1
- Work time for Assignment 4

Reminders

- Review chapters 1 through 4 of the textbook
- Work on Assignment 4