CS 209
Data Structures and Mathematical Foundations

01 / 31 / 2024

Instructor: Michael Eckmann

Today's Topics

- Questions/Comments?
- Continue Python review
  - Dictionaries
  - sorted
  - Writing our own classes – the classes we write will be new types

Word frequency

- Let's write a program utilizing split and a dictionary to determine word frequency in a file.
- We did that last class --- let's reread the code we have already and add more functionality to it.

Dictionary methods and sorted

- These can be called on a dictionary object to return a list of information:
  - items()
  - keys()
  - values()
- Let's see what each of these returns
- sorted(____) is a function that **returns** a sorted list of the parameter

Dictionary methods and sorted

- sorted( ) has an optional key parameter that is a **function** that determines what the list will be sorted by (the sort key).
- The word key in the parameter of sorted() is not the same meaning as the key in a dictionary.
- The function will expect as a parameter 1 element of the 1st parameter to sorted.

class

- Recall what a type is in Python. e.g. str is a type for string data, int is a type for integer data and so on.
- Python allows programmers to define their own type by creating a class.
- Let's look at the str class (which is built-in to Python) to get a sense of the kinds of things it contains (data and methods that work on that data).
  - help(str)

class

- Think of a **class** as containing
  - Data (aka **instance variables**)
  - **Methods** (functions that work on one or more instance variables of one **object** of the class)
- An **object** is an instance of a class.
- Example:

  word = ‘Skidmore’

  another_word = ‘College’
- word and another_word are both objects of the class str.

class

- To allow a class to have objects created of that class type (that is, for a class to be an instantiable class) a special method called the constructor should be defined.

def **__init__**(self)

- Can have more parameters but self is first.
- Instance variables are preceded by **self.** within methods of the class.
- The __init__ method is called automatically when instantiating an object of a class type.
- e.g. __init__ in str class is called when we do:

word = 'Skidmore'

- Or more explicitly:

another_word = str('College')

class

- Instance variables that start with __ (two underscores) are private otherwise they are public.
- The effect of private instance variable are they cannot be referenced via an object. Public instance variables may be referenced via an object.

Special methods / operator overloading

- Some special methods in classes
  - __str__ --- gets called by str(__)
  - __eq__ --- overloads the == operator
  - __gt__
  - __lt__
  - __ge__
  - __le__

class

- Let's define a class that represents a Card.
- Then let's create Card objects in a program and use them.
- If there's time, let's start to create a Deck class.
- Things to think about:
  - What data represents a Card? That is, what does a Card have? Answers should become instance variables in the Card class.
  - What methods should exist in Card? That is, what can we do to/with a Card? Answers should become methods in the Card class.
  - Similar questions should be asked about a Deck,