SI 506 Week 11

Objectives

- Understand the concept of classes
  - How to define classes
  - Common class methods (`__init__ and __str__`)
  - Sorting class instances by attribute

Materials

- The iTunes search API. Documentation can be seen here: https://www.apple.com/itunes/affiliates/resources/documentation/itunes-store-web-service-search-api.html
- "User-defined Classes" section in the textbook: here

Step 1: Create a Song Class

- Attributes:
  - `artist_name` [string]
  - `track_title` [string]
  - `track_url` [string]
  - `is_explicit` [boolean]
  - `track_duration` [integer]
- Methods:
  - `__init__()`
  - `__str__()`
    When an instance of this class is printed, it should look like the following:
    `<Song Title> by <Artist> | <Track Duration> ms`
    Thus, if you had the song "Bohemian Rhapsody" by Queen, it would print:
    `Bohemian Rhapsody by Queen | 4000 ms`

Step 2: Create a sample instance of the Song class

Use the following information to create your instance, and save it in the variable `sample_inst`:

- The artist name is "Queen"
- The song title is "Bohemian Rhapsody"
- The track url is "https://itunes.apple.com/us/album/bohemian-rhapsody/932648190?i=932648449&uo=4"

- The track time is 40000
- The track is not explicit

Now you can experiment with your sample instance:

- Print the instance to see if your `__str__` method prints the song as expected.

  ```
  print(sample_inst)
  ```

- Try printing attributes of the instance, for example:

  ```
  print(sample_inst.track_url)
  print(sample_inst.is_explicit)
  ```

Step 3: Fetch Song Data from iTunes and return Song instances

- Modify the `get_from_itunes()` function from previous discussion sections to return a list of Song objects instead of just the song titles.

  Hint: The Song object attributes should map to the following keys from the iTunes API response
  - `artist_name` → (artistName) [string]
  - `track_title` → (trackName) [string]
  - `track_url` → (trackViewUrl) [string]
  - `is_explicit` → (trackExplicitness) [boolean]
  - `track_duration` → (trackTimeMillis) [integer]

```
import json
import requests

def get_from_itunes(name, mtype="song"):
    baseurl = "https://itunes.apple.com/search"
    parameters = {}
    parameters["term"] = name
    parameters["entity"] = mtype
    print("Making request to iTunes API...")
    response = requests.get(baseurl, params=parameters)
    python_obj = json.loads(response.text)

    # use python_obj to create instances of the Song class
```

- Call your modified **`get_from_itunes()`** function for songs by the band “Queen” and print the returned songs.

Step 4: Sort the list based on the Song title

- Call your modified **`get_from_itunes()`** function for songs by the band “Queen”.
- Sort the returned list of Song instances based on the `track_title` attribute.
- Print the sorted list. The first few lines of your output will look like this:

```
A Kind of Magic by Queen | 261823 ms
Another One Bites the Dust by Queen | 215336 ms
Another One Bites the Dust by Queen | 215336 ms
Another One Bites the Dust (Remix) by Queen, Wyclef Jean & Pras
Michel | 260721 ms
Bicycle Race by Queen | 181201 ms
Bicycle Race by Queen | 181201 ms
Bohemian Rhapsody by Queen | 355145 ms
Bohemian Rhapsody by Queen | 355145 ms
Crazy Little Thing Called Love by Queen | 164904 ms
Crazy Little Thing Called Love by Queen | 164904 ms
Don't Stop Me Now by Queen | 210333 ms
Don't Stop Me Now by Queen | 210333 ms
```

Challenge if you are done:

- Add a method `play()` to the Song class, that opens its `track_url` in a web browser.
  - Hint: Use the `webbrowser` module’s `open()` function.
- Call the `play()` method on the `sample_inst` you created in step 2.