Homework 6: APIs, JSON, and Caching

In this assignment, you will get data using the Star Wars API (SWAPI). You will also store the data in a cache file so that you can retrieve the data from the cache instead of requesting data from the API repeatedly.

This assignment does not require you to generate an API key, but we strongly recommend reading the SWAPI documentation first. For this assignment, you will mostly be working with the People resources, which has different information on characters within the Star Wars universe.

Strongly Recommended

Choose an online JSON viewer. We recommend printing the API data/cache data and pasting it in the viewer to examine the structure of the data. Here are few of the many available options for JSON viewers:

1. https://jsonformatter.org/
2. https://jsoneditoronline.org/

Tasks

def load_json(filename):

This function reads a cached JSON file (*filename*) and returns a dictionary with JSON data or an empty dictionary if the cache does not exist. Hint: use a *try except*

def write_json(filename, dict):

This function encodes the cache dictionary into JSON format and writes the JSON to the cache file (*filename*) to save the search results.

def get_swapi_info(url, params=None):

This function checks whether the *params* dictionary has been specified, then makes a request to access data with the *url* and *params* given, if any. If the request is successful, it returns a dictionary representation of the decoded JSON. If the request is unsuccessful, it prints out "Exception!" and returns **None**.

def cache_all_pages(people_url, filename):
The SWAPI People resources has multiple pages. This function uses the passed url to get information from each page and write it out to a cache file. It first checks if the page number is in the dictionary returned by the function ***load_json***. If the page number does not exist in the dictionary, it makes a request to get the data (using ***get_swapi_info***), then adds the data to the dictionary (the key is the page number (Ex: **page 1**) and the value is the **results** (see example below). Then, write out the dictionary to a file(*filename*) using ***write_json***.

```
{
    "page 1": [
        {
            "name": "Luke Skywalker",
            "height": "172",
            "mass": "77",
            "hair_color": "blond",
            "skin_color": "fair",
            "eye_color": "blue",
            "birth_year": "19BBY",
            "gender": "male",
            "homeworld": "https://swapi.dev/api/planets/1/",
            "films": [
                "https://swapi.dev/api/films/1/",
                "https://swapi.dev/api/films/2/",
                "https://swapi.dev/api/films/3/",
                "https://swapi.dev/api/films/6/"
            ],
            "species": [],
            "vehicles": [
                "https://swapi.dev/api/vehicles/14/",
                "https://swapi.dev/api/vehicles/30/"
            ],
            "starships": [
                "https://swapi.dev/api/starships/12/",
                "https://swapi.dev/api/starships/22/"
            ],
            "created": "2014-12-09T13:50:51.644000Z",
            "edited": "2014-12-20T21:17:56.891000Z",
            "url": "https://swapi.dev/api/people/1/"
        },
        {
            "name": "C-3PO",
            "height": "167",
            "mass": "75",
            "hair_color": "n/a",
            "skin_color": "gold",
```


def get_starships(filename):

This function accesses the starships' url for each character (if any) from the cache file and passes it to the ***get_swapi_info*** function to get data about a character's starship. It returns a dictionary with the character's name as a key and a list of the names of their starships as the value. Do not include the characters that don't have starships in the dictionary.

Note: It might take a bit longer than usual to run the unit tests as you are making multiple requests.

Extra Credit - 6 points

def calculate_bmi(filename):

This function calculates the Body Mass Index (BMI) of each character if their height and mass is known and returns a dictionary with the name of the character as the key and the BMI as the value. Do not include the characters that are missing a height and/or mass values in the dictionary. The formula for BMI is:

$$BMI = \frac{weight\ in\ kg}{(height\ in\ m)^2} \text{ or } BMI = \frac{weight\ in\ kg}{(height\ in\ cm)^2} * 10000$$

Check the documentation to see the unit of measurement for mass and height.

Grading Rubric

Item	***Points***
load_json	***10***
write_json	***10***
get_swapi_info	***10***
cache_all_pages	***15***
get_starships	***15***
Extra credit	***6***
