Assignment 6

In this assignment you will grab some data from the Digital Public Library of America and save it to a file.

Submission:

- Save your code in a file named `hw6.py`
- Put your name in a comment at the top.
- Submit it via CCLE. You do NOT need to submit the resulting csv file.

Part 1 - Set Up

1a. Install the `requests` package using `pip`.

```
pip install requests
```

If you have both Python2 and Python3 installed on your machine (this will be the case for most Mac users) then you will need to specify that you want to install the Python3 package, so instead you would do the following:

```
pip3 install requests
```

1b. Get your DPLA API key:http://dp.la/info/developers/codex/policies/#get-a-key. You should be able to do this using `curl`. This is a command line utility that should already be installed on Macs. Just perform the following at the command line (change the email address to yours):

```
curl -v -XPOST http://api.dp.la/v2/api_key/YOUR_EMAIL@example.com
```

For Windows users, you can download `curl` from here:https://curl.haxx.se/download.html. Just scroll down to the Generic Win32 or Win64 download links depending on if you have a 32-bit or 64-bit machine. If that proves problematic, you can use Python to send the API key request instead. Write a short script that has the following code in it (change the email address to yours):

```
import requests

requests.post('http://api.dp.la/v2/api_key/YOUR_EMAIL@example.com')
```

1c. Check your email for the API key. Use this in the following part of the assignment.

Part 2 - Getting Data and Saving It

Following the example in the lecture notes, make a call to the DPLA API with a general keyword search for the artist *Rene Magritte*.

The API limits your result set to 10 items by default. Let's increase it to 25 by adding `page_size` to the URL parameters.

Convert the JSON string to Python data structures using the `json()` method. See the example in the lecture notes.

We want metadata for the objects. That can be found in the `docs` part of the data dictionary.

Loop through the docs and write the following fields to a CSV file named **rene_magritte.csv**.

- Get the **title**, which can be found in

  ```
  doc['sourceResource']['title']
  ```

  - However, sometimes there can be more than one title, so it will give you a list. We will just grab the first one.
  - This means you need to test if the data is a string or a list:

  ```
   title = doc['sourceResource']['title']
          if type(title) == list:
              title = title[0]
  ```

- Now get the **date**.
  - There can be multiple **date** entries as well. So perform the same kind of test as above and grab the first entry.
  - The **date** entry is actually a dictionary with a*display date*, a *begin date*, and an *end date*. Let's just grab the display date.

    ```
    date = date['displayDate']
    ```

- Finally, get the **creator** field.
  - Not all objects have a **creator** field. This means your code will throw an error if you try to grab it using a dictionary index:

    ```
    doc['sourceResource']['creator']
    ```

  - To avoid such an error, use the dictionary**get(key, default)** method.
    - The **default** argument lets you specify what it should return if the key does not exist. If you don't give a default it will use**None**.
    - Let's tell it to give us '*Anonymous*' if there is no creator:

      ```
      creator = doc['sourceResource'].get('creator', 'Anonymous')
      ```

  - As with the first two pieces of data, test if**creator** is a list, and grab the first entry if it is.
- Once you have grabbed the three pieces of data, write it out as a row in the csv file.