Week 5: Assignment 2
Scrap the Website

[*Last Updated on:* ***05th May 2021, 14:00 Hrs***]

Aim

Your big brother (Tutor in an academy) has recently got a new batch of students for the upcoming academic year.

The names, email address and other details are hosted on a website of the academy for these batch of students.

He wants you to find out the *username* in the email addresses of these students given the *full name*.

So your program when provides the appropriate *username*, he can then create accounts for these students in the virtual software the academy uses for teaching few subjects.

Are you in for this? - says your big brother!

Given

One file is provided to solve this assigment.

- Skeleton program file: **`assignment2.py`**
- The skeleton consists of one function which you have to modify:
  - **`getUserName()`**

Procedure

- Open the skeleton program file, **`assignment2.py`** .
- You will notice pre-written comments included in skeleton program for your assistance to solve the assignment.
- One function to modify is:

- `getUserName()`

Function Name	`getUserName()`
Purpose	Gets the username from a website of the person whose full name is provided as input.
Input Arguments	`url_website` : [***str***] URL of website to scrape `full_name` : [***str***] Full name of person whose username is to be returned by scraping the given website
Output Arguments	`username` : [***str***] Username of person for the provided Full name
Example Call	***username = getUserName(url_website, full_name)***

- Points to be considered:
  - URL of website: https://www.cse.iitb.ac.in/archive/page222?batch=MTech1
  - The details of each students is present inside a table with:
    - ***Full Name*** in first column written in **bold** format and hyperlinked to the individual webpage of students.
    - ***Email address*** in second column written in **bold** format divided into these parts:
      1. *username* - the part of email address before the **@**
      2. *@* - an image of the special character (@)
      3. *cse* - part1 of email domain in text
      4. *.* - an image of the dot (.)
      5. *iitb* - part2 of email domain in text
      6. *.* - an image of the dot (.)
      7. *ac* - part3 of email domain in text
      8. *.* - an image of the dot (.)
      9. *in* - part4 of email domain in text
  - You can verify the above formatting and division of these fields using the **Inspect** tool in browser which can be invoked with ***Ctrl + Shift + C*** and just hover on the fields you wish to inspect.

NOTE: All of the above logic to scrape the *username* with the given *full_name* should be written ***ONLY*** inside the `getUserName` function under the block of:

####### ADD YOUR CODE HERE ########

- To run and debug your solution, type the below command in Terminal:

```
$ python3 assignment2.py
```

  This command will run the Python script `assignment2.py`.

- Refer the **Expected Output** section below and debug your code to get the correct output.

Expected Output

- For example, the values of variables `url_website` and `full_name` are defined in `main` function which are as follows in lines starting with **#**.
- The expected output of program `assignment2.py` i.e. the function `getUserName` should return
  - the appropriate username for the given full name listed on the website

-OR-

- the text: ***Full Name does not exist on website*** if the given full name is not found listed on the website

- You can test your solution with following values and their expected outputs:

```
# url_website = "https://www.cse.iitb.ac.in/archive/page222?batch=MTech1"
# full_name = "ABHISHEK"
$ python3 assignment2.py
The username of ABHISHEK is: abhishekthakur

# url_website = "https://www.cse.iitb.ac.in/archive/page222?batch=MTech1"
# full_name = "ADITYA KRISHNAKUMAR"
$ python3 assignment2.py
The username of ADITYA KRISHNAKUMAR is: Full Name does not exist on website
```

Grading and Submission Instructions

- Navigate to the folder where the ***ey-mooc-grader-sfc*** application resides.
- To grade your solution, run the `check` command of the application as follows:

```
$ ./ey-mooc-grader-sfc check -w 5 -a 2 Week_5/Assignment_2/assignment2.py
```

- This will run your program `assignment2.py` against random test cases and grade it. Marks and appropriate remarks will be provided as shown in Figure 1.
- Your program file `assignment2.py` , marks scored and remarks will get uploaded to the MOOC portal.

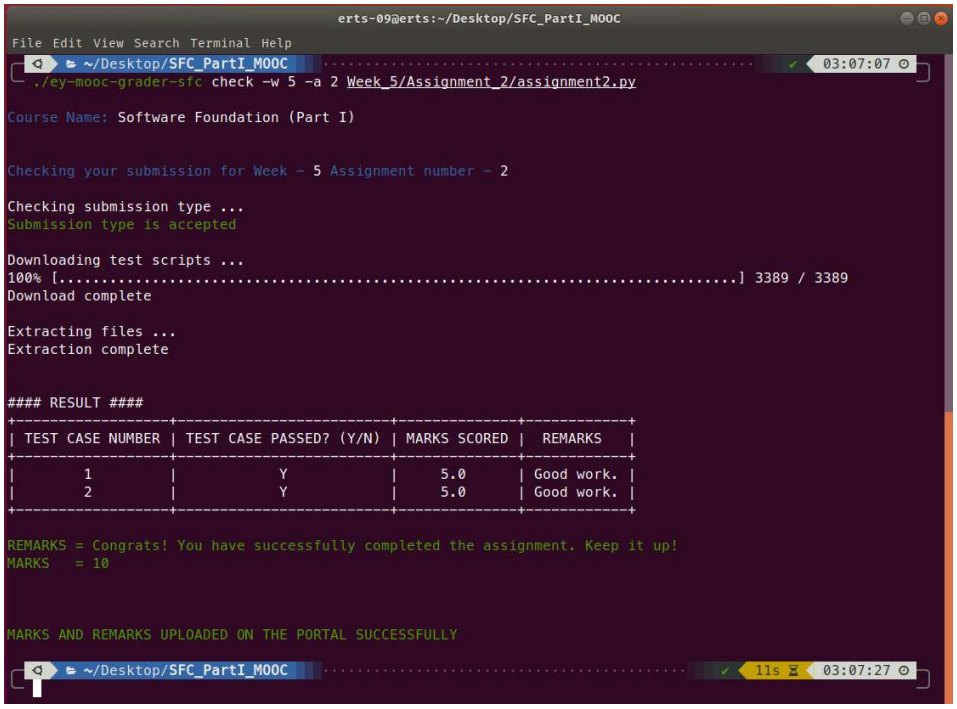

Figure 1: Output of running check command for Week 5 Assignment 2

- While grading this assignment, you might get an error like this:

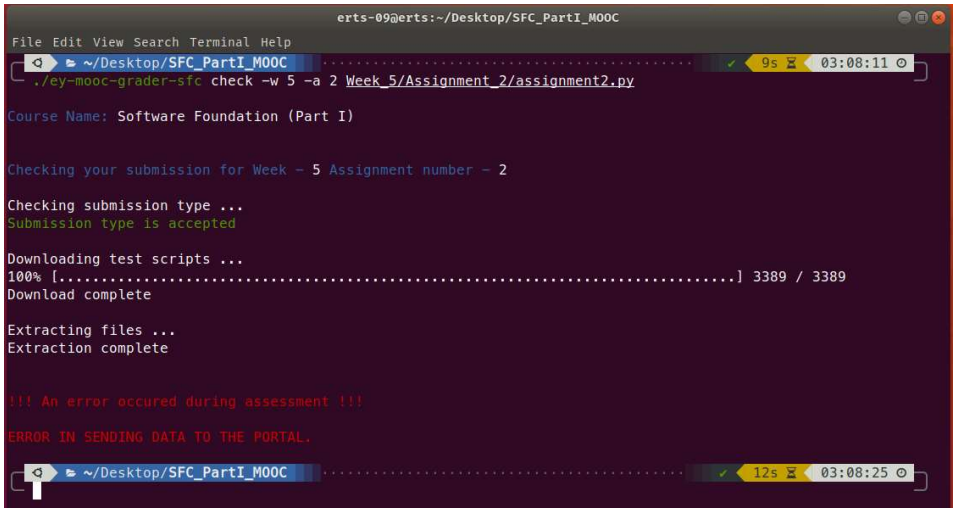

Figure 2: Error in running check command for Week 5 Assignment 2

- The cause of the above error is your code took longer than **10 seconds** when Grader App tried to check or evaluate it. But you need not worry, make sure you have ***very good Internet connection*** and you will be sorted.
- You can verify this by running the `status` command of the application as given below, refer Figure 2.

```
$ ./ey-mooc-grader-sfc status -w 5 -a 2
```

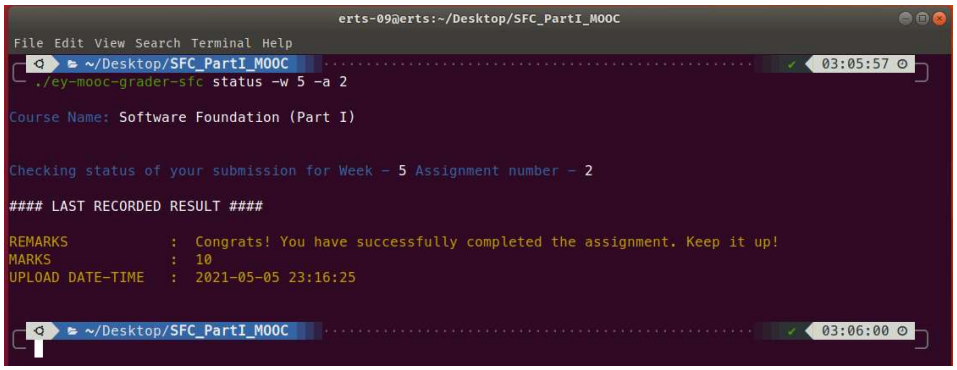

Figure 3: Output of running status command for Week 5 Assignment 2

References

- Official Python documentation on Requests
- Beautiful Soup Documentation

All The Best!