Web Scraping_Assignment 3

March 14, 2024

[128]:

```
import requests
from bs4 import BeautifulSoup
import os
import time


os.makedirs("hockey data")
#creating a directory hockey data

def scrape_page(page_number):
    html_page = requests.get(f'https://www.scrapethissite.com/pages/forms/?
 ↪page_num={page_number}').text
    soup = BeautifulSoup(html_page, 'lxml')
    return soup.find_all('tr', class_='team')

'''In the above function, we have customised the url by passing a integer to␣
 ↪the fuction. We can invoke this function to
move through various pages and then using the beautiful soup we then find the␣
 ↪<tr> tags of the html page associated with the
class named "team" '''

def hockey_data():
    for page_number in range(1, 25):
        all_data = scrape_page(page_number)
        save_data(page_number, all_data)

'''We have created another function which we will use to pass page_number␣
 ↪values to the function above and invoke it
and this function which has another function invocation of the save_data␣
 ↪function which is invoked next'''

def save_data(page_number, all_data):

    with open(f'hockey data/hockeydata_page{page_number}.txt', 'w') as f:
        for index, data in enumerate(all_data):
            team_name = data.find('td', class_='name').text.strip()
            year = data.find('td', class_='year').text.strip()
            wins = data.find('td', class_='wins').text.strip()
            losses = data.find('td', class_='losses').text.strip()
            overtime_losses = data.find('td', class_='ot-loses').text.strip()␣
 ↪if data.find('td', class_='ot-loses') else ''
            win_percentage = data.find('td', class_='pct text-success').text.
 ↪strip() if data.find('td', class_='pct text-success') else ''
            goals_for = data.find('td', class_='gf').text.strip()
            goals_against = data.find('td', class_='ga').text.strip()
            diff_between_goals = data.find('td', class_='diff text-success').
 ↪text.strip() if data.find('td', class_='diff text-success') else ''

            f.write(f"Team Name: {team_name}\n")
            f.write(f"Year: {year}\n")
            f.write(f"Wins: {wins}\n")
            f.write(f"Losses: {losses}\n")
            f.write(f"Overtime Losses: {overtime_losses}\n")
            f.write(f"Win Percentage: {win_percentage}\n")
            f.write(f"Goals For: {goals_for}\n")
            f.write(f"Goals Against: {goals_against}\n")
            f.write(f"Difference Between Goals: {diff_between_goals}\n")

            f.write("\n")
            #adding a newline between each row of data

        print(f'Data saved for page {page_number} whose file name is␣
 ↪{filename}')

'''Next we have created the actual logic for moving through each data point in␣
 ↪the pagenated web pages with a table of data.
we create the function save_data to which we pass the values of page number␣
 ↪through page_number and all_data which
has the returned value/'result' of the invoked function␣
 ↪scrape_page(page_number). It is through the scrape_page function we
find the tr and td tags associated with a particular page. For page 1 we have a␣
 ↪text file named hockeydata_page1 text file
and similarly for page 2 we have a text file namedhockeydata_page2 text file.␣
 ↪In the for loop, we use the td tags and
their corresponding classes to extract the cell values and write them to the␣
 ↪text files. We have also included a print statement
which tells us that a particular page's data is saved in a particular file'''

'''It is important to note that we have used if else to ignore the empty values␣
 ↪in the tables of the pages and to create
an empty string instead if and when we encounter them.'''
```

```
if __name__ == '__main__':
    hockey_data()'
    time.sleep(1)
#we add a small delay of 1 second between each request
'''Here finally we invoke hockey_data function. The flow of the program is such␣
 ↪that by invoking hockey data function we
pass he page value to scrape_page function (from 1 to 24 which are the pages of␣
 ↪the pagenated webpage) whose returned
value/'result' is stored in the all_data object and then save_data function is␣
 ↪invoked with the values page_number and
all_data passed to it which is where actual extraction and writing of the data␣
 ↪into the text file takes place.'''
```

```
Data saved for page 1 to hockey data/hockeydata_page1.txt
Data saved for page 2 to hockey data/hockeydata_page2.txt
Data saved for page 3 to hockey data/hockeydata_page3.txt
Data saved for page 4 to hockey data/hockeydata_page4.txt
Data saved for page 5 to hockey data/hockeydata_page5.txt
Data saved for page 6 to hockey data/hockeydata_page6.txt
Data saved for page 7 to hockey data/hockeydata_page7.txt
Data saved for page 8 to hockey data/hockeydata_page8.txt
Data saved for page 9 to hockey data/hockeydata_page9.txt
Data saved for page 10 to hockey data/hockeydata_page10.txt
Data saved for page 11 to hockey data/hockeydata_page11.txt
Data saved for page 12 to hockey data/hockeydata_page12.txt
Data saved for page 13 to hockey data/hockeydata_page13.txt
Data saved for page 14 to hockey data/hockeydata_page14.txt
Data saved for page 15 to hockey data/hockeydata_page15.txt
Data saved for page 16 to hockey data/hockeydata_page16.txt
Data saved for page 17 to hockey data/hockeydata_page17.txt
Data saved for page 18 to hockey data/hockeydata_page18.txt
Data saved for page 19 to hockey data/hockeydata_page19.txt
Data saved for page 20 to hockey data/hockeydata_page20.txt
Data saved for page 21 to hockey data/hockeydata_page21.txt
Data saved for page 22 to hockey data/hockeydata_page22.txt
Data saved for page 23 to hockey data/hockeydata_page23.txt
Data saved for page 24 to hockey data/hockeydata_page24.txt
```

[129]:

```
os.getcwd()
'''We use getcwd of os module to get an idea of where the current directory is␣
 ↪present locally in the PC
when the program is run through Jupyter Notebook.'''
```

[129]:

```
'C:\\Users\\bvsro'
```