

WEB SCRAPING

WORKSHEET – 1

In Q1 to Q9, only one option is correct, Choose the correct option:

1. Which of the following extracts information from user generated content?
A) Java script tagging B) Web scraping
C) A/B testing D) MROCs

ANS: B) Web scraping

2. Which of the following is not a web scraping library in python?
A) selenium B) Beautiful soup
C) Requests D) scrapy

ANS: C) scrapy

3. Selenium tests _____?
A) Browser based applications B) DOS applications
C) GUI applications D) All of the above

ANS: A) Browser based applications

4. Task of crawling is performed by a complex software which is known as:
A) Scraper B) Crawler
C) Boat D) Spider

ANS: A) Scraper

5. Which of the following commands is used to access name of a tag in BeautifulSoup?
A) tag.attrs B) tag.name
C) tag.id D) tag['id']

ANS: B) tag.name

6. Which of the following is the default parser in BeautifulSoup?
A) html.parser B) html5lib
C) lxml D) lxml-xml

ANS: C) lxml

7. In selenium the webdriver is used to?
A) design a test using selenese
B) test a web application on firefox only
C) execute tests on HtmlUnit browser
D) to download any content from a webpage

ANS: D) to download any content from a webpage

8. In selenium, `driver.find_elements_by_xpath('given xpath')` returns:
- A) the first webelement associated with the 'given xpath'
 - B) the url of first webelement associated with the 'given xpath'
 - C) the list of all webelements associated with the 'given xpath'
 - D) all the attributes of the first webelement associated with the 'given xpath'

ANS: C) the list of all webelements associated with the 'given xpath'

9. The script `'window.scrollTo(0,a)` scrolls the webpage by?
- A) 'a' number of horizontal spaces
 - B) 'a' number of lines
 - C) 'a' number of pixels horizontally
 - D) 'a' number of pixels vertically

ANS: D) 'a' number of pixels vertically

In Q10, more than one options are correct, Choose all the correct options:

10. Which of the following is(are) tags of HTML?
- A) `<a>`
 - B) ``
 - C) `<image>`
 - D) `<href>`

ANS: A) `<a>` B) `` D) `<href>`

Q10 to Q13 are subjective answer type questions, Answer them briefly.

11. What is the main difference between a web scraper and a web crawler?

Web Scraper: -

- A **Web Scraper** is a tool used for web scraping. It can be defined as a method/technique to access & extract data from websites through any browser within few times. It also helps to save the data achieved in local storage.
- It is used to download information from the websites.
- Used in the field of Machine learning, Marketing etc.
- It may or may not obey Robot.txt

Web Crawler: -

- A **Web Crawler** also called as **Spiders**. It can be defined as a method where it accesses a website and analyse the links to other webpages present within it and again those links are accessed for more links this form a deep web of information. The aim of crawling is to build search engine index using the information achieved by crawling.
- It is used for indexing of web pages.
- It is used by search engines like Google and Yahoo to rank the web pages and give better result to the user
- It always obeys Robot.txt protocol

12. What is '**robots.txt**' file? What is the use of '**robots.txt**' file?

- In the web any website can be accessed, crawled and indexed by bots and search engine and this leads to a high information gain.

- Sometimes the secret or private data of the website also get disclosed so to prevent this from happening the webpage designers use a text file called as Robots.txt which put some constraint over the search engine and the bots.
- It simply protects the important information by preventing the search engine and bots to access it. Robots.txt is a text file that comes under REP protocol which regulates the way of crawling and indexing.
- It is a text file which contains a list of allowed and disallowed site and based on this list the bot crawls only the allowable sites mentioned. This text file helps to keep important data private and inaccessible by any bot or search engine.

13. What are static and dynamic web pages?

Static Web Pages:

- The static webpages are fixed in nature and they show the same content to all the user accessing the page.
- Static webpages have fixed layout
- These are irresponsive of any user action
- These are created with the help of HTML, CSS and a text editor like notepad with very little time and effort.
- Static webpages are informational in nature which doesn't changes more often and don't even require interaction.

Dynamic Web Pages:

- The information shown in a dynamic webpage are user responsive and much more functional. The content of the page changes for every user accessing it.
- These are created using languages like HTML, Java Script, PHP, CSS with an IDE like IntelliJ.
- It allows the user to interact with the information provided.

Q14 and Q15 are programming practice questions. Solve it using JUPYTER NOTEBOOK and paste the solution in your answer sheets.

14. Write a python program to check whether a webpage contains a title or not.

Code lines:

```
def title(url):
    from selenium import webdriver
    import time
    driver=webdriver.Chrome(executable_path="C:\\Users\\mishr\\driver\\cd.exe")
    driver.get(url)
    time.sleep(5)
    title= driver.title
    if title== None:
        print("NO TITLE FOUND IN ", driver.current_url)
    else:
        print("BROWSER USED:\t",driver.name.upper(),
              "\nURL OPENED:\t",driver.current_url,
              "\nTITLE OF THE PAGE:\t",title)
```

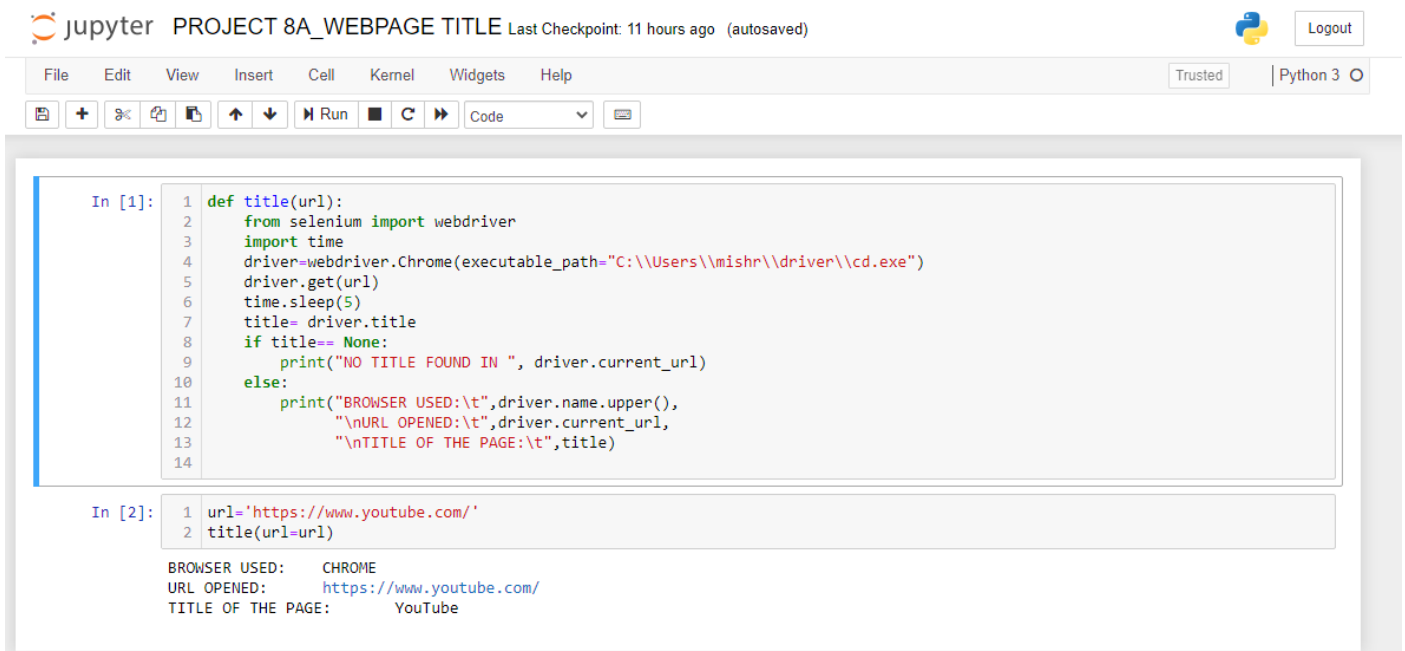
Execution Input:

```
url='https://www.youtube.com/'
title(url=url)
```

Output:

```
BROWSER USED:      CHROME
URL OPENED:        https://www.youtube.com/
TITLE OF THE PAGE: YouTube
```

Screenshot:



```
Jupyter PROJECT 8A_WEBPAGE TITLE Last Checkpoint: 11 hours ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [1]: 1 def title(url):
2         from selenium import webdriver
3         import time
4         driver=webdriver.Chrome(executable_path="C:\\Users\\mishr\\driver\\cd.exe")
5         driver.get(url)
6         time.sleep(5)
7         title= driver.title
8         if title== None:
9             print("NO TITLE FOUND IN ", driver.current_url)
10        else:
11            print("BROWSER USED:\t",driver.name.upper(),
12                  "\nURL OPENED:\t",driver.current_url,
13                  "\nTITLE OF THE PAGE:\t",title)
14
In [2]: 1 url='https://www.youtube.com/'
2        title(url=url)

BROWSER USED:    CHROME
URL OPENED:      https://www.youtube.com/
TITLE OF THE PAGE:  YouTube
```

15. Write a python program to access the search bar and search button on images.google.com.

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

driver=webdriver.Chrome(executable_path=r"C:\Users\mishr\driver\cd.exe")
url="https://images.google.com/"
driver.get(url)

#URL of the page opened
print("URL:\t",driver.current_url)

URL:      https://images.google.com/

#Accessing the search bar using find_element_by_name method
search_bar=driver.find_element(By.NAME,"q")



#clicking on search button
search_button=driver.find_element_by_xpath("//div[@class='FAuhyb']")

#giving input
search_word= 'pagani zonda'
search_bar.send_keys(search_word)











#clicking on search button
search_button.click()
time.sleep(7)

driver.quit()
```

SCREENSHOT

 jupyter PROJECT 8B_SEARCH BAR & SEARCH BUTTON Last Checkpoint: 11 hours ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

          Code

```
In [1]: 1 import time
2 from selenium import webdriver
3 from selenium.webdriver.common.keys import Keys
4 from selenium.webdriver.common.by import By

In [2]: 1 driver=webdriver.Chrome(executable_path=r"C:\Users\mishr\driver\cd.exe")
2 url="https://images.google.com/"
3 driver.get(url)

In [3]: 1 #URL of the page opened
2 print("URL:\t",driver.current_url)

URL:      https://images.google.com/

In [4]: 1 #Accessing the search bar using find_element_by_name method
2 search_bar=driver.find_element(By.NAME,"q")

In [5]: 1 #clicking on search button
2 search_button=driver.find_element_by_xpath("//div[@class='FAuhyb']")

In [6]: 1 #giving input
2 search_word= 'pagani zonda'
3 search_bar.send_keys(search_word)

In [7]: 1 #clicking on search button
2 search_button.click()

In [8]: 1 time.sleep(7)

In [9]: 1 driver.quit()
```