

Program No 14

Program to implement a webcrawler

Code

```
import requests
from bs4 import BeautifulSoup

url = "https://www.rottentomatoes.com/top/bestofrt/"
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/63.0.3239.132 '
    'Safari/537.36 QIHU 360SE '
}
f = requests.get(url, headers=headers)
movies_lst = []
soup = BeautifulSoup(f.content, 'lxml')
movies = soup.find('table', {
    'class': 'table'
}).find_all('a')
print(movies)
num = 0
for anchor in movies:
    urls = 'https://www.rottentomatoes.com' + anchor['href']
    movies_lst.append(urls)
print(movies_lst)
num += 1
movie_url = urls
movie_f = requests.get(movie_url, headers=headers)
movie_soup = BeautifulSoup(movie_f.content, 'lxml')
movie_content = movie_soup.find('div', {'class': 'movie_synopsis clamp clamp-6 js-clamp'})
print(num, urls, '\n', 'Movie:' + anchor.string.strip())

print('Movie info:' + movie_content.string.strip())
```

Output

```
C:\Programming\Python39\python.exe C:/Users/asifk/PycharmProjects/ML/16-02-2022/webcrawler.py
[<a class="unstyled articleLink" href="/m/it_happened_one_night">
    It Happened One Night (1934)</a>, <a class="unstyled articleLink" href="/m/citizen_kane">
    Citizen Kane (1941)</a>, <a class="unstyled articleLink" href="/m/the_wizard_of_oz_1939">
    The Wizard of Oz (1939)</a>, <a class="unstyled articleLink" href="/m/modern_times">
    Modern Times (1936)</a>, <a class="unstyled articleLink" href="/m/black_panther_2018">
    Black Panther (2018)</a>, <a class="unstyled articleLink" href="/m/parasite_2019">
    Parasite (Gisaengchung) (2019)</a>, <a class="unstyled articleLink" href="/m/avengers_endgame">

    Baby Driver (2017)</a>, <a class="unstyled articleLink" href="/m/spider_man_homecoming">
    Spider-Man: Homecoming (2017)</a>, <a class="unstyled articleLink" href="/m/godfather_part_ii">
    The Godfather, Part II (1974)</a>, <a class="unstyled articleLink" href="/m/the_battle_of_algiers">
    The Battle of Algiers (La Battaglia di Algeri) (1967)</a>]
['https://www.rottentomatoes.com/m/it_happened_one_night', 'https://www.rottentomatoes.com/m/citizen_kane', 'https://www.rottentomatoes.com/m/avengers_endgame']

Process finished with exit code 0
```

Program No 15

Code:

```
from bs4 import BeautifulSoup
import requests

pages_crawled = []

def crawler(url):
    page = requests.get(url)
    soup = BeautifulSoup(page.text, 'html.parser')
    links = soup.find_all('a')

    for link in links:
        if 'href' in link.attrs:
            if link['href'].startswith('/wiki') and ':' not in link['href']:
                if link['href'] not in pages_crawled:
                    new_link = f"https://en.wikipedia.org{link['href']}"
                    pages_crawled.append(link['href'])
                    try:
                        with open('data.csv', 'a') as file:
                            file.write(f'{soup.title.text};{soup.h1.text};{link["href"]}\n')
                        crawler(new_link)
                    except:
                        continue
crawler('https://en.wikipedia.org')
```

Output

1	Wikipedia, the free encyclopedia;Main Page;/wiki/Wikipedia
2	Wikipedia, the free encyclopedia;Main Page;/wiki/Free_content
3	Wikipedia, the free encyclopedia;Main Page;/wiki/Encyclopedia
4	Wikipedia, the free encyclopedia;Main Page;/wiki/English_language
5	Wikipedia, the free encyclopedia;Main Page;/wiki/Wonderful_Parliament
6	Wikipedia, the free encyclopedia;Main Page;/wiki/Legislative_session
7	Wikipedia, the free encyclopedia;Main Page;/wiki/Parliament_of_England
8	Wikipedia, the free encyclopedia;Main Page;/wiki/Westminster_Abbey
9	Wikipedia, the free encyclopedia;Main Page;/wiki/Richard_II_of_England
10	Wikipedia, the free encyclopedia;Main Page;/wiki/Favourite
11	Wikipedia, the free encyclopedia;Main Page;/wiki/Hundred_Years%27_War
12	Wikipedia, the free encyclopedia;Main Page;/wiki/Lord_Chancellor
13	Wikipedia, the free encyclopedia;Main Page;/wiki/Michael_de_la_Pole,_1st_Earl_of_Suffolk
14	Wikipedia, the free encyclopedia;Main Page;/wiki/Impeachment

Program No 16

Implement a program to scrap the webpage of any popular website

Code


```
import requests
from bs4 import BeautifulSoup
import csv
import lxml

url = "https://www.values.com/inspirational-quotes"
r = requests.get(url)
print(r.content)
soup = BeautifulSoup(r.content, 'lxml')
print(soup.prettify())
quotes = []
table = soup.find('div', attrs={'id': 'all_quotes'})
for row in table.findAll('div',
                        attrs={'class': 'col-6 col-lg-3 text-center margin-30px-bottom sm-margin-30px-top'}):
    quote = {}
    quote['theme'] = row.h5.text
    quote['url'] = row.a['href']
    quote['img'] = row.img['src']
    quote['lines'] = row.img['alt'].split(" #")[0]
    quote['author'] = row.img['alt'].split(" #")[1]
    quotes.append(quote)
filename = 'inspirational_quotes.csv'
with open(filename, 'w', newline='') as f:
    w = csv.DictWriter(f, ['theme', 'url', 'img', 'lines', 'author'])
    w.writeheader()
    for quote in quotes:
        w.writerow(quote)
```

Output

```
C:\Programming\Python39\python.exe C:/Users/asifk/PycharmProjects/ML/16-02-2022/scrap.py
b'<!DOCTYPE html>\n<html class="no-js" dir="ltr" lang="en-US">\n    <head>\n        <title>Inspirational Quotes - Motivational Quotes - Leadership Quotes | PassItOn.com</title>\n    <!DOCTYPE html>\n    <html class="no-js" dir="ltr" lang="en-US">\n    <head>\n    <title>\n        Inspirational Quotes - Motivational Quotes - Leadership Quotes | PassItOn.com\n    </title>\n    <meta charset="utf-8"/>\n    <meta content="text/html; charset=utf-8" http-equiv="content-type"/>\n    <meta content="IE=edge" http-equiv="X-UA-Compatible"/>\n    <meta content="width=device-width,initial-scale=1.0" name="viewport"/>\n    <meta content="The Foundation for a Better Life | Pass It On.com" name="description"/>\n    <link href="/apple-touch-icon.png" rel="apple-touch-icon" sizes="180x180"/>\n    <link href="/favicon-32x32.png" rel="icon" sizes="32x32" type="image/png"/>\n    <link href="/site.webmanifest" rel="manifest" type="manifest"/>\n    </div>\n    </div>\n    </div>\n</footer>\n<a class="scroll-top-arrow" href="javascript:void(0);">\n    <i class="ti-arrow-up">\n    </i>\n</a>\n<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.4/jquery.js">\n</script>\n<script crossorigin="anonymous" integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1cLHTMga3JDZwrnQq4sF86dIHNDz0W1" src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js">\n</script>\n<script crossorigin="anonymous" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js">\n</script>\n<script src="/assets/pofo-1a7dc0d92519266568dcfcc8a6e53534.js">\n</script>\n</body>\n</html>
```

Process finished with exit code 0

 PyCharm and plugin updates
Restart to activate plugin updates

Program No 17

Python program for natural language processing-Ngram(without using in-built functions)

Code

```
def generate_ngrams(text, WordsToCombine):  
    words = text.split()  
    output = []  
    for i in range(len(words) - WordsToCombine + 1):  
        output.append(words[i:1 + WordsToCombine])  
    return output
```

```
x = generate_ngrams(text="this is a very good book study", WordsToCombine=3)  
print(x)
```

Output

```
C:\Programming\Python39\python.exe C:/Users/asifk/PycharmProjects/ML/16-02-2022/ngram.py  
[['this', 'is', 'a', 'very']]  
  
Process finished with exit code 0
```

Program No 18

Python program for natural language processing-Ngram(with using in-built functions)

Code

```
import nltk
nltk.download()
from nltk.util import ngrams

sampletext="This is a very good book to study"
NGRAMS=ngrams(sequence=nltk.word_tokenize(sampletext),n=2)
for grams in NGRAMS:
    print(grams)
```

Output

```
C:\Programming\Python39\python.exe "C:/Users/asifk/PycharmProjects/ML/16-02-2022/ngram withbuiltin funct.py"
showing info https://raw.githubusercontent.com/nltk/nltk\_data/gh-pages/index.xml
('This', 'is')
('is', 'a')
('a', 'very')
('very', 'good')
('good', 'book')
('book', 'to')
('to', 'study')

Process finished with exit code 0
```

Program No 19

Python program for natural language processing- part of speech tagging

Code

```
from cgitb import text

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

stop_words = set(stopwords.words('english'))

text
txt = "Sukanya, Rajib and Naba are my good friends. " \
      "Sukanya is getting married next year. " \
      "Marriage is a big step in ones's life" \
      "It is both exciting and frightening. " \
      "But friendship is a sacred bond between people." \
      "It is a special kind of love between us. " \
      "Many of you must have tried searching for a friend " \
      "but never found the right one."

tokenized = sent_tokenize(txt)
for i in tokenized:
    wordsList = nltk.word_tokenize(i)
    wordsList = [w for w in wordsList if not w in stop_words]
    tagged = nltk.pos_tag(wordsList)
    print(tagged)
```

Output

```
C:\Programming\Python39\python.exe C:/Users/asifk/PycharmProjects/ML/16-02-2022/nlp.py
[('Sukanya', 'NNP'), (',', ','), ('Rajib', 'NNP'), ('Naba', 'NNP'), ('good', 'JJ'), ('friends', 'NNS'), ('.', '.')]
[('Sukanya', 'NNP'), ('getting', 'VBG'), ('married', 'VBN'), ('next', 'JJ'), ('year', 'NN'), ('.', '.')]
[('Marriage', 'NN'), ('big', 'JJ'), ('step', 'NN'), ('ones', 'NNS'), ('''s'', 'POS'), ('LifeIt', 'NN'), ('exciting', 'VBG'), ('frightening', 'NN'), ('.', '.')]
[('But', 'CC'), ('friendship', 'NN'), ('sacred', 'VBD'), ('bond', 'NN'), ('people.It', 'NN'), ('special', 'JJ'), ('kind', 'NN'), ('love', 'VB'), ('us', 'PRP'), ('.', '.')]
[('Many', 'JJ'), ('must', 'MD'), ('tried', 'VB'), ('searching', 'VBG'), ('friend', 'NN'), ('never', 'RB'), ('found', 'VBD'), ('right', 'JJ'), ('one', 'CD'), ('.', '.')]

Process finished with exit code 0
```