```
pip install bs4
```

```
Requirement already satisfied: bs4 in /usr/local/lib/python3.6/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.6/dist-packages
```

```
pip install requests
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (2.23
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
```

```
pip install texttable
```

```
Collecting texttable
  Downloading https://files.pythonhosted.org/packages/06/f5/46201c428aebe0eecfa83df66bf
Installing collected packages: texttable
Successfully installed texttable-1.6.3
```

P1 - Scraping Quotes from website

URL: Quotes Website

scrapes the website and saves quotes to a file inspirational_quotes.csv

```
#Python program to scrape website and save quotes to a file inspirational_quotes.csv

import requests
from bs4 import BeautifulSoup
import csv

URL = "http://www.values.com/inspirational-quotes"
r = requests.get(URL)

soup = BeautifulSoup(r.content, 'html5lib')

quotes=[] # a list to store 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-to
  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)
```

P2 - Scraping Covid-19 stats

URL: COVID-19 STATS COUNTRY WISE

```
# URl to Scrap: https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spr

import requests
from bs4 import BeautifulSoup
import texttable as tt

# URL for scrapping data
url = 'https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/'

# get URL's html
page = requests.get(url)
soup = BeautifulSoup(page.text, 'html.parser')

data = []

# soup.find_all('td') will scrape every element in the url's table
data_iterator = iter(soup.find_all('td'))
# data_iterator is the iterator of the table

# This loop will keep repeating till there is data available in the iterator
while True:
  try:
    country = next(data_iterator).text
    confirmed = next(data_iterator).text
    deaths = next(data_iterator).text
    continent = next(data_iterator).text

    # For 'confirmed' and 'deaths', make sure to remove the commas and convert to int
    data.append((
      country,
      int(confirmed.replace(',', '')),
      int(deaths.replace(',', '')),
      continent
    ))

  # StopIteration error is raised when there are no more elements left to iterate through
  except StopIteration:
    break

# Sort the data by the number of confirmed cases
data.sort(key = lambda row: row[1], reverse = True)


# create texttable object
table = tt.Texttable()
table.add_rows([(None, None, None, None)] + data)  # Add an empty row at the beginning for th
table.set_cols_align(('c', 'c', 'c', 'c'))  # 'l' denotes left, 'c' denotes center, and 'r' d
table.header((' Country ', ' Number of cases ', ' Deaths ', ' Continent '))

print(table.draw())
```

P3 - Scraping GPU Card Product Information

URL: GPU Card Info

```
from bs4 import BeautifulSoup as soup
from urllib.request import urlopen as ureq

my_url = 'https://www.newegg.com/p/pl?d=graphics+card&nm_mc=KNC-GoogleKWLess-Search-Broad&cm_
uclient = ureq(my_url)
page_html = uclient.read()
uclient.close()

page_soup = soup(page_html, "html.parser")
print(page_soup)
#print(page_soup.body.id)
containers = page_soup.findAll("div",{"class":"item-container"})

filename = "products.csv"
f = open(filename, "w")
headers = "brand, productname, shipping\n"
f.write(headers)
print("before for")
for container in containers:
  print("after for")
  brand = container.div.div.a.img["title"]
  title_container = container.findAll("a",{"class":"item-title"})
  product_name = title_container[0].text
  shipping = container.findAll("li",{"class":"price-ship"})
  shipping_price = shipping[0].text.strip()
  print(brand)
  print(product_name)
  print(shipping_price)
  f.write(brand + "," + product_name.replace(",","|") + "," + shipping_price + "\n")
f.close()
```

```
pip install fake_useragent
```

```
Collecting fake_useragent
  Downloading https://files.pythonhosted.org/packages/d1/79/af647635d6968e2deb57a208d309
Building wheels for collected packages: fake-useragent
  Building wheel for fake-useragent (setup.py) ... done
  Created wheel for fake-useragent: filename=fake_useragent-0.1.11-cp36-none-any.whl siz
  Stored in directory: /root/.cache/pip/wheels/5e/63/09/d1dc15179f175357d3f5c00cbffbac37
Successfully built fake-useragent
Installing collected packages: fake-useragent
Successfully installed fake-useragent-0.1.11
```

P4 - Web Scraping Customer Reports

URL: http://www.consumerreports.org/cro/a-to-z-index/products/index.htm

```
import requests
from fake_useragent import UserAgent

url = 'http://www.consumerreports.org/cro/a-to-z-index/products/index.htm'
file_name = 'consumer_reports.txt'              # output file name having complete HTML conte

user_agent = UserAgent()

page = requests.get(url,headers={'user-agent':user_agent.chrome})
with open(file_name,'w') as file:
    file.write(page.content.decode('utf-8')) if type(page.content) == bytes else file.write(p
```

```
from bs4 import BeautifulSoup
import re

def read_file():
    file = open('consumer_reports.txt')
    data = file.read()
    file.close()
    return data

soup = BeautifulSoup(read_file(),'lxml')
```

```
'''
When you inspect the consumer_report website we can see that all catagories From A - Z:
for example air conditioners are present in 'a' tag which inside the 'div' tag.
hence we use below code to extract all the 'a' tags.
'''

all_divs = soup.find_all('div',attrs={'class':'crux-body-copy'})

#for div in all_divs:
  #print(div.a.string)

products = [div.a.string for div in all_divs]

for product in products:
    print(product)
    print()

'''
Assignment: remove the space before ad after the string and copy the content to
a csv file consumer_list_formatted.
'''


'''
Here we are using the same consumer_reports.txt and creating a dictionary with
product name as key and product link as value and trying to dispaly.
'''

from bs4 import BeautifulSoup
import re


def read_file():
    file = open('consumer_reports.txt')
    data = file.read()
    file.close()
    return data

soup = BeautifulSoup(read_file(),'lxml')
products = {}   # product name - key and product link - value


product_names = [div.a.string for div in soup.find_all('div',class_='crux-body-copy')]

product_links = [div.a['href'] for div in soup.find_all('div',class_='crux-body-copy')]

products = {div.a.string:div.a['href'] for div in soup.find_all('div',class_='crux-body-copy'

for key,value in products.items():
    print(key , '   -->',value)
```

```
'''
Assignment: remove the space before ad after the string and copy the content (Name + URL) to
a csv file consumer_list_link.
'''
```

P5 - Scraping Multiple web Pages

Task is to Scrap java questions from codingbat website

URL: http://codingbat.com/java

I will divide the project into 3 parts:

1. First script will describe you how to fetch the link of each section of Java questions.
2. Secondly we will open each section(catagory)and we scrap link for each question.
3. Thirdly we will open each question and get the problem statement, example associated with it.

```
#Part 1 - script will describe you how to fetch the link of each section of Java questions.

import requests
from bs4 import BeautifulSoup
from fake_useragent import UserAgent


user_agent = UserAgent()
main_url = 'http://codingbat.com/java'
page = requests.get(main_url,headers={'user-agent':user_agent.chrome})
soup = BeautifulSoup(page.content,'lxml')

base_url = 'http://codingbat.com'

'''
Here we are scraping the link to each section.
Observe in inspect element that link is a ralative link (Warm-up) not absolute link
thus we used base_url above
'''
all_divs = soup.find_all('div',class_='summ')

#prints all the relative link
for div in all_divs:
    print(div.a['href']) #Here 'a' is a child of 'div' tag


#prints all the absolute link
for div in all_divs:
    print(base_url + div.a['href'])  #Here 'a' is a child of 'div' tag
```

```
#Secondly we will open each section and we scrap link for each question.
#--------Start - Same as above Script ----------------------------
import requests
from bs4 import BeautifulSoup
from fake_useragent import UserAgent


user_agent = UserAgent()
main_url = 'http://codingbat.com/java'
page = requests.get(main_url,headers={'user-agent':user_agent.chrome})
soup = BeautifulSoup(page.content,'lxml')

base_url = 'http://codingbat.com'

all_divs = soup.find_all('div',class_='summ')


# all_links has link for each section (Page 1)
all_links = [base_url + div.a['href'] for div in all_divs] # This is list Comprahension

#--------End - Same as above Script ----------------------------

#Below code is to get link for each/all the section

for link in all_links:
    #link correspons to 2nd page ex:https://codingbat.com/java/Warmup-1
    inner_page = requests.get(link,headers={'user-agent':user_agent.chrome})
    inner_soup = BeautifulSoup(inner_page.content,'lxml')

    #Now we need to scrap the link from 2nd inner page. (Inspect the HTML Page)

    div = inner_soup.find('div',class_='tabc')
    question_links = [base_url + td.a['href'] for td in div.table.find_all('td')] # has link
    print(question_links)

    break #on commenting it you will get complete links for all the sections
```

```
['http://codingbat.com/prob/p187868', 'http://codingbat.com/prob/p181646', 'http://codi
```

```
#Final Script

#part 1

import requests
from bs4 import BeautifulSoup
from fake_useragent import UserAgent
```

```
user_agent = UserAgent()
main_url = 'http://codingbat.com/java'
page = requests.get(main_url,headers={'user-agent':user_agent.chrome})
soup = BeautifulSoup(page.content,'lxml')

base_url = 'http://codingbat.com'

all_divs = soup.find_all('div',class_='summ')

all_links = [base_url + div.a['href'] for div in all_divs]


# part 2

for link in all_links:
    inner_page = requests.get(link,headers={'user-agent':user_agent.chrome})
    inner_soup = BeautifulSoup(inner_page.content,'lxml')
    div = inner_soup.find('div',class_='tabc')
    question_links = [base_url + td.a['href'] for td in div.table.find_all('td')]


# part 3

    for question_link in question_links:
        final_page = requests.get(question_link)
        final_soup = BeautifulSoup(final_page.content, 'lxml')
        indent_div = final_soup.find('div', attrs={'class':'indent'})

        problem_statement = indent_div.table.div.string

        siblings_of_statement = indent_div.table.div.next_siblings

        examples = [sibling for sibling in siblings_of_statement if sibling.string is not Non

        print(problem_statement)
        for example in examples:
            print(example)

        print('\n\n\n')
```