**Libraries**

Beautifulsoup4

Requests

**Import command**

from bs4 import BeautifulSoup

import requests

**code to connect to a url and then format data using bs**

from bs4 import BeautifulSoup

import requests

url = "https://boston.craigslist.org/search/sof"

response = requests.get(url)

#print(response)

data = response.text

#print(data)

soup = BeautifulSoup(data,'html.parser')

tags = soup.find\_all('a')

for tag in tags:

print(tag.get('href'))

**get data of specific classes**

titles =soup.find\_all("a",{"class":"result-title"})

for title in titles:

print(title.text)

adresses= soup.find\_all("span",{"class":"result-hood"})

for adress in adresses:

print(adress.text)

**get data from wrapper**

jobs = soup.find\_all('p',{'class':'result-info'})

for job in jobs:

title = job.find('a',{'class':'result-title'}).text

location\_tag = job.find('span',{'class':'result-hood'})

location = location\_tag.text[2:-1] if location\_tag else "N/A" -- **check if not exist**

date = job.find('time',{'class':'result-date'}).text

link = job.find('a',{'class':'result-title'}).get('href') -- **to get the link details**

print('Job Title:', title, '\nLocation', location, '\nDate:', date, '\nLink:', link, '\n---')

**move to next page and extract data**

while True:

response = requests.get(url)

data = response.text

soup = BeautifulSoup(data,'html.parser')

jobs = soup.find\_all('p',{'class':'result-info'})

for job in jobs:

title = job.find('a',{'class':'result-title'}).text

location\_tag = job.find('span',{'class':'result-hood'})

location = location\_tag.text[2:-1] if location\_tag else "N/A"

date = job.find('time', {'class': 'result-date'}).text

link = job.find('a', {'class': 'result-title'}).get('href')

job\_response = requests.get(link)

job\_data = job\_response.text

job\_soup = BeautifulSoup(job\_data, 'html.parser')

job\_description = job\_soup.find('section',{'id':'postingbody'}).text

job\_attributes\_tag = job\_soup.find('p',{'class':'attrgroup'})

job\_attributes = job\_attributes\_tag.text if job\_attributes\_tag else "N/A"

job\_no+=1

print('Job Title:', title, '\nLocation:', location, '\nDate:', date, '\nLink:', link,"\n", job\_attributes, '\nJob Description:', job\_description,'\n---')

url\_tag = soup.find('a',{'title':'next page'})

if url\_tag.get('href'):

url= 'https://boston.craigslist.org' + url\_tag.get('href')

print(url)

else:

break

print("Total Jobs:", job\_no)