GITHUB Topic and Repository Scraping Project

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import os
```

In [ ]:
```
# Get URL for scraping main topics
response= requests.get('https://github.com/topics')
```

In [ ]:
```
response.status_code
```

Out[ ]:
```
200
```

In [ ]:
```
len(response.text)
```

Out[ ]:
```
152059
```

In [ ]:
```
page_content=response.text
```

In [ ]:
```
# Get target page content on local machine
with open('webpage.html', 'w', encoding="utf-8") as f:
    f.write(page_content)
```

In [ ]:
```
# Create BeautifulSoup Object for parsing
soup= BeautifulSoup(page_content, 'html.parser')
```

In [ ]:
```
type(soup)
```

Out[ ]:
```
bs4.BeautifulSoup
```

Scraping Main topic .their Description & Topic URL in start

In [ ]:
```
topic_title_tags= soup.find_all('p', {'class': 'f3 lh-condensed mb-0 mt-1 Link--pr
```

In [ ]:
```
topic_title_tags[:5]
```

Out[ ]:
```
[<p class="f3 lh-condensed mb-0 mt-1 Link--primary">3D</p>,
 <p class="f3 lh-condensed mb-0 mt-1 Link--primary">Ajax</p>,
 <p class="f3 lh-condensed mb-0 mt-1 Link--primary">Algorithm</p>,
 <p class="f3 lh-condensed mb-0 mt-1 Link--primary">Amp</p>,
 <p class="f3 lh-condensed mb-0 mt-1 Link--primary">Android</p>]
```

In [ ]:
```
topic_title_desc= soup.find_all('p', {'class', 'f5 color-fg-muted mb-0 mt-1'})
```

In [ ]:
```
topic_title_desc[:5]
```

Out[ ]:

```
[<p class="f5 color-fg-muted mb-0 mt-1">
           3D modeling is the process of virtually developing the surface and stru
cture of a 3D object.
         </p>,
 <p class="f5 color-fg-muted mb-0 mt-1">
           Ajax is a technique for creating interactive web applications.
         </p>,
 <p class="f5 color-fg-muted mb-0 mt-1">
           Algorithms are self-contained sequences that carry out a variety of tas
ks.
         </p>,
 <p class="f5 color-fg-muted mb-0 mt-1">
           Amp is a non-blocking concurrency library for PHP.
         </p>,
 <p class="f5 color-fg-muted mb-0 mt-1">
           Android is an operating system built by Google designed for mobile devi
ces.
         </p>]
```

In [ ]:

```
topics_url= soup.find_all('a', {'class': 'no-underline flex-1 d-flex flex-column'}
```

In [ ]:

```
topic0_url="https://github.com" + topics_url[0]['href']
```

In [ ]:

```
topic_title=[]

for title in topic_title_tags:
    topic_title.append(title.text)
print(topic_title)
```

```
['3D', 'Ajax', 'Algorithm', 'Amp', 'Android', 'Angular', 'Ansible', 'API', 'Arduin
o', 'ASP.NET', 'Atom', 'Awesome Lists', 'Amazon Web Services', 'Azure', 'Babel',
'Bash', 'Bitcoin', 'Bootstrap', 'Bot', 'C', 'Chrome', 'Chrome extension', 'Command
line interface', 'Clojure', 'Code quality', 'Code review', 'Compiler', 'Continuous
integration', 'COVID-19', 'C++']
```

In [ ]:

```
topic_desc=[]

for title in topic_title_desc:
    topic_desc.append(title.text.strip())
print(topic_desc[:5])
```

```
['3D modeling is the process of virtually developing the surface and structure of
a 3D object.', 'Ajax is a technique for creating interactive web applications.',
'Algorithms are self-contained sequences that carry out a variety of tasks.', 'Amp
is a non-blocking concurrency library for PHP.', 'Android is an operating system b
uilt by Google designed for mobile devices.']
```

In [ ]:

```
topic_url=[]

for url in topics_url:
    topic_url.append("https://github.com" + url['href'])
print(topic_url[:5])
```

```
['https://github.com/topics/3d', 'https://github.com/topics/ajax', 'https://githu
b.com/topics/algorithm', 'https://github.com/topics/amphp', 'https://github.com/to
pics/android']
```

Create a Dictionary for saving scraped data from Main Topics

In [ ]:

```
topic_dict={'topic': topic_title,
'topic_desc': topic_desc,
'topic_url': topic_url}
```

```
topic_df= pd.DataFrame(topic_dict)
```

In [ ]:
```
topic_df.head()
```

Out[ ]:

	topic	topic_desc	topic_url
0	3D	3D modeling is the process of virtually develo...	https://github.com/topics/3d
1	Ajax	Ajax is a technique for creating interactive w...	https://github.com/topics/ajax
2	Algorithm	Algorithms are self-contained sequences that c...	https://github.com/topics/algorithm
3	Amp	Amp is a non-blocking concurrency library for ...	https://github.com/topics/amphp
4	Android	Android is an operating system built by Google...	https://github.com/topics/android

In [ ]:
```
# Write main_topic_list file in CSV
topic_df.to_csv('main_topic_list.csv', index=None)
```

Now Scrap Username, Repository Name , Repository URL and Stars from each Topic Page

In [ ]:
```
topic_page_url=topic_url[0]
```

In [ ]:
```
topic_page_url
```

Out[ ]:
```
'https://github.com/topics/3d'
```

In [ ]:
```
response= requests.get(topic_page_url)
```

In [ ]:
```
response.status_code
```

Out[ ]:
```
200
```

In [ ]:
```
soup= BeautifulSoup(response.text, 'html.parser')
```

In [ ]:
```
user_name= soup.find_all('h3', {'class': 'f3 color-fg-muted text-normal lh-condense
```

In [ ]:
```
a_tags=user_name[0].find_all('a')
```

In [ ]:
```
a_tags[0].text.strip()
```

Out[ ]:
```
'mrdoob'
```

In [ ]:
```
a_tags[1].text.strip()
```

Out[ ]:
```
'three.js'
```

In [ ]:
```
base_url="https://github.com"
repo_url= base_url + a_tags[1]['href']
print (repo_url)
```

```
https://github.com/mrdoob/three.js
```

In [ ]:
```
stars=soup.find_all('span', {'class': 'Counter js-social-count'})
```

In [ ]:
```
stars[0].text
```

Out[ ]:
```
'85.6k'
```

In [ ]:
```
def parse_stars_count(stars_str):
    if (stars_str[-1] == 'k'):
        return int (float(k[:-1]) * 1000)
    return int(stars_str)
```

In [ ]:
```
parse_stars_count(stars[0].text)
```

Out[ ]:
```
85600
```

Encapsule all working for scraping Username, Repo Name , Repo URL in function

In [ ]:
```
def get_topic_page (page_url):
    response= requests.get(page_url)
    if response.status_code != 200:
        raise Exception('Failed to load Page  '.format(page_url))

    soup= BeautifulSoup(response.text, 'html.parser')

    return soup

def get_repo_info(user_tag, stars_tag):
    a_tags= user_tag.find_all('a')
    user_name=a_tags[0].text.strip()
    repo_name=a_tags[1].text.strip()
    repo_url1=base_url + a_tags[1]['href']
    stars_tag= parse_stars_count(stars_tag.text)
    return  user_name, repo_name, repo_url1, stars_tag

def get_topic_repos (soup):
    user_name= soup.find_all('h3', {'class': 'f3 color-fg-muted text-normal lh-cond
    stars=soup.find_all('span', {'class': 'Counter js-social-count'})

    topic_repo_dict={
    "user_name": [],
    "Repo_name": [],
    "Repo_URL": [],
    "Repo_stars": []
}

    for i in range (len(user_name)):
        repo_info= get_repo_info(user_name[i], stars[i])
        topic_repo_dict['user_name'].append(repo_info[0])
        topic_repo_dict['Repo_name'].append(repo_info[1])
        topic_repo_dict['Repo_URL'].append(repo_info[2])
        topic_repo_dict['Repo_stars'].append(repo_info[3])

    return pd.DataFrame (topic_repo_dict)
```

Testing the functions

In [ ]:
```
topic_url[6]
```

Out[ ]:
```
'https://github.com/topics/ansible'
```

In [ ]:
```
get_topic_repos(get_topic_page(topic_url[6]))
```

Out[ ]:

	user_name	Repo_name	Repo_URL	Repo_stars
0	ansible	ansible	https://github.com/ansible/ansible	85600
1	bregman-arie	devops-exercises	https://github.com/bregman-arie/devops-exercises	85600
2	trailofbits	algo	https://github.com/trailofbits/algo	85600
3	StreisandEffect	streisand	https://github.com/StreisandEffect/streisand	85600
4	MichaelCade	90DaysOfDevOps	https://github.com/MichaelCade/90DaysOfDevOps	85600
5	kubernetes-sigs	kubespray	https://github.com/kubernetes-sigs/kubespray	85600
6	ansible	awx	https://github.com/ansible/awx	85600
7	easzlab	kubeasz	https://github.com/easzlab/kubeasz	85600
8	geerlingguy	ansible-for-devops	https://github.com/geerlingguy/ansible-for-devops	85600
9	khuedoan	homelab	https://github.com/khuedoan/homelab	85600
10	Tikam02	DevOps-Guide	https://github.com/Tikam02/DevOps-Guide	85600
11	ansible-semaphore	semaphore	https://github.com/ansible-semaphore/semaphore	85600
12	geerlingguy	mac-dev-playbook	https://github.com/geerlingguy/mac-dev-playbook	85600
13	rundeck	rundeck	https://github.com/rundeck/rundeck	85600
14	KubeOperator	KubeOperator	https://github.com/KubeOperator/KubeOperator	85600
15	clong	DetectionLab	https://github.com/clong/DetectionLab	85600
16	netbootxyz	netboot.xyz	https://github.com/netbootxyz/netboot.xyz	85600
17	ansible-community	molecule	https://github.com/ansible-community/molecule	85600
18	litmuschaos	litmus	https://github.com/litmuschaos/litmus	85600
19	opendevops-cn	opendevops	https://github.com/opendevops-cn/opendevops	85600

Save all Topic Scalped Repository data in Data folder in CSV format

In [ ]:

```
os.makedirs('data', exist_ok=True)
for i in range(len(topic_url)):
    fname= topic_title[i] + '.csv'
    if os.path.exists(fname):
        print("The file already Exist".format(fname))
    topics_repos=get_topic_repos(get_topic_page(topic_url[i]))
    topics_repos.to_csv('data/' + fname, index=None)
```

In [ ]: