1.Pick a website and describe your objective

-Browse through different sites and pick on to scrape. Check the "Project Ideas" section for inspiration. -Identify the information you'd like to scrape from the site. Decide the format of the output CSV file. -Summarize your project idea and outline your strategy in a Juptyer notebook. Use the "New" button above.

Project outline: -we,re going to scrape the page https://github.com/topics -we will get list of topic,for each topic ,we will get topic title,topic page URL and topic description -For each topic ,we'll get the top 25 repositories in the topic from the topic page -For each repository we'll grab the repo name,user name,stars and URL -for each topic we will create a csv file in the format:

Repo Name,Username,Stars,Repo URL three.js,mrdoob,97300,https://github.com/mrdoob/three.js libgdx,libgdx,22500,https://github.com/libgdx

In [ ]:

```
```

2.Use the requests library to download web pages

In [1]:

```
!pip install requests --upgrade --quiet
```

In [2]:

```
import requests
```

In [3]:

```
topics_url = 'https://github.com/topics'
```

In [4]:

```
response = requests.get(topics_url)
```

In [5]:

```
response.status_code
```

Out[5]:

```
200
```

In [6]:

```
len(response.text)
```

Out[6]:

```
170725
```

In [7]:

```
page_contents=response.text
```

In [8]:

```
page_contents[:1000]
```

Out[8]:

```
'\n\n<!DOCTYPE html>\n<html\n  lang="en"\n  \n  data-color-mode="auto" data-light-theme="l
ight" data-dark-theme="dark"\n  data-a11y-animated-images="system" data-a11y-link-underlin
es="true"\n  >\n\n\n\n\n  <head>\n    <meta charset="utf-8">\n  <link rel="dns-prefetch" h
ref="https://github.githubassets.com">\n  <link rel="dns-prefetch" href="https://avatars.g
ithubusercontent.com">\n  <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaw
s.com">\n  <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">\n  
<link rel="preconnect" href="https://github.githubassets.com" crossorigin>\n  <link rel="p
reconnect" href="https://avatars.githubusercontent.com">\n\n  \n\n  <link crossorigin="ano
nymous" media="all" rel="stylesheet" href="https://github.githubassets.com/assets/light-0e
ace2597ca3.css" /><link crossorigin="anonymous" media="all" rel="stylesheet" href="http
```

```
s://github.githubassets.com/assets/dark-a167e256da9c.css" /><link data-color-theme="dark_d
immed" crossorigin="anonymous" media="a'
```

In [9]:

```python
with open('webpage.html','w') as f:
    f.write(page_contents)
```

In [ ]:

3.Use Beautiful Soup to parse and extract information

In [10]:

```python
!pip install beautifulsoup4 --upgrade --quiet
```

In [11]:

```python
from bs4 import BeautifulSoup
```

In [12]:

```python
doc = BeautifulSoup(page_contents, 'html.parser')
```

In [13]:

```python
selection_class='f3 lh-condensed mb-0 mt-1 Link--primary'
topic_title_tags = doc.find_all('p', {'class':selection_class})
```

In [14]:

```python
len(topic_title_tags)
```

Out[14]:

```
30
```

In [15]:

```python
topic_title_tags[:5]
```

Out[15]:

```
[<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 [ ]:

In [16]:

```python
desc_selector ='f5 color-fg-muted mb-0 mt-1'
topic_desc_tags = doc.find_all('p',{'class':desc_selector})
```

In [17]:

```python
topic_desc_tags[:5]
```

Out[17]:

```
[<p class="f5 color-fg-muted mb-0 mt-1">
           3D refers to the use of three-dimensional graphics, modeling, and animation in
various industries.
         </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 tasks.
         </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 devices.
         </p>]
```

In [18]:

```python
topic_title_tag0= topic_title_tags[0]
```

In [19]:

```python
div_tag = topic_title_tag0.parent
```

In [ ]:

In [20]:

```python
topic_link_tags = doc.find_all('a',{'class': 'no-underline flex-1 d-flex flex-column'})
```

In [21]:

```python
len(topic_link_tags)
```

Out[21]:

```
30
```

In [22]:

```python
topic0_url= "https://github.com"+ topic_link_tags[0]['href']
print(topic0_url)
```

```
https://github.com/topics/3d
```

In [ ]:

In [23]:

```python
topic_titles = []
topic_descriptions =[]

for tag in topic_title_tags:
    topic_titles.append(tag.text)

print(topic_titles)
```

```
['3D', 'Ajax', 'Algorithm', 'Amp', 'Android', 'Angular', 'Ansible', 'API', 'Arduino', 'AS
P.NET', 'Atom', 'Awesome Lists', 'Amazon Web Services', 'Azure', 'Babel', 'Bash', 'Bitcoi
n', 'Bootstrap', 'Bot', 'C', 'Chrome', 'Chrome extension', 'Command line interface', 'Cloj
ure', 'Code quality', 'Code review', 'Compiler', 'Continuous integration', 'COVID-19', 'C+
+']
```

In [24]:

```python
topic_descs = []

for tag in topic_desc_tags :
    topic_descs.append(tag.text.strip())

topic_descs[:5]
```

Out[24]:

```
['3D refers to the use of three-dimensional graphics, modeling, and animation in various i
ndustries.',
 '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 built by Google designed for mobile devices.']
```

In [25]:

```python
topic_urls = []
base_url='https://github.com'
```

Loading [MathJax]/extensions/Safe.js

```python
pic_link_tags:
    topic_urls.append(base_url + tag['href'])

topic_urls
```

Out[25]:

```
['https://github.com/topics/3d',
 'https://github.com/topics/ajax',
 'https://github.com/topics/algorithm',
 'https://github.com/topics/amphp',
 'https://github.com/topics/android',
 'https://github.com/topics/angular',
 'https://github.com/topics/ansible',
 'https://github.com/topics/api',
 'https://github.com/topics/arduino',
 'https://github.com/topics/aspnet',
 'https://github.com/topics/atom',
 'https://github.com/topics/awesome',
 'https://github.com/topics/aws',
 'https://github.com/topics/azure',
 'https://github.com/topics/babel',
 'https://github.com/topics/bash',
 'https://github.com/topics/bitcoin',
 'https://github.com/topics/bootstrap',
 'https://github.com/topics/bot',
 'https://github.com/topics/c',
 'https://github.com/topics/chrome',
 'https://github.com/topics/chrome-extension',
 'https://github.com/topics/cli',
 'https://github.com/topics/clojure',
 'https://github.com/topics/code-quality',
 'https://github.com/topics/code-review',
 'https://github.com/topics/compiler',
 'https://github.com/topics/continuous-integration',
 'https://github.com/topics/covid-19',
 'https://github.com/topics/cpp']
```

In [26]:

```
!pip install pandas --quiet
```

In [27]:

```
import pandas as pd
```

In [28]:

```
topics_dict = {
    'title': topic_titles,
    'descriptions':topic_descs,
    'url':topic_urls
}
```

In [29]:

```
topics_df = pd.DataFrame(topics_dict)
```

In [30]:

```
topics_df = pd.DataFrame(topics_dict)
```

In [31]:

```
topics_df
```

Out[31]:

	title	descriptions	url
0	3D	3D refers to the use of three-dimensional grap...	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

Loading [MathJax]/extensions/Safe.js

	title	descriptions	url
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
5	Angular	Angular is an open source web application plat...	https://github.com/topics/angular
6	Ansible	Ansible is a simple and powerful automation en...	https://github.com/topics/ansible
7	API	An API (Application Programming Interface) is ...	https://github.com/topics/api
8	Arduino	Arduino is an open source platform for buildin...	https://github.com/topics/arduino
9	ASP.NET	ASP.NET is a web framework for building modern...	https://github.com/topics/aspnet
10	Atom	Atom is a open source text editor built with w...	https://github.com/topics/atom
11	Awesome Lists	An awesome list is a list of awesome things cu...	https://github.com/topics/awesome
12	Amazon Web Services	Amazon Web Services provides on-demand cloud c...	https://github.com/topics/aws
13	Azure	Azure is a cloud computing service created by ...	https://github.com/topics/azure
14	Babel	Babel is a compiler for writing next generatio...	https://github.com/topics/babel
15	Bash	Bash is a shell and command language interpret...	https://github.com/topics/bash
16	Bitcoin	Bitcoin is a cryptocurrency developed by Satos...	https://github.com/topics/bitcoin
17	Bootstrap	Bootstrap is an HTML, CSS, and JavaScript fram...	https://github.com/topics/bootstrap
18	Bot	A bot is an application that runs automated ta...	https://github.com/topics/bot
19	C	C is a general purpose programming language th...	https://github.com/topics/c
20	Chrome	Chrome is a web browser from the tech company ...	https://github.com/topics/chrome
21	Chrome extension	Chrome extensions enable users to customize th...	https://github.com/topics/chrome-extension
22	Command line interface	A CLI, or command-line interface, is a console...	https://github.com/topics/cli
23	Clojure	Clojure is a dynamic, general-purpose programm...	https://github.com/topics/clojure
24	Code quality	Automate your code review with style, quality,...	https://github.com/topics/code-quality
25	Code review	Ensure your code meets quality standards and s...	https://github.com/topics/code-review
26	Compiler	Compilers are software that translate higher-l...	https://github.com/topics/compiler
27	Continuous integration	Automatically build and test your code as you ...	https://github.com/topics/continuous-integration
28	COVID-19	The coronavirus disease 2019 (COVID-19) is an ...	https://github.com/topics/covid-19
29	C++	C++ is a general purpose and object-oriented p...	https://github.com/topics/cpp

In [ ]:

```

```

In [ ]:

```

```

4.Create CSV file(s) with the extracted information

Loading [MathJax]/extensions/Safe.js

In [32]:

```
topics_df.to_csv('topics.csv')
```

In [ ]:

```

```

In [33]:

```
##GETTING INFOSRMAION OUT OF TOPIC PAGE
```

In [34]:

```
topic_page_url = topic_urls[0]
```

In [35]:

```
topic_page_url
```

Out[35]:

```
'https://github.com/topics/3d'
```

In [36]:

```
response = requests.get(topic_page_url)
```

In [37]:

```
response.status_code
```

Out[37]:

```
200
```

In [38]:

```
len(response.text)
```

Out[38]:

```
488666
```

In [39]:

```
topic_doc = BeautifulSoup(response.text,'html.parser')
```

In [40]:

```
h3_selection_class = 'f3 color-fg-muted text-normal lh-condensed'
repo_tags = topic_doc.find_all('h3',{'class': h3_selection_class})
```

In [41]:

```
len(repo_tags)
```

Out[41]:

```
20
```

In [42]:

```
a_tags = repo_tags[0].find_all('a')
```

In [43]:

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

Out[43]:

```
'mrdoob'
```

In [44]:

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

Out[44]:

```
'three.js'
```

In [45]:

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

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

In [46]:

```
star_tags = topic_doc.find_all('span',{'class':'Counter js-social-count'})
```

In [47]:

```
len(star_tags)
```

Out[47]:

```
20
```

In [48]:

```
star_tags[0].text.strip()
```

Out[48]:

```
'97.4k'
```

In [49]:

```
def parse_star_count(stars_str):
    stars_str = stars_str.strip()
    if stars_str[-1]=='k':
          return int(float(stars_str[:-1])* 1000)
```

In [50]:

```
parse_star_count(star_tags[0].text.strip())
```

Out[50]:

```
97400
```

In [ ]:

```

```

In [51]:

```
def get_repo_info(h3_tag, star_tag):
    #returns all the required info about repository
    a_tags = h3_tag.find_all('a')
    username = a_tags[0].text.strip()
    repo_name = a_tags[1].text.strip()
    repo_url = base_url + a_tags[1]['href']
    stars = parse_star_count(star_tags[0].text.strip())
    return username, repo_name, stars,repo_url
```

In [52]:

```
get_repo_info(repo_tags[0],star_tags[0])
```

Out[52]:

```
('mrdoob', 'three.js', 97400, 'https://github.com/mrdoob/three.js')
```

In [53]:

```
topic_repos_dict = {
    'username':[],
    'repo_name':[],
    'stars':[],
    'repo_url':[]
}


for i in range(len(repo_tags)):
    repo_info = get_repo_info(repo_tags[i],star_tags[i])
    topic_repos_dict['username'].append(repo_info[0])
    topic_repos_dict['repo_name'].append(repo_info[1])
    topic_repos_dict['stars'].append(repo_info[2])
    topic_repos_dict['repo_url'].append(repo_info[3])
```

In [54]:

```
##FINAL CODE
```

In [55]:

```
import os
def get_topic_page(topic_url):
        #download the page
        response = requests.get(topic_page_url)
        #check successful response
        if response.status_code !=200:
            raise Exception('failed to load page{}'.format(topic_url))
        #parse using BeautifulSoup
        topic_doc = BeautifulSoup(response.text,'html.parser')
        return topic_doc


def get_repo_info(h3_tag, star_tag):
    #returns all the required info about repository
    a_tags = h3_tag.find_all('a')
    username = a_tags[0].text.strip()
    repo_name = a_tags[1].text.strip()
    repo_url = base_url + a_tags[1]['href']
    stars = parse_star_count(star_tags[0].text.strip())
    return username, repo_name, stars,repo_url

def get_topic_repos(topic_doc):

        #get the h3 tag containing repo title,repo URL and username
        h3_selection_class = 'f3 color-fg-muted text-normal lh-condensed'
        repo_tags = topic_doc.find_all('h3',{'class': h3_selection_class})
        #get star tags
        star_tags = topic_doc.find_all('span',{'class':'Counter js-social-count'})

        topic_repos_dict = { 'username':[],'repo_name':[],'stars':[],'repo_url':[]}


        #get repo info
        for i in range(len(repo_tags)):
            repo_info = get_repo_info(repo_tags[i],star_tags[i])
            topic_repos_dict['username'].append(repo_info[0])
            topic_repos_dict['repo_name'].append(repo_info[1])
            topic_repos_dict['stars'].append(repo_info[2])
            topic_repos_dict['repo_url'].append(repo_info[3])

        return pd.DataFrame(topic_repos_dict)##Pandas dataframe

def scrape_topic(topic_url,path):
    if os.path.exists(path):
        print("The file {} already exists.Skipping...".format(path))
    topic_df = get_topic_repos(get_topic_page(topic_url))
    topic_df.to_csv(path)
```

In [ ]:

```

```

In [56]:

```
get_topic_repos(get_topic_page(topic_urls[6])).to_csv('3d.csv')
```

write a single function to: 1.Get the list of topics from topic page 2.Get the list of top repos from the individual topic pages 3.For each topic create CSV of top repos for the the topic

Loading [MathJax]/extensions/Safe.js

In [57]:

```
def get_topic_titles(doc):
    selection_class='f3 lh-condensed mb-0 mt-1 Link--primary'
    topic_title_tags = doc.find_all('p', {'class':selection_class})


    topic_titles = []
    for tag in topic_title_tags:
        topic_titles.append(tag.text)
    return topic_titles
    pass


def get_topic_description(doc):
    desc_selector ='f5 color-fg-muted mb-0 mt-1'
    topic_desc_tags = doc.find_all('p',{'class':desc_selector})


    topic_descs = []
    for tag in topic_desc_tags :
            topic_descs.append(tag.text.strip())
    return topic_descs


def get_topic_urls(doc):
    topic_link_tags = doc.find_all('a',{'class': 'no-underline flex-1 d-flex flex-column'}

    topic_urls = []
    base_url='https://github.com'
    for tag in topic_link_tags:
        topic_urls.append(base_url + tag['href'])
    return topic_urls


##list of topics
def scrape_topics():
    topic_url = 'https://github.com/topics'
    response = requests.get(topics_url)
    if response.status_code !=200:
            raise Exception('failed to load page{}'.format(topic_url))
    doc = BeautifulSoup(response.text,'html.parser')
    topic_dict = {
        'title': get_topic_titles(doc),
        'description': get_topic_description(doc),
        'url':get_topic_urls(doc)
    }
    return pd.DataFrame(topics_dict)
```

In [ ]:

In [58]:

```
##megafunction we put everything in single function for taking infos
def scrape_topic_repos():
    print('Scraping list of topics')
    topic_df = scrape_topics()
    #Create a folder

    os.makedirs('data',exist_ok=True)

    for index,row in topic_df.iterrows():
        print('Scraping top Repositories for "{}" '.format(row['title']))
Loading [MathJax]/extensions/Safe.js e_topic(row['url'], 'data/{}.csv'.format(row['title']))
```

In [59]:

```
scrape_topic_repos()
```

```
Scraping list of topics
Scraping top Repositories for "3D"
Scraping top Repositories for "Ajax"
Scraping top Repositories for "Algorithm"
Scraping top Repositories for "Amp"
Scraping top Repositories for "Android"
Scraping top Repositories for "Angular"
Scraping top Repositories for "Ansible"
Scraping top Repositories for "API"
Scraping top Repositories for "Arduino"
Scraping top Repositories for "ASP.NET"
Scraping top Repositories for "Atom"
Scraping top Repositories for "Awesome Lists"
Scraping top Repositories for "Amazon Web Services"
Scraping top Repositories for "Azure"
Scraping top Repositories for "Babel"
Scraping top Repositories for "Bash"
Scraping top Repositories for "Bitcoin"
Scraping top Repositories for "Bootstrap"
Scraping top Repositories for "Bot"
Scraping top Repositories for "C"
Scraping top Repositories for "Chrome"
Scraping top Repositories for "Chrome extension"
Scraping top Repositories for "Command line interface"
Scraping top Repositories for "Clojure"
Scraping top Repositories for "Code quality"
Scraping top Repositories for "Code review"
Scraping top Repositories for "Compiler"
Scraping top Repositories for "Continuous integration"
Scraping top Repositories for "COVID-19"
Scraping top Repositories for "C++"
```

In [ ]:

In [60]:

```
topic_repos_df = pd.DataFrame(topic_repos_dict)
```

In [61]:

```
topic_repos_dict
```

Out[61]:

```
{'username': ['mrdoob',
  'pmndrs',
  'libgdx',
  'BabylonJS',
  'ssloy',
  'FreeCAD',
  'lettier',
  'aframevr',
  'CesiumGS',
  'blender',
  'MonoGame',
  'metafizzy',
  'isl-org',
  'timzhang642',
  'nerfstudio-project',
  'a1studmuffin',
  'domlysz',
  'FyroxEngine',
  'openscad'],
 'repo_name': ['three.js',
  'react-three-fiber',
  'libgdx',
  'Babylon.js',
  'tinyrenderer',
  'FreeCAD',
  '3d-game-shaders-for-beginners',
  'aframe',
  'cesium',
  'blender',
  'MonoGame',
  'zdog',
  'Open3D',
  '3D-Machine-Learning',
  'nerfstudio',
  'SpaceshipGenerator',
  'BlenderGIS',
  'Fyrox',
  'model-viewer',
  'openscad'],
 'stars': [97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400,
  97400],
 'repo_url': ['https://github.com/mrdoob/three.js',
  'https://github.com/pmndrs/react-three-fiber',
  'https://github.com/libgdx/libgdx',
  'https://github.com/BabylonJS/Babylon.js',
  'https://github.com/ssloy/tinyrenderer',
  'https://github.com/FreeCAD/FreeCAD',
  'https://github.com/lettier/3d-game-shaders-for-beginners',
  'https://github.com/aframevr/aframe',
  'https://github.com/CesiumGS/cesium',
  'https://github.com/blender/blender',
  'https://github.com/MonoGame/MonoGame',
  'https://github.com/metafizzy/zdog',
  'https://github.com/isl-org/Open3D',
  'https://github.com/timzhang642/3D-Machine-Learning',
  'https://github.com/nerfstudio-project/nerfstudio',
  'https://github.com/a1studmuffin/SpaceshipGenerator',
  'https://github.com/domlysz/BlenderGIS',
  'https://github.com/FyroxEngine/Fyrox',
  'https://github.com/google/model-viewer',
  'https://github.com/openscad/openscad']}
```

In [62]:

```
topic_repos_df= pd.DataFrame(topic_repos_dict)
```

Loading [MathJax]/extensions/Safe.js

In [63]:

```
topic_repos_df
```

Out[63]:

	username	repo_name	stars	repo_url
0	mrdoob	three.js	97400	https://github.com/mrdoob/three.js
1	pmndrs	react-three-fiber	97400	https://github.com/pmndrs/react-three-fiber
2	libgdx	libgdx	97400	https://github.com/libgdx/libgdx
3	BabylonJS	Babylon.js	97400	https://github.com/BabylonJS/Babylon.js
4	ssloy	tinyrenderer	97400	https://github.com/ssloy/tinyrenderer
5	FreeCAD	FreeCAD	97400	https://github.com/FreeCAD/FreeCAD
6	lettier	3d-game-shaders-for-beginners	97400	https://github.com/lettier/3d-game-shaders-for...
7	aframevr	aframe	97400	https://github.com/aframevr/aframe
8	CesiumGS	cesium	97400	https://github.com/CesiumGS/cesium
9	blender	blender	97400	https://github.com/blender/blender
10	MonoGame	MonoGame	97400	https://github.com/MonoGame/MonoGame
11	metafizzy	zdog	97400	https://github.com/metafizzy/zdog
12	isl-org	Open3D	97400	https://github.com/isl-org/Open3D
13	timzhang642	3D-Machine-Learning	97400	https://github.com/timzhang642/3D-Machine-Lear...
14	nerfstudio-project	nerfstudio	97400	https://github.com/nerfstudio-project/nerfstudio
15	a1studmuffin	SpaceshipGenerator	97400	https://github.com/a1studmuffin/SpaceshipGener...
16	domlysz	BlenderGIS	97400	https://github.com/domlysz/BlenderGIS
17	FyroxEngine	Fyrox	97400	https://github.com/FyroxEngine/Fyrox
18	google	model-viewer	97400	https://github.com/google/model-viewer
19	openscad	openscad	97400	https://github.com/openscad/openscad

In [67]:

```
import jovian
```

In [68]:

```
jovian.commit()
```

```
[jovian] Updating notebook "ayishathrifa5716/scraping-github-topics-repositories-rough" on
https://jovian.com
[jovian] Committed successfully! https://jovian.com/ayishathrifa5716/scraping-github-topic
s-repositories-rough
```

Out[68]:

```
'https://jovian.com/ayishathrifa5716/scraping-github-topics-repositories-rough'
```

In [ ]:

In [ ]:

In [ ]:

5.Document and share your work

Loading [MathJax]/extensions/Safe.js

In [ ]:

In [ ]: