Scraping Top Repositories for GitHub Topics

What is web Scraping?

Web scraping is the process of extracting and parsing data from websites in an automated fashion using a computer program. It's a useful technique for creating datasets for research and learning.

Project Outline

- We're going to scrape https://github.com/topics
- We'll get a list of topics. For each topic, we'll 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, username, stars and repo URL
- For each topic we'll create a CSV file in the following format:

```
Repo Name,Username,Stars,Repo URL
three.js,mrdoob,69700,https://github.com/mrdoob/three.js
libgdx,libgdx,18300,https://github.com/libgdx/libgdx
```

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]:
```
141725
```

In [7]:
```
page_contents = response.text
```

In [8]:
```
web_contents = page_contents[:10000]
```

In [9]:
```
with open("webpage.html", "w") as f:
    f.write(web_contents)
```

Use Beautiful Soup to parse and extract information

In [10]:
```
! pip install beautifulsoup4  --quiet
```

In [11]:
```
from bs4 import BeautifulSoup
```

In [12]:
```
doc = BeautifulSoup(page_contents,'html.parser')
```

In [13]:
```
type(doc)
```

Out[13]:
```
bs4.BeautifulSoup
```

In [14]:
```
selection_class = 'f3 lh-condensed mb-0 mt-1 Link--primary'
title_tages = doc.find_all('p',{'class':selection_class})
```

In [15]:
```
len(title_tages)
```

Out[15]:
```
30
```

In [16]:
```
title_tages[:6]
```

Out[16]:
```
[<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>,
 <p class="f3 lh-condensed mb-0 mt-1 Link--primary">Angular</p>]
```

In [17]:
```
desc_tags = doc.find_all('p',{'class':'f5 color-text-secondary mb-0 mt-1'})
```

In [18]:
```
len(desc_tags)
```

Out[18]:
```
30
```

In [19]:
```
desc_tags[:6]
```

Out[19]:
```
[<p class="f5 color-text-secondary mb-0 mt-1">
               3D modeling is the process of virtually developing the surface and structure of a 3D object.
             </p>,
 <p class="f5 color-text-secondary mb-0 mt-1">
               Ajax is a technique for creating interactive web applications.
             </p>,
 <p class="f5 color-text-secondary mb-0 mt-1">
               Algorithms are self-contained sequences that carry out a variety of tasks.
             </p>,
 <p class="f5 color-text-secondary mb-0 mt-1">
               Amp is a non-blocking concurrency framework for PHP.
             </p>,
 <p class="f5 color-text-secondary mb-0 mt-1">
               Android is an operating system built by Google designed for mobile devices.
             </p>,
 <p class="f5 color-text-secondary mb-0 mt-1">
               Angular is an open source web application platform.
             </p>]
```

In [20]:
```
link_tags = doc.find_all('a',{'class':'d-flex no-underline'})
```

In [21]:
```
len(link_tags)
```

Out[21]:
```
30
```

In [22]:
```
topic_urls = "https://github.com" + link_tags[0]['href']
print(topic_urls)
```

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

In [23]:
```
topic_titles = []

for tag in title_tages:
    topic_titles.append(tag.text)
print(topic_titles)
```

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

In [24]:
```
topic_descr = []

for tag in desc_tags:
    topic_descr.append(tag.text.strip())

topic_descr[:6]
```

Out[24]:
```
['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 framework for PHP.',
 'Android is an operating system built by Google designed for mobile devices.',
 'Angular is an open source web application platform.']
```

In [25]:
```
topic_urls = []
base_url = 'https://github.com'
for tag in 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,
               'description':topic_descr,
               'url':topic_urls
               }
```

In [29]:
```
topics_df = pd.DataFrame(topics_dict)
```

In [30]:
```
topics_df
```

Out[30]:

	title	description	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 framework fo...	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 hardware and softwar...	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	Google Chrome Extensions are add-ons that allo...	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

Create CSV file

In [31]:
```
topics_df.to_csv('topics.csv', index=None)
```

In [32]:
```
sheet = 'topicss.xlsx'        # convert to excel file
topics_df.to_excel(sheet)
```

In [33]:
```
topic_page_url = topic_urls[0]
```

In [34]:
```
topic_page_url
```

Out[34]:
```
'https://github.com/topics/3d'
```

In [35]:
```
response = requests.get(topic_page_url)
```

In [36]:
```
response
```

Out[36]:
```
<Response [200]>
```

In [37]:
```
len(response.text)
```

Out[37]:
```
632380
```

In [38]:
```
topic_doc = BeautifulSoup(response.text, 'html.parser')
```

In [39]:
```
repo_tags = topic_doc.find_all('h3',{'class':'f3 color-text-secondary text-normal lh-condensed'})
```

In [40]:
```
len(repo_tags)
```

Out[40]:
```
30
```

In [41]:
```
a_tags = repo_tags[0].find_all('a')
```

In [42]:
```
a_tags[0].text.strip()
```

Out[42]:
```
'mrdoob'
```

In [43]:
```
repo_url = base_url + a_tags[1]['href']
print(repo_url)
```

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

In [44]:
```
star_tags = topic_doc.find_all('a',{'class':'social-count float-none'})
```

In [45]:
```
len(star_tags)
```

Out[45]:
```
30
```

In [46]:
```
star_tags[0].text.strip()
```

Out[46]:
```
'74.5k'
```

In [47]:
```
def parse_star_count(stars_str):
    stars_str = stars_str.strip()
    if stars_str[-1] == 'k':
        return int(float(stars_str[:-1]) * 1000)
```

In [48]:
```
parse_star_count(star_tags[0].text.strip())
```

Out[48]:
```
74500
```

In [49]:
```
def get_repo_info(h3_tag,star_tag):
    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_tag.text.strip())
    return username,repo_name,stars,repo_url
```

In [50]:
```
get_repo_info(repo_tags[0],star_tags[0])
```

Out[50]:
```
('mrdoob', 'three.js', 74500, 'https://github.com/mrdoob/three.js')
```

In [51]:
```
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])
```

Final code to get topic_repos for all topics of github

In [52]:
```
def get_topic_page(topic_url):
    response = requests.get(topic_url)
    if response.status_code != 200:
        raise Exception('failed to load page{}'.format(topic_url))

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

def get_repo_info(h3_tag,star_tag):
    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_tag.text.strip())
    return username,repo_name,stars,repo_url


def get_topic_repos(topic_doc):
    repo_tags = topic_doc.find_all('h3',{'class':'f3 color-text-secondary text-normal lh-condensed'})
    star_tags = topic_doc.find_all('a',{'class':'social-count float-none'})
    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])

    return pd.DataFrame(topic_repos_dict)
```

In [53]:
```
get_topic_repos(get_topic_page(topic_urls[4]))
```

Out[53]:

	username	repo_name	stars	repo_url
0	flutter	flutter	131000	https://github.com/flutter/flutter
1	justjavac	free-programming-books-zh_CN	83100	https://github.com/justjavac/free-programming-...
2	Genymobile	scrcpy	55000	https://github.com/Genymobile/scrcpy
3	Hack-with-Github	Awesome-Hacking	46000	https://github.com/Hack-with-Github/Awesome-Ha...
4	google	material-design-icons	43900	https://github.com/google/material-design-icons
5	wasabeef	awesome-android-ui	41400	https://github.com/wasabeef/awesome-android-ui
6	square	okhttp	40900	https://github.com/square/okhttp
7	android	architecture-samples	39500	https://github.com/android/architecture-samples
8	square	retrofit	38800	https://github.com/square/retrofit
9	Solido	awesome-flutter	37600	https://github.com/Solido/awesome-flutter
10	PhilJay	MPAndroidChart	34000	https://github.com/PhilJay/MPAndroidChart
11	fastlane	fastlane	32900	https://github.com/fastlane/fastlane
12	shadowsocks	shadowsocks-android	31900	https://github.com/shadowsocks/shadowsocks-and...
13	airbnb	lottie-android	31800	https://github.com/airbnb/lottie-android
14	bumptech	glide	31800	https://github.com/bumptech/glide
15	Trinea	android-open-project	30800	https://github.com/Trinea/android-open-project
16	Blankj	AndroidUtilCode	29900	https://github.com/Blankj/AndroidUtilCode
17	xitu	gold-miner	29600	https://github.com/xitu/gold-miner
18	bilibili	ijkplayer	29500	https://github.com/bilibili/ijkplayer
19	zxing	zxing	28400	https://github.com/zxing/zxing
20	codepath	android_guides	27500	https://github.com/codepath/android_guides
21	skylot	jadx	27200	https://github.com/skylot/jadx
22	square	leakcanary	26700	https://github.com/square/leakcanary
23	laurent22	joplin	25500	https://github.com/laurent22/joplin
24	alibaba	fastjson	23800	https://github.com/alibaba/fastjson
25	scwang90	SmartRefreshLayout	23000	https://github.com/scwang90/SmartRefreshLayout
26	CymChad	BaseRecyclerViewAdapterHelper	22300	https://github.com/CymChad/BaseRecyclerViewAda...
27	react-native-elements	react-native-elements	21400	https://github.com/react-native-elements/react...
28	google	iosched	20800	https://github.com/google/iosched
29	NativeScript	NativeScript	20500	https://github.com/NativeScript/NativeScript

Dataframe of single topic_repos_dict

In [54]:
```
topic_repos_df = pd.DataFrame(topic_repos_dict)
```

In [55]:
```
topic_repos_df
```

Out[55]:

	username	repo_name	stars	repo_url
0	mrdoob	three.js	74500	https://github.com/mrdoob/three.js
1	libgdx	libgdx	19000	https://github.com/libgdx/libgdx
2	pmndrs	react-three-fiber	15100	https://github.com/pmndrs/react-three-fiber
3	BabylonJS	Babylon.js	14900	https://github.com/BabylonJS/Babylon.js
4	aframevr	aframe	13100	https://github.com/aframevr/aframe
5	ssloy	tinyrenderer	11300	https://github.com/ssloy/tinyrenderer
6	lettier	3d-game-shaders-for-beginners	11200	https://github.com/lettier/3d-game-shaders-for...
7	FreeCAD	FreeCAD	9900	https://github.com/FreeCAD/FreeCAD
8	metafizzy	zdog	8700	https://github.com/metafizzy/zdog
9	CesiumGS	cesium	7500	https://github.com/CesiumGS/cesium
10	timzhang642	3D-Machine-Learning	7100	https://github.com/timzhang642/3D-Machine-Lear...
11	a1studmuffin	SpaceshipGenerator	6900	https://github.com/a1studmuffin/SpaceshipGener...
12	isl-org	Open3D	5500	https://github.com/isl-org/Open3D
13	spritejs	spritejs	4600	https://github.com/spritejs/spritejs
14	tensorspace-team	tensorspace	4500	https://github.com/tensorspace-team/tensorspace
15	jagenjo	webglstudio.js	4400	https://github.com/jagenjo/webglstudio.js
16	YadiraF	PRNet	4400	https://github.com/YadiraF/PRNet
17	AaronJackson	vrn	4400	https://github.com/AaronJackson/vrn
18	domlysz	BlenderGIS	4300	https://github.com/domlysz/BlenderGIS
19	openscad	openscad	4300	https://github.com/openscad/openscad
20	ssloy	tinyraytracer	3900	https://github.com/ssloy/tinyraytracer
21	mosra	magnum	3700	https://github.com/mosra/magnum
22	google	model-viewer	3400	https://github.com/google/model-viewer
23	blender	blender	3400	https://github.com/blender/blender
24	gfxfundamentals	webgl-fundamentals	3200	https://github.com/gfxfundamentals/webgl-funda...
25	cleardusk	3DDFA	3100	https://github.com/cleardusk/3DDFA
26	jasonlong	isometric-contributions	3000	https://github.com/jasonlong/isometric-contrib...
27	rg3dengine	rg3d	2800	https://github.com/rg3dengine/rg3d
28	cnr-isti-vclab	meshlab	2400	https://github.com/cnr-isti-vclab/meshlab
29	antvis	L7	2400	https://github.com/antvis/L7

write a single function

In [56]:
```
import os

def get_topic_page(topic_url):
    # Download the page
    response = requests.get(topic_url)
    # Check successful response
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(topic_url))
    # Parse using Beautiful soup
    topic_doc = BeautifulSoup(response.text, 'html.parser')
    return topic_doc

def get_repo_info(h1_tag, star_tag):
    # returns all the required info about a repository
    a_tags = h1_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_tag.text.strip())
    return username, repo_name, stars, repo_url

def get_topic_repos(topic_doc):
    # Get the h1 tags containing repo title, repo URL and username
    h1_selection_class = 'f3 color-text-secondary text-normal lh-condensed'
    repo_tags = topic_doc.find_all('h1', {'class': h1_selection_class} )
    # Get star tags
    star_tags = topic_doc.find_all('a', { 'class': 'social-count float-none'})

    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)


def scrape_topic(topic_url, path):
    if os.path.exists(path):
        print("The file {} already exists. Skipping...".format(path))
        return
    topic_df = get_topic_repos(get_topic_page(topic_url))
    topic_df.to_csv(path, index=None)
```

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

def get_topic_descs(doc):
    desc_selector = 'f5 color-text-secondary 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': 'd-flex no-underline'})
    topic_urls = []
    base_url = 'https://github.com'
    for tag in topic_link_tags:
        topic_urls.append(base_url + tag['href'])
    return topic_urls


def scrape_topics():
    topics_url = 'https://github.com/topics'
    response = requests.get(topics_url)
    if response.status_code != 200:
        raise Exception('Failed to load page {}'.format(topic_url))
    topics_dict = {
        'title': get_topic_titles(doc),
        'description': get_topic_descs(doc),
        'url': get_topic_urls(doc)
    }
    return pd.DataFrame(topics_dict)
```

In [58]:
```
def scrape_topics_repos():
    print('Scraping list of topics')
    topics_df = scrape_topics()

    os.makedirs('data', exist_ok=True)
    for index, row in topics_df.iterrows():
        print('Scraping top repositories for "{}"'.format(row['title']))
        scrape_topic(row['url'], 'data/{}.csv'.format(row['title']))
```

In [59]:
```
scrape_topics_repos()
```

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

Thank You!