

```

import requests
from bs4 import BeautifulSoup
from tqdm import tqdm
import re
from collections import Counter
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

base_url = "https://github.com/PacktPublishing?page={}&tab=r
headers = {"User-Agent": "Mozilla/5.0"}
all_data = []

for page in tqdm(range(1, 298)): # с т р а н и ц ы 1-297
    url = base_url.format(page)
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, "html.parser")

    repos = soup.select("li[itemprop='owns']")

    for repo in repos:
        title_tag = repo.select_one("a[itemprop='name codeRe
        if not title_tag:
            continue
        title = title_tag.text.strip()
        href = "https://github.com" + title_tag['href']
        stars_tag = repo.select_one("a[href$='/stargazers']")
        stars = stars_tag.text.strip() if stars_tag else "0"
        all_data.append((title, href, stars))

```

100%|██████████| 297/297 [00:32<00:00, 9.07it/s]

```

all_words = []
for title, _, _ in all_data:
    words = re.findall(r'\b\w+\b', title.lower())
    all_words.extend(words)

```

```

from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
import time

```

```

options = Options()
options.add_argument("--headless")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")
driver = webdriver.Chrome(options=options)

```

```

# Загружаем страницу
url = "https://github.com/PacktPublishing?page=1&tab=repo
driver.get(url)

```

ValueError

ValueError ...



다음 오류를 설명해 주세요.

ValueError: We need at least 1



Hypothesis

The error "ValueError: We need at least 1 word to plot a word cloud, got 0" arises because the `filtered_words` list is empty, resulting in an empty string being passed to the

`WordCloud.generate()` method.

This happens because either:

1. The initial word extraction from the web scraping process failed to retrieve any words.
2. The filtering process using `stopwords` removed all the extracted words, leaving `filtered_words` empty.

Since the global variables show that `filtered_words` is an empty list, the issue is likely due to the filtering process removing all the words from the list.

Suggested Changes

Adding words to stopwords

You might need to add more words to your `stopwords` set. These could include common words specific to the dataset that don't provide meaningful insights, such as

```
time.sleep(3)

soup = BeautifulSoup(driver.page_source, "html.parser")

# Новый селектор: ищем ссылки на репозитории
repos = soup.select("a[href^='/PacktPublishing/']")

print("🔍 Найдено репозиторий в:", len(repos))
for a in repos[:5]:
    title = a.text.strip()
    href = "https://github.com" + a['href']
    print("-", title, "→", href)

driver.quit()
```

```
# Сохраним в таблицу
df = pd.DataFrame(data, columns=["Title", "URL"])
df.to_excel("packt_repos.xlsx", index=False)
print("✅ Готово! Данные сохранены в
```

100% | 297/297 [15:59<00:00, 3.23s/it]
✅ Готово! Данные сохранены в 'p

```
import re
from collections import Counter
```

```
all_words = []
for title in df['Title']:
    words = re.findall(r'\b\w+\b', title.lower()) # пер
    all_words.extend(words)
```

```
top10 = Counter(all_words).most_common(10)
print("📈 Топ-10 слов:", top10)
```

📈 📈 Топ-10 слов: [('0', 5310), ('with', 885), ('

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
stopwords = set([
    "and", "with", "the", "for", "from", "using", "in", "of",
    "a", "an", "at", "as", "is", "this", "that", "you", "you"
])
```

```
filtered = [w for w in all_words if w not in stopwords]
text = " ".join(filtered)
```

```
wc = WordCloud(width=800, height=400, background_color='white')
```

```
plt.figure(figsize=(12, 6))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

```
# ... (your existing code) ..
```

```
text = " ".join(filtered_words)
```

```
# Check if filtered_words is
if not filtered_words:
```

```
    print("filtered_words is
    text = "example sample data")
```

```
wc = WordCloud(width=800, height=400, background_color='white')
# ... (rest of your code) ...
```

[Use code with caution](#)

▶ 출처

답변 평가 👍 💬



```

from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from bs4 import BeautifulSoup
import pandas as pd
import time
from tqdm import tqdm

options = Options()
options.add_argument("--headless")
options.add_argument("--no-sandbox")
options.add_argument("--disable-dev-shm-usage")

driver = webdriver.Chrome(options=options)

data = []

for page in tqdm(range(1, 298)): # 1~297
    url = f"https://github.com/PacktPublishing?page={page}&ta
    driver.get(url)
    time.sleep(2.5)

    soup = BeautifulSoup(driver.page_source, "html.parser")
    repos = soup.select("li[itemprop='owns']")

    for repo in repos:
        title_tag = repo.select_one("a[itemprop='name codeRep
        if not title_tag:
            continue
        title = title_tag.text.strip()
        href = "https://github.com" + title_tag['href']

        star_tag = repo.select_one("a[href$='/stargazers']")
        stars = star_tag.text.strip() if star_tag else "0"

        data.append((title, href, stars))

```

```
driver.quit()
```

```
# Сохраняем в Excel
```

```
df = pd.DataFrame(data, columns=["Title", "URL", "Stars"])
```

```
df.to_excel("packt_books.xlsx", index=False)
```

```
print("✅ Данные сохранены в packt_books.xlsx")
```

↻ 100% | ██████████ | 297/297 [15:55<00:00, 3.22s/it] ✅ Д



여기에 메시지를 입력하세요.



0/2000

답변에 Google의 관점을 반영하지 않는 부정확하거나 불쾌감을 주는 정보가 표시될 수 있습니다.
[자세히 알아보기](#)